



CodeIguanas

CAHIER DES CHARGES

APPLICATION HOMESKOLAR

SPECIFICATIONS FONCTIONNELLES

1. GESTION DES COMPTES UTILISATEUR

- **Inscription** : L'application doit permettre aux utilisateurs (élèves et tuteurs) de créer un compte personnel.
- **Connexion** : Les utilisateurs pourront se connecter à leur compte pour accéder aux fonctionnalités de la plateforme.
- **Gestion du mot de passe** : Les utilisateurs pourront réinitialiser ou modifier leur mot de passe en cas de besoin, pour garantir la sécurité de leur compte.
- **Gestion des informations personnelles** : Les utilisateurs doivent pouvoir modifier leurs informations personnelles.

BESOINS COUVERTS :

Cette fonctionnalité répond au besoin fondamental d'identification des utilisateurs et de sécurisation de leurs données, facilitant ainsi un accès personnalisé aux services de HomeSkolar.

2. SYSTEME DE COMMUNICATION

- **Messagerie intégrée** : L'application intégrera un système de messagerie permettant aux élèves et aux tuteurs d'échanger des messages en lien avec les sessions de soutien scolaire et les devoirs.
- **Épinglage de messages** : Les élèves pourront épingler certains messages importants pour les retrouver facilement lorsqu'ils en auront besoin.
- **Notifications de messages non lus** : Les utilisateurs recevront une notification pour tout message non lu afin de ne rater aucune information importante.

BESOINS COUVERTS :

La messagerie répond au besoin de communication continue entre élèves et tuteurs, garantissant une interaction simplifiée et une meilleure organisation des rendez-vous et tâches.

3. PLANIFICATION DES RENDEZ-VOUS ELEVE-TUTEUR

- **Calendrier** : L'application proposera un calendrier permettant aux utilisateurs de consulter et de gérer leurs rendez-vous. Chaque utilisateur pourra visualiser l'ensemble de ses événements planifiés.
- **Planification des sessions hebdomadaires** : Le système de calendrier permettra aux élèves et aux tuteurs de fixer des rendez-vous.

BESOINS COUVERTS :

Cette fonctionnalité répond au besoin de planification des rencontres, permettant aux élèves de bénéficier d'un accompagnement organisé, et aux tuteurs de gérer facilement leur emploi du temps avec les élèves.

4. GESTION DES TACHES ET SUIVI

- **Tâches post-rencontre** : À l'issue de chaque rencontre, les élèves recevront une liste de tâches à accomplir avant la prochaine session. Cela comprend les devoirs ou exercices recommandés par le tuteur.
- **Création de tâches à réaliser** : A l'issue de la rencontre, le tuteur crée une liste de tâches à accomplir par l'élève pour la prochaine session.
- **Création de tâches personnelles** : Les utilisateurs (élèves et tuteurs) pourront créer des tâches ou des mémos personnels pour organiser leur travail.
- **Consultation de tâches à accomplir** : Les utilisateurs pourront consulter leurs tâches.
- **Notifications de rappel pour les tâches** : Les élèves recevront des rappels pour les tâches assignées afin de les aider à se préparer pour la prochaine rencontre avec leur tuteur.

BESOINS COUVERTS :

Cette fonctionnalité permet de suivre le travail entre les sessions, favorisant ainsi un apprentissage continu et structuré pour l'élève. Elle permet aussi aux tuteurs de s'assurer que l'élève progresse entre chaque rendez-vous.

RESUME DES FONCTIONNALITES ET OBJECTIFS

Ces fonctionnalités répondent directement aux besoins exprimés par HomeSkolar. En permettant la gestion des comptes, une communication simplifiée, une planification efficace des rendez-vous et un suivi des tâches, l'application offrira une expérience cohérente et complète pour atteindre l'objectif de soutien scolaire à distance. Ces spécifications permettent de concevoir une application intuitive, parfaitement alignée avec les besoins des élèves et des tuteurs, assurant ainsi une expérience utilisateur optimisée pour un soutien scolaire efficace.

