

Spécifications Techniques - HomeSkolar

1. Architecture générale

1.1. Type d'application

- **Architecture** : Application web avec architecture client-serveur
 - Application web accessible via navigateur pour faciliter l'accès aux utilisateurs
 - Pas d'installation requise, compatible tous supports
 - Interface responsive pour s'adapter à tous les écrans

1.2. Technologies Frontend

- **Framework** : React 18
 - Choisi pour sa grande communauté et sa documentation complète
 - Facilite la création d'interfaces interactives
 - Permet un développement rapide avec ses nombreux composants
- **Bibliothèques complémentaires** :
 - React Router : Navigation entre les pagesv
 - React Big Calendar : Gestion du calendrier
 - Bootstrap : Design responsive

1.3. Technologies Backend

- **Framework** : Spring Boot
 - Choisi pour sa robustesse et sa facilité d'utilisation avec Java
 - Gestion intégrée de la sécurité
 - Excellente intégration avec les bases de données
- **Base de données** : MySQL
 - Solution stable et éprouvée
 - Facile à maintenir
 - Adapté au volume de données attendu

2. Solutions techniques par fonctionnalité

2.1. Authentification

- Système d'authentification Spring Security
- Gestion des sessions utilisateurs
- Hachage sécurisé des mots de passe
- Protection contre les attaques basiques (injection SQL, XSS)

2.2. Communication

- API REST pour les échanges client/serveur
- Stockage des messages dans MySQL
- Système simple de notifications en temps réel
- Possibilité d'épingler les messages importants

2.3. Calendrier

- Interface utilisateur avec React Big Calendar
- Synchronisation via API REST
- Données stockées dans MySQL
- Système de notifications pour les rappels

2.4. Gestion des tâches

- API REST pour la création/modification des tâches
- Stockage dans MySQL avec statuts (à faire, en cours, terminé)
- Notifications automatiques pour les échéances
- Association possible avec les rendez-vous du calendrier

3. Sécurité

- HTTPS obligatoire
- Authentification sécurisée
- Protection des données utilisateurs
- Validation des données côté serveur et client

4. Performance

- Optimisation des requêtes base de données
- Mise en cache des données fréquemment utilisées
- Chargement optimisé des ressources frontend
- Pagination des listes (messages, tâches)