Rapport de Projet - Réservation de Rendez-vous Médical

# Introduction

Le projet de réservation de rendez-vous médical a été conçu pour offrir une solution conviviale et efficace aux utilisateurs qui souhaitent planifier des rendez-vous médicaux en ligne. Dans ce rapport, nous justifions les choix technologiques, évaluons le respect des normes et principes, et présentons les résultats et enseignements tirés de ce projet.

## Portée du Projet

La portée du projet incluait la création d'une application web permettant aux utilisateurs de rechercher des hôpitaux, de planifier des rendez-vous, et de consulter leurs rendez-vous existants.

# Justification des Technologies Utilisées

## Front

### React

React a été choisi pour son approche déclarative et sa flexibilité. Il permet une construction modulaire de l'interface utilisateur, facilitant ainsi le développement, la maintenance et l'évolutivité de l'application.

### React Router

React Router a été utilisé pour la gestion de la navigation, offrant une expérience utilisateur fluide et une synchronisation efficace entre l'URL et l'état de l'application.

### Google Maps API et @react-google-maps/api

L'intégration de Google Maps API a été cruciale pour permettre aux utilisateurs de sélectionner des emplacements de rendez-vous. La bibliothèque @react-google-maps/api simplifie cette intégration avec React.

### Jest

Jest a été choisi comme framework de test en raison de sa simplicité, de sa robustesse et de son intégration transparente avec les projets React.

## Back

### Java

Java est sélectionné en raison de sa fiabilité, de sa gestion automatisée de la mémoire et de sa robustesse, faisant de lui un choix solide pour les applications critiques. Son écosystème étendu offre un support riche avec des bibliothèques et des frameworks, facilitant le développement d'applications complexes. La nature orientée objet de Java est également bien adaptée à la modélisation des entités dans le contexte d'un système de réservation médicale.

### Spring Boot

Spring Boot est retenu pour sa simplicité, favorisant la convention plutôt que la configuration, ce qui accélère le développement d'applications Java. Les fonctionnalités telles que l'injection de dépendances et la configuration automatique améliorent la productivité des développeurs.

### MySQL

MySQL est choisi pour sa nature relationnelle, adaptée à la modélisation des relations entre les entités dans un système de réservation médicale. En tant que base de données open source, MySQL offre une flexibilité et une accessibilité accrues, souvent bénéfiques pour les projets avec des contraintes budgétaires. Sa popularité dans l'industrie, due à sa performance et à sa stabilité, en fait un choix courant pour les applications de gestion de données.

# Respect des Normes et Principes

## Respect des Normes RGPD

Le projet a été développé en tenant compte des principes de protection des données. Aucune donnée utilisateur sensible n'est stockée localement, et l'utilisation de données de localisation est strictement conforme aux pratiques de confidentialité.

## Architecture Modulaire

L'application suit des principes d'architecture modulaire avec des composants React indépendants, favorisant la maintenabilité et la réutilisabilité du code.

# Résultats et Enseignements

## Ce qui a fonctionné

Intégration Réussie de Google Maps : L'intégration de Google Maps a été réalisée avec succès, offrant aux utilisateurs une expérience fluide pour la sélection d'emplacements.

Navigation Efficace : React Router a permis une gestion efficace de la navigation au sein de l'application, assurant une transition harmonieuse entre les différentes pages.

Le système de réservation de rendez-vous a été mis en œuvre avec succès, permettant aux utilisateurs de créer des rendez-vous et de les consulter.

L'intégration de Spring Boot et MySQL a facilité le développement et la gestion des données.

## Enseignements tirés

Tests essentiels : Les tests unitaires ainsi que selenium ont été cruciaux pour garantir le bon fonctionnement de l'application. L'expansion des tests pourrait améliorer davantage la robustesse du projet.

Les retours d'utilisateurs sur l'interface utilisateur et les fonctionnalités seront cruciaux pour le succès du projet final.

L'expérience utilisateur et la simplicité d'utilisation sont des facteurs clés à prendre en compte.

## A prevoir pour le projet réel

Ce projet constitue une base solide pour le développement ultérieur du système de réservation de rendez-vous médicaux, avec des perspectives positives quant à sa qualité et sa convivialité, ainsi que ses performances.

### Page de Connexion et Gestion des Rôles avec Spring Security

L'intégration de Spring Security dans notre application de réservation médicale est essentielle pour garantir la sécurité et le contrôle d'accès aux fonctionnalités. Voici ce que nous envisageons pour la mise en œuvre :

1. Authentification Sécurisée :

Mise en place d'une page de connexion robuste utilisant Spring Security.

Utilisation de mécanismes d'authentification standard tels que le formulaire de connexion.

2. Attribution de Rôles :

Définition de rôles spécifiques tels que "UTILISATEUR" et "ADMINISTRATEUR" pour les utilisateurs.

Les rôles détermineront les autorisations accordées à chaque utilisateur.

3. Protection des APIs :

Utilisation d'annotations Spring Security comme @PreAuthorize pour définir des autorisations spécifiques sur les méthodes et les APIs.

Les rôles attribués aux utilisateurs seront utilisés dans ces annotations pour définir les accès autorisés.

4. Gestion des Erreurs et des Sessions :

Configuration de mécanismes de gestion des erreurs d'authentification pour rediriger les utilisateurs vers des pages d'erreur personnalisées.

Mise en œuvre de la gestion des sessions pour assurer la sécurité contre les attaques par session.

5. Intégration avec d'Autres Services :

Exploration des possibilités d'intégration de Spring Security avec d'autres services d'authentification tels que OAuth, LDAP, etc.

Cette intégration permettra de répondre à des exigences de sécurité spécifiques du projet.

Avantages Anticipés :

Renforcement de la sécurité de l'application grâce à une authentification solide.

Contrôle d'accès basé sur les rôles pour sécuriser chaque fonctionnalité.

Protection des APIs contre les accès non autorisés.

Gestion efficace des erreurs et des sessions, améliorant ainsi la fiabilité.

Flexibilité pour intégrer d'autres services d'authentification selon les besoins.

Cette approche renforce la confiance dans la sécurité de notre application de réservation médicale et assure une expérience utilisateur sécurisée. Les fonctionnalités avancées de Spring Security seront exploitées pour répondre aux normes de sécurité les plus élevées, garantissant ainsi le succès du projet.