ORION

Justification des choix techniques
Projet MDD

Auteur : MERCIER Lucas
Version 0.0.1

Table des matières

Aperçu / Synthèse

Pour réaliser ce projet, qui doit être dans un premier temps un MVP, nous devons nous garder en tête des objectifs de simplicité, de rapidité, de faible coût, tout en restant maintenable et évolutif afin que cette base puisse servir aux potentiels développements d'évolutions.

Afin de respecter ces objectifs, nous développerons une application sur la base d'un architecture client-serveur. Le client sera basé sur le framework Angular et le serveur sur le framework Spring.

Le contexte d'un MVP fait que nous n'avons pas le besoin d'utiliser une architecture orientée service ou évènement car ceci embarquera une complexité non nécessaire au peu de fonctionnalités prévues dans ce MVP.

Choix techniques

Choix de l'architecture technique

choix technique	lien vers le site / la documentation / une ressource	but du choix
Architecture orienté client-serveur	*https://openclassrooms.com/fr/courses/7210131-definissez-votre-architecture-logicielle-grace-aux-standards-reconnus/7370196-apprenez-larchitecture-client-serveur*	Séparation des responsabilités

L'architecture client-serveur contribue à rendre notre application maintenable en séparant les responsabilités. La couche de présentation se situera au niveau du client tandis que la couche business se situera sur le serveur.
Cette architecture nous permet également de porter notre application sur plusieurs supports clients différents sans avoir à développer à nouveau une nouvelle couche logique en se greffant directement sur notre serveur.

Choix des librairies backend

choix technique	lien vers le site / la documentation / une ressource	but du choix
Spring framework	*https://spring.io/projects/spring-framework*	Contrôle exécution application serveur
Spring boot	*https://spring.io/projects/spring-boot*	Configuration simplifiée
Spring starter security	*https://spring.io/projects/spring-security*	Configuration et sécurisation des données
Spring starter web	*https://docs.spring.io/spring-boot/reference/web/index.html*	Configuration API REST
Lombok	*https://projectlombok.org/*	Génération de code
H2	*https://www.h2database.com/html/main.html*	Base de données de développement

Spring Framework :
Spring Framework nous permettra de contrôler le flot d'exécution de notre application. C'est lui qui joue le rôle de contrôleur frontal, il réceptionne les requête http et les redistribue sur nos contrôleurs. C'est un framework qui existe depuis de nombreuses années, majoritairement utilisé sur les projets Java, et qui offre donc un support de la part de la communauté de développeur très conséquent.
De nombreuses problématiques ont déjà été posées et résolues.

Spring Boot :
Cette librairie nous permet de simplifier la configuration de notre serveur puisqu'elle propose de nombreuses configurations par défaut.

Spring Starter Security :
Cette librairie, disponible grâce à la définition de notre projet comme étant un enfant de Spring Boot, nous offre une configuration par défaut de Spring Security qui va sécuriser pour nous notre application.
On aura désormais une sécurisation sur les endpoints de nos contrôleurs, un système d'authentification intégré ou encore un système de contrôle de permissions.

Spring Starter Web :
Cette librairie, disponible grâce à la définition de notre projet comme étant un enfant de Spring Boot, nous offre une configuration par défaut de Spring Web qui va nous permettre de définir notre application comme une application web. Grâce à cette librairie nous disposons de nombreuses fonctionnalités permettant de développer une API basée sur le protocole de communication REST.

Lombok :
Cette librairie, qui fonctionne grâce à l'annotation processing, une technique permettant de générer du code à la compilation, nous permet de réduire le nombre de tache répétitif dans nos développements.
Elle permet par exemple de générer des constructeurs, des getters / setters, des fonctions de la classe Object et bien d'autres fonctionnalités.

Grâce à ces librairies nous répondons à des besoins de rapidité, évolutivité, maintenabilité et faibles coûts. Toutes ces librairies nous permettent de développer beaucoup plus rapidement. D'autant plus que celles-ci sont toujours maintenues à jour et également très utilisées dans ce type de projet.

Choix des librairies frontend

choix technique	lien vers le site / la documentation / une ressource	but du choix
Angular	*https://angular.dev/*	Contrôle exécution application client
Primeng	*https://primeng.org/*	Librairie de composants visuels
Primeflex	*https://primeflex.org/*	Librairie de style
primeicons	*https://primeng.org/*	Librairie d'icônes

Angular :
Angular est un framwork qui nous permet de développer des applications web de manière fluide. Ce framework est orienté composant, il favorise la mutualisation et la réutilisation des couches visuelles.

Primeng :
Plusieurs librairies ont été analysées dans le cadre de cette sélection.
Angular Material est la librairie majoritairement utilisée dans les projets Angular. C'est d'ailleurs le choix qu'avait fait Heidi.
En termes de coûts, Angular Material ne couvre pas toutes les fonctionnalités d'une application à l'instar de primeng.
Primeng est la librairie de composants visuels offrant le plus de fonctionnalités. Elle est également très populaire avec plus de 400k téléchargement par semaine. Grâce à cette popularité, de nombreuse documentation et une grande entraide sont disponibles.
Primeng est également compatible avec d'autres librairies de composants visuels complémentaire.

Primeflex :
Primeflex est une librairie de composants visuels développée par les mêmes développeurs que primeng. Celles-ci sont donc compatibles entre-elles. Primeflex nous apporte de nombreuses classes utilitaires pour styliser nos composants. Ceci nous évite d'avoir à créer nos propres styles et nous permet donc d'augmenter notre rapidité sur les développements de composants visuels.

Primeicons :
Primeicons est également une librairie développée par les développeurs de primeng. Celle-ci nous apporte de nombreux icones.

Ces trois librairies de primeng semblent être les librairies qui ont été utilisées sur la maquette de notre MVP.