MISE EN PLACE D'UNE APPLICATION DES GESTIONS DES RDVS

ENONCE

Evaluer La planification d'un Rendez-vous entre un patient et un Médecin peut s'avérer compliquée car cela dépend de plusieurs facteurs exogènes.

Les solutions collaboratives de prise de rendez-vous résolvent ce problème en donnant la possibilité aux patients d'accéder à un agenda mettant en évidence les disponibilités d'un médecin. Ces patients pourront ensuite faire une demande de rendez-vous. Les médecins sont libres d'accepter ou décliner la demande.

Dans cette perspective, nous avons pensé à développer une application Web de planification des rendez-vous en ligne offrant aux patients et médecins un portail permettant aux premiers de prendre un rendez-vous en ligne et aux derniers de suivre et de gérer ces rendez-vous.

Ce système de gestion des rendez-vous permet aux patients et médecins de collaborer dans un espace de travail unique. Il permet aussi la mise à jour et l'accès aux données en temps réel.

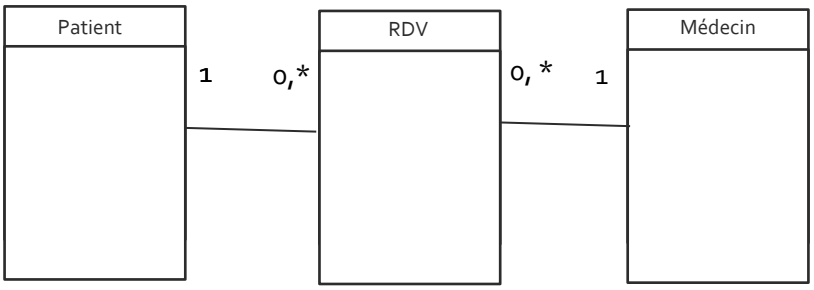
(Optionnel) : Des fonctionnalités supplémentaires peuvent être incluses : des e-mails automatiques intégrés de notifications.

L'exercice consiste à réaliser une application Web de planification des rendez-vous entre patients et médecins avec l'automatisation de tous les processus impliqués.

IDENTIFICATION DES ACTEURS

- Le Patient (sans authentification) a le droit de :
  - Choisir un médecin et visualiser son agenda.
  - Prendre/Annuler un rendez-vous sur un Agenda ; (chaque RDV doit avoir un code unique que le patient doit garder pour l'annulation)
- Le Médecin (après authentification) a le droit de :
  - Afficher les rendez-vous ;
  - Accepter/Refuser un rendez-vous ;
  - Gérer (CRUD) le contenu de son agenda

DIAGRAMME DE CLASSES



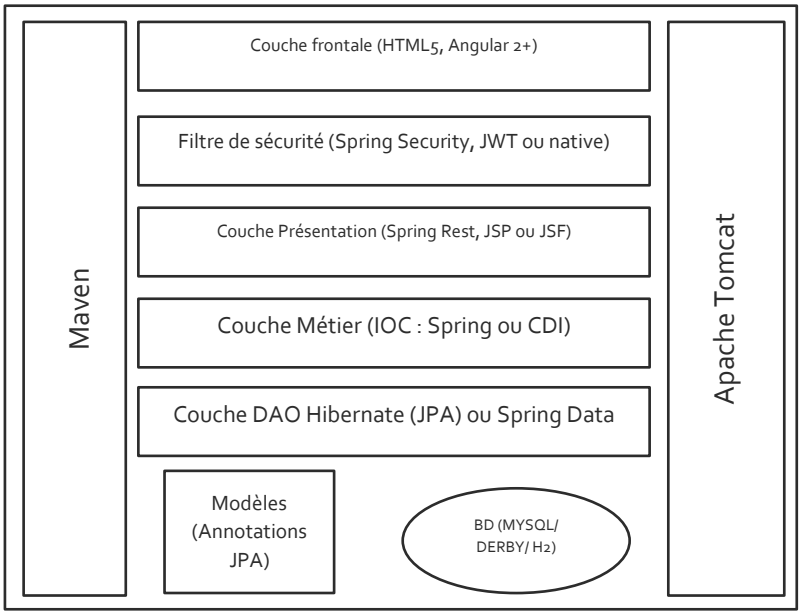
ARCHITECTURE TECHNIQUE

Nous demandons une structuration de l'application en couches, en factorisant les briques applicatives et en structurant les échanges entre les différentes couches pour que les changements dans l'implémentation d'une couche n'affectent pas la façon dont elle est utilisée par la couche supérieure.

Pour les technologies, nous nous orientons principalement sur les nouvelles technologies du marché à savoir : Spring, Hibernate JWT, Angular …

(Optionnel) Vous pouvez envisager une brique pour les tests unitaires via Junit ou une autre bibliothèque des TUs.

Pour simplifier la configuration du projet, nous proposons de travailler avec Spring Boot (via les starters) mais vous avez le libre choix de partir sur une configuration de base en respectant le schéma ci-dessous :



Pour la couche frontale, il est préférable d'utiliser Bootstrap pour assurer la responsivité et l'ergonomie de l'application. C'est possible aussi de ne pas utiliser Angular si vous n'êtes pas familiarisé avec la technologie.

Le filtre de sécurité doit concerner les opérations des médecins (vous pouvez envisager une page d'authentification)

(Optionnel) : L'interface de l'agenda peut se réaliser via un code natif ou en se basant sur des outils comme Angular Calendar, PrimeFaces Schedule …

LE LIVRABLE

Le livrable doit correspondre au code source de l'application (compilable et déployable) avec les différents scripts d'automatisation (s'ils existent). Le livrable doit être indépendant de l'environnement ou de l'IDE (pas de fichiers « settings » ou de configuration)

Note : Merci de s'assurer que le contenu du target (Maven) est vide.

L'EVALUATION DU LIVRABLE

- Le « Must » :
  - Le module Patient
- Le « Should » :
  - Le module Médecin
- Le « Could » :
  - Ergonomie
  - Responsivité
  - Spring Boot
- Le « would » :
  - Les Tests Unitaires
  - Notifications ou emails
  - Angular

Note : Le code source sera également évalué sur la base du respect de l'architecture technique proposée, l'utilisation des Design Patterns et les bonnes pratiques de développement.