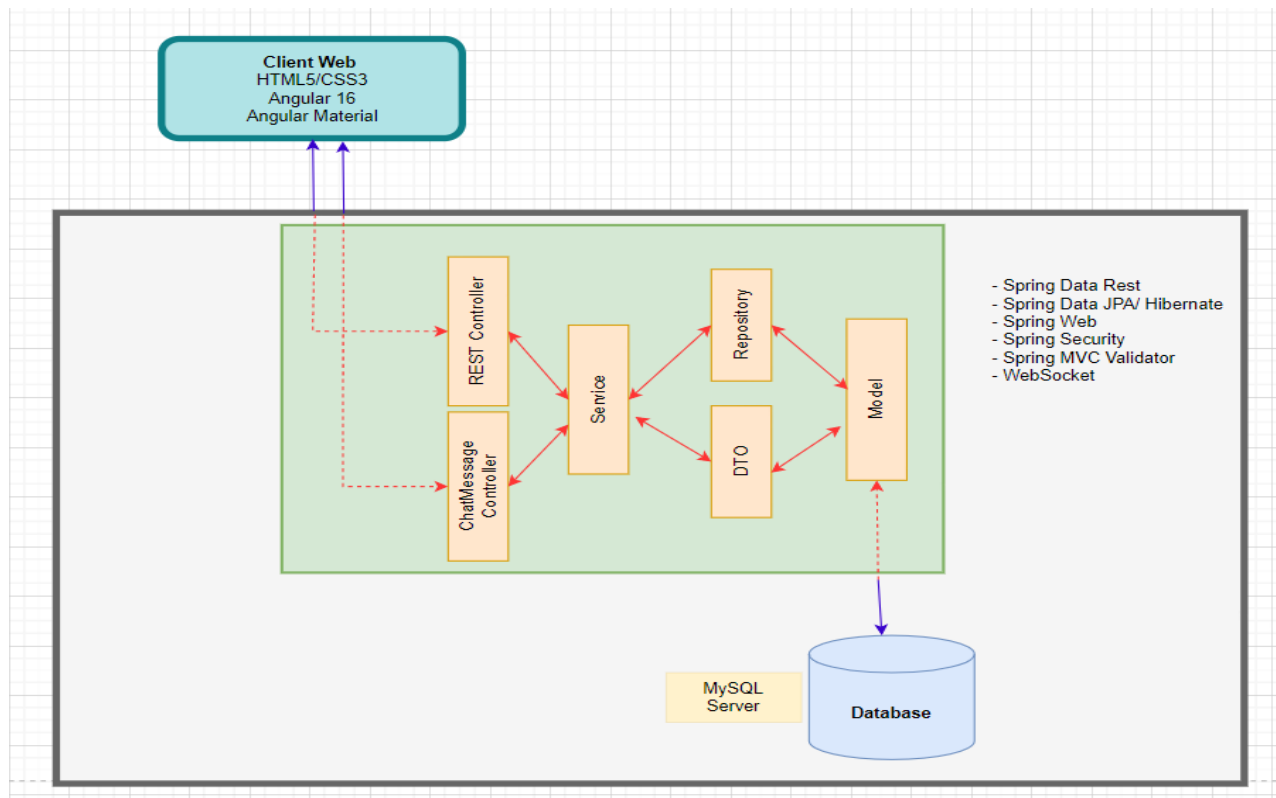


Description générale de l'architecture

L'architecture en couche est défini de la manière suivante :

- *Couche Présentation dite Client Web* : interface utilisateur qui permet d'afficher des informations et de les collecter.
- *Couche REST Controller* : gestion des interactions entre l'utilisateur de l'application et l'application avec l'usage des API REST.
- *Couche ChatMessage Controller* : canal de communication entre le navigateur et le serveur avec l'usage de l'API WebSocket.
- *Couche Service* : implémentation des traitements métiers.
- *Couche DTO* objet léger utilisé pour transférer des données entre les couches d'une application.
- *Couche Repository* : interaction avec les sources de données externes.
- *Couche Model* : implémentation des objets métiers.



Liste de contrôle de l'architecture terminée

Composants logiciels

- S'assurer de la communication entre les différents composants au travers d'une API.
- S'assurer que les requêtes HTTP communiquent bien entre les couches présentations et Rest controller.
- S'assurer que le protocole WebSocket assure l'ouverture d'un canal de communication entre le navigateur et le serveur au travers de l'API WebSocket.
- S'assurer que le tchat inclus la réception et de l'envoi de message entre les utilisateurs (client et support).
- En cas d'erreurs, s'assurer de la génération d'exception pour capturer et rectifier les erreurs.
- Vérifier que toutes les dépendances nécessaires pour le développement de l'application soient bien intégrées avec les versions récentes.
- Une API doit être définie afin de rendre le site multilingue, à minima français et anglais.

Services ou composants tiers logiciels

- S'assurer de la disponibilité du service lié au paiement.
- Vérifier l'intégration de ce service à l'architecture de l'application, notamment au travers de la configuration de celui-ci.
- S'assurer de la communication de ce service avec les autres composants logiciels de l'application.
- Vérifier que les informations de paiement soient bien transmises au travers d'une API Rest.

Gestion des données

- S'assurer que les données parcourent les composants de l'application, de la base de données jusqu'au client.
- S'assurer de la mise en place de méthodes afin de manipuler les données. Les opérations CRUD standard sont requises pour chaque domaine.
- Vérifier que les données manipulées correspondent bien aux attendus de l'application (gestion utilisateur, gestion offre de location, gestion réservation).
- S'assurer de l'accès aux données au travers d'une API soit possible pour la consultation et la modification des données traitées.

Infrastructure

- S'assurer que le matériel mis en place est récent pour assurer le bon fonctionnement de l'architecture tout en préservant la performance et la maintenabilité de la future application.
- Vérifier que la dernière version des systèmes d'exploitations soit utilisée.
- Respecter les technologies mentionnées dans le document "Architecture definition document", avec les versions respectives exception faite si une version récente est sortie.
- Utiliser un outil de déploiement simple, adapté avec une actualisation régulière de l'outil pour assurer la performance de l'application.

Sécurité

- Mettre en place des pratiques de codage sûres avec la validation des entrées, le codage des sorties, et les requêtes paramétrées afin d'éviter toutes instructions SQL nuisibles, avec notamment le hachage du mot de passe et un mot de passe renforcé.
- Utiliser le chiffrement sécurisé des données avec le protocole SSL/TLS afin de crypter toute communication entre le serveur et le navigateur.
- Définir des rôles dans l'application (utilisateur, admin), bien que l'application soit à destination du client.
- Mettre en place dans le développement de l'application, un service permettant de vérifier l'authentification de l'utilisateur avec notamment l'usage du token avec JWT.

