

Rapport de Projet : Application de Gestion d'un Hôpital

1. Introduction

L'objectif de ce projet est de développer une application web complète pour la gestion d'un hôpital. Cette application permet de gérer les patients, médecins, infirmiers, chambres, matériels, paiements, rendez-vous, hospitalisations, et notes médicales. Elle est composée d'une interface frontend en Angular et d'un backend en Play Framework (Scala).

2. Architecture fonctionnelle

L'architecture fonctionnelle repose sur la gestion de plusieurs entités clés liées aux activités de l'hôpital :

Patients : création, modification, suppression, listing.

Médecins / Infirmiers : même fonctionnalité que les patients.

Chambres : gestion du nombre de lits disponibles.

Matériels : gestion des équipements médicaux.

Paiements : gestion des paiements effectués par les patients.

Rendez-vous : planification avec les médecins.

Hospitalisations : lien entre patients, chambres, durée de séjour.

Notes médicales : suivi des diagnostics et traitements.

3. Architecture logicielle

L'application suit une architecture en couches classique :

3.1 Frontend (Angular)

Composants : chaque entité (Patient, Médecin, etc.) a deux composants principaux : list et form.

Services : permettent de communiquer avec le backend via HTTP (GET, POST, PUT, DELETE).

Routing : configuré pour permettre la navigation entre les différents modules.

3.2 Backend (Play Framework)

Contrôleurs : exposent des endpoints REST pour chaque entité.

Services : contiennent la logique métier et communiquent avec la base de données.

Modèles : entités représentant les tables de la base de données.

4. Choix technologiques

Technologie	Rôle
Angular	Frontend moderne et dynamique
TypeScript	Langage du frontend Angular
Bootstrap	Mise en forme des composants
Play Framework	Backend en Scala
Scala	Langage serveur
Slick + MySQL	ORM et base de données
Git + GitHub	Gestion de version et déploiement

5. Conclusion

Le projet offre une solution complète pour la gestion hospitalière. Les modules sont bien structurés et extensibles. Le choix d'Angular et Play Framework assure une bonne séparation des responsabilités, une bonne performance, et une facilité d'entretien.