Pour la mise en œuvre de l'application HomeSkolar, nous avons mené une veille technologique approfondie pour identifier les technologies adaptées, fiables et maintenables. Voici trois sources surveillées et leur intérêt :

1. DOCUMENTATION OFFICIELLE DE SPRING BOOT

SOURCE : <https://spring.io/>

INTERET :

Spring Boot est une solution mature pour le développement back-end avec Java. La documentation officielle fournit des guides détaillés pour configurer des applications scalables et modulaires. Elle permet également d'explorer les nombreuses fonctionnalités natives comme l'intégration facile avec des bases de données ou des services REST.

Rôle dans le projet : Nous surveillons cette ressource pour garantir une mise en œuvre efficace des besoins back-end (gestion des utilisateurs, planification des tâches, etc.).

2. DOCUMENTATION OFFICIELLE DE REACT.JS

SOURCE : <https://reactjs.org/>

INTERET :

React.js est un framework front-end moderne qui facilite le développement d'interfaces utilisateur dynamiques et performantes. La documentation est exhaustive, avec des exemples pratiques et une communauté active qui publie régulièrement des mises à jour.

Rôle dans le projet : Nous consultons cette ressource pour concevoir une interface utilisateur réactive et ergonomique, répondant aux besoins d'accessibilité des élèves et des tuteurs.

3. DOCUMENTATION MARIADB

SOURCE : <https://mariadb.com/kb/en/documentation/>

INTERET :

MariaDB est un système de gestion de bases de données relationnel open-source, réputé pour sa compatibilité avec MySQL, ses performances élevées, et sa flexibilité. La documentation officielle fournit des guides détaillés pour la configuration, l'utilisation des requêtes SQL et la gestion des relations complexes entre les données.

Rôle dans le projet : Cette source nous permet de garantir une gestion efficace des données critiques (telles que les utilisateurs, les rendez-vous, et les messages) tout en bénéficiant d'une solution moderne, optimisée et performante. MariaDB est également parfaitement compatible avec Spring Boot, assurant une intégration fluide dans l'architecture technique du projet.

SPECIFICATIONS TECHNIQUES

TECHNOLOGIES UTILISEES

1. FRONT-END : REACT.JS + HTML + CSS

RAISON DU CHOIX :

React.js a été choisi pour sa simplicité, sa flexibilité et ses performances. Comparé à des alternatives comme Angular (plus complexe) ou Vue.js (moins documenté pour les grands projets), React.js offre un compromis parfait entre puissance et facilité de prise en main. Il dispose également d'un écosystème riche avec des bibliothèques compatibles pour la gestion de l'état ou l'internationalisation.

RÔLE DANS LE PROJET :

React.js sera utilisé pour développer une interface utilisateur réactive et intuitive. Cela inclut les fonctionnalités de planification (calendriers interactifs), de messagerie et de gestion de tâches.

2. BACK-END : SPRING BOOT (JAVA)

RAISON DU CHOIX :

Spring Boot est une solution éprouvée pour la création d'API REST robustes et scalables. Par rapport à Node.js, qui excelle dans les applications légères, Spring Boot offre une meilleure gestion des transactions complexes et une sécurité intégrée (comme Spring Security).

ROLE DANS LE PROJET :

Le back-end sera responsable de la gestion des utilisateurs, des tâches, des notifications et des rendez-vous. Spring Boot permet de centraliser ces fonctionnalités tout en restant compatible avec PostgreSQL pour la gestion des données.

3. BASE DE DONNEES : MARIADB

RAISON DU CHOIX :

MariaDB a été choisi pour sa compatibilité avec MySQL, tout en offrant des améliorations en termes de performances et de sécurité. Sa nature open-source garantit une flexibilité accrue et une indépendance vis-à-vis des fournisseurs propriétaires. MariaDB est optimisé pour les applications nécessitant des opérations fréquentes de lecture et d'écriture, tout en bénéficiant d'une large communauté active et d'une documentation riche. De plus, son intégration avec Spring Boot via Spring Data JPA est simple et efficace, ce qui facilite la gestion des données relationnelles et accélère le développement.

ROLE DANS LE PROJET :

MariaDB assurera la persistance des données critiques telles que les informations des utilisateurs, les rendez-vous, les tâches et les messages. Il gèrera efficacement les relations complexes entre les entités (par exemple, entre tuteurs, élèves, calendriers et rendez-vous) et garantira des requêtes

rapides pour une expérience utilisateur fluide, tout en offrant une solution robuste et moderne pour la base de données.

DIAGRAMME DE CLASSES

