

**REPUBLIQUE DU SENEGAL**

Ministère de l’Enseignement Supérieur de la Recherche

et de l’Innovation

-------------------

UNIVERSITE GASTON BERGER



**Rapport : Application de gestion médicale**

**Présenté par** : Maty SECK

Pape Abdoulaye NDIAYE

**Professeur** : D. MBOUP

**Classe**: ING 3 GeIT

**Année académique** : 2024-2025

# **Introduction**

Le **Projet Scala** est une application web développée dans le cadre d’un travail de groupe visant à concevoir un système complet de gestion médicale pour une structure de soins. Ce projet repose sur une architecture moderne et modulaire, combinant un backend robuste écrit en **Scala** avec le **Play Framework**, et une interface utilisateur interactive réalisée avec **Angular**. L’objectif principal de cette application est de faciliter la gestion des patients, des dossiers médicaux, du personnel soignant, des ressources matérielles, ainsi que des opérations courantes telles que les rendez-vous, la facturation ou encore les recherches avancées.

La pertinence de ce projet réside dans sa capacité à centraliser les informations médicales, à fluidifier la communication entre les différents services et à garantir un haut niveau de sécurité et de performance dans le traitement des données sensibles.

# **Présentation des fonctionnalités**

## **Gestion des patients**

L’une des fonctionnalités fondamentales de l’application est la gestion des patients. Dès l’arrivée d’un patient, **l’utilisateur autorisé peut créer une fiche contenant toutes les informations personnelles nécessaires**. Par la suite**, ces données peuvent être modifiées ou supprimées** selon les besoins. Un **système de visualisation permet d’accéder rapidement à la fiche complète d’un patient**, incluant son identifiant, ses données démographiques et ses informations de contact. **Une fonctionnalité de recherche par identifiant (ID)** facilite également la navigation dans les bases de données, surtout en contexte hospitalier où les volumes d’information peuvent être importants.

## **Gestion des dossiers médicaux**

À chaque patient est **automatiquement associé un dossier médical** dès sa création. Ce dossier est le cœur du suivi médical et regroupe l’ensemble des informations cliniques liées au patient : antécédents médicaux, allergies, traitements en cours ou passés, diagnostics, comptes rendus médicaux et notes du personnel soignant. Ce système permet de mettre à jour en temps réel les données, garantissant ainsi une traçabilité optimale et une continuité des soins. Les professionnels de santé peuvent également y ajouter des notes à tout moment, ce qui enrichit le suivi clinique et favorise la collaboration entre praticiens.

## **Sécurité et authentification**

La sécurité des données étant cruciale dans le domaine médical, le projet intègre un mécanisme d’authentification basé sur **les** **JWT (JSON Web Tokens)**. Lorsqu’un utilisateur s’authentifie, un jeton est généré et doit être fourni pour toute interaction ultérieure avec le serveur. Les routes du backend sont protégées via un système de directives (authDirective) qui vérifie la validité du jeton à chaque requête. Ce mécanisme assure que seules les personnes autorisées ont accès aux informations sensibles, conformément aux bonnes pratiques en matière de cybersécurité.

## **Fonctionnalités complémentaires**

En plus des fonctions de base, le projet inclut plusieurs modules avancés qui enrichissent l’application. Le premier est un module de **gestion du personnel médical**, permettant d’ajouter, modifier ou consulter les informations sur les médecins, infirmiers et autres agents de santé. Ensuite, un système de **gestion des chambres et du matériel médical** est intégré, permettant de suivre l’occupation des lits, la disponibilité des équipements, et leur état.

Un **module de prise de rendez-vous** permet aux patients et aux professionnels de santé d’organiser des consultations selon les disponibilités. Ce module est interconnecté avec les autres parties de l’application pour éviter les conflits de planning.

Enfin, un système complet de **facturation et de paiement** est proposé, permettant la génération de factures selon les soins reçus, la prise en compte des assurances, et la gestion des paiements.

L’ensemble de ces fonctionnalités est accompagné d’un moteur de **recherche avancée**, capable de filtrer efficacement les patients, les médecins, les ressources ou les dossiers selon divers critères.

# **Architecture et organisation du projet**

L'application repose sur une architecture **MVC (Modèle – Vue – Contrôleur)** clairement séparée, garantissant la maintenabilité et l’évolutivité du projet.

## **Le Backend avec Scala**

Le backend est développé en **Scala**, un langage puissant et expressif bien adapté aux applications web complexes. Nous avons utilisé le **Play Framework**, qui offre une structure modulaire et performante pour la gestion des routes, des requêtes HTTP et des communications avec la base de données.

Les **modèles** définissent les différentes entités manipulées dans l’application, telles que Patient, DossierMedical, RendezVous, Facture, etc. Ces modèles sont reliés à une base de données **PostgreSQL** via l’ORM **Slick**, qui permet une écriture fluide et sécurisée des requêtes SQL.

La **logique métier** est centralisée dans des services Scala qui se chargent des opérations complexes : traitement des données, règles de validation, interactions multi-tables, etc. Les **routes REST** exposées par le backend permettent à Angular de communiquer avec le serveur de manière sécurisée.

**Le Frontend avec Angular**

Le frontend est conçu avec le framework **Angular**, reconnu pour sa réactivité et sa structuration en composants. Chaque section de l’application (patients, dossiers, personnel, rendez-vous) est divisée en composants Angular, **qui dialoguent avec le backend via des services utilisant HttpClient**. Le routage Angular permet une navigation fluide et dynamique, en fonction des paramètres comme les identifiants ou les statuts des utilisateurs.

**Échange de données et communication**

Les données sont échangées entre le frontend et le backend en **format JSON**, ce qui permet une compatibilité totale avec les standards du web moderne. Toutes les interactions sont assurées via des **API REST** conçues pour être performantes, lisibles et facilement exploitables.

**Choix technologiques**

Notre choix technologique repose sur des outils modernes, largement utilisés dans les environnements professionnels :

| **Élément** | **Choix** |
| --- | --- |
| **Langage backend** | **Scala** |
| **Framework Web** | **Play Framework** |
| **Base de données** | **PostgreSQL** |
| **ORM** | **Slick** |
| **Authentification** | **JWT (JSON Web Token)** |
| **Frontend** | **Angular** |
| **Échange de données** | **JSON via API REST** |
| **Architecture logicielle** | **MVC (Modèle-Service-Route)** |
| **Contrôle de version** | **Git + GitHub** |

# **Conclusion**

Le **Projet Scala** a permis à notre groupe d'explorer et de mettre en œuvre des technologies modernes dans un contexte applicatif réel et utile. Cette plateforme de gestion médicale répond à des exigences techniques et fonctionnelles élevées, tout en restant modulable pour une extension future.

Ce projet nous a permis d’approfondir notre compréhension de l’architecture logicielle, de la communication client-serveur, de la gestion des données sensibles, ainsi que de l’organisation du travail en équipe dans un environnement de développement professionnel. À l’avenir, le système pourrait être enrichi par des modules d’intelligence artificielle pour assister au diagnostic, ou encore intégrer des systèmes de messagerie sécurisée pour les patients.

Souhaitez-vous que je vous exporte ce rapport en **PDF** ou **Word** ? Je peux aussi ajouter une page de garde, une table des matières ou une conclusion personnalisée selon votre rôle dans le projet.