



CodeIguanas

Cahier des charges

Application HomeSkolar

Spécifications fonctionnelles

Les fonctionnalités clés de notre application comprennent :

1. Authentification et Gestion de Compte

- Inscription et connexion facilitées pour les élèves et les tuteurs, avec demande d'informations de base pour la création de compte.
- Mise en place d'une authentification à deux facteurs pour renforcer la sécurité des comptes utilisateurs.
- Possibilité pour les utilisateurs de modifier leurs informations personnelles, y compris le mot de passe, pour tenir leur compte à jour, en veillant à l'accessibilité pour tous les utilisateurs.

2. Système de Communication

- Messagerie intégrée permettant les échanges directs entre élève et tuteur, incluant la possibilité d'épingler les messages importants.
- Notifications pour les nouveaux messages non lus, garantissant l'attention aux communications essentielles.
- Introduction d'un filtrage des messages pour une gestion optimale et un support pour l'échange de fichiers multimédias, enrichissant ainsi la communication.

3. Planification des Rencontres

- Calendrier interactif pour visualiser et gérer les rendez-vous hebdomadaires, avec options pour ajouter, modifier, et annuler des rencontres.
- Intégration d'alertes pour rappeler les rencontres à venir, favorisant la ponctualité.

- Intégration avec des calendriers externes (Google Calendar, Microsoft Outlook, Full Calendar, Notion etc.) et gestion automatique des fuseaux horaires pour faciliter l'organisation des rencontres entre élèves et tuteurs de différentes zones géographiques.

4. Suivi des Tâches

- Fonctionnalité permettant aux tuteurs de créer et suivre des tâches spécifiques pour les élèves, avec des options de feedback constructif.
 - Possibilité pour les élèves et les tuteurs de noter des tâches personnelles ou des mémos, avec des rappels personnalisables pour une meilleure organisation personnelle.
 - Notifications améliorées pour une visibilité accrue des tâches à accomplir, soutenant efficacement les objectifs d'apprentissage.
-

Veille Technologique pour HomeSkolar

Introduction

Le développement de HomeSkolar requiert une sélection technologique réfléchie pour garantir robustesse, sécurité et une expérience utilisateur optimale. Notre évaluation, centrée sur des frameworks modernes, a abouti à une architecture combinant [Django](#) pour le back-end, [Vue.js](#) pour le front-end, et [PostgreSQL](#) pour la gestion des données. Ce choix est le résultat d'une comparaison minutieuse avec d'autres technologies, appuyée par une analyse approfondie de leurs avantages et limites. Nous avons également pris en considération notre environnement de programmation existant et les compétences de notre équipe, assurant ainsi une adéquation parfaite avec les besoins du projet.

Frameworks Back-end Évalués

Dans notre quête pour le back-end idéal, nous avons scruté **Django (Python)**, **Laravel (PHP)**, **Rails (Ruby)**, et **Express.js (Node.js)** à travers plusieurs prismes : rapidité de développement, sécurité, écosystème et communauté, performances, ainsi que la facilité d'intégration avec les technologies front-end contemporaines.

Django s'est imposé comme le choix optimal pour notre back-end, grâce à sa philosophie 'batteries incluses'. Cette approche fournit d'emblée une suite complète de fonctionnalités, notamment une sécurité renforcée et une architecture favorisant un développement agile et performant. Comparé à Laravel et Rails, qui offrent eux aussi une excellente vitesse de développement, Django se distingue particulièrement par sa solidité en termes de sécurité et sa

[bonne intégration avec les frameworks front-end, comme Vue.js](#). Ces atouts, combinés à une vaste communauté et à une documentation exhaustive et [multilingues](#), ont renforcé notre conviction qu'il s'agit du choix le plus judicieux pour notre infrastructure back-end.

Nous avons également évalué Express.js, attirés par l'uniformité linguistique qu'offre l'utilisation de JavaScript sur l'ensemble du stack et par sa grande flexibilité. Cependant, après réflexion, nous avons décidé de ne pas le choisir pour HomeSkolar. Bien que la possibilité de personnaliser la sécurité et la liberté structurelle d'Express.js puissent être perçues comme des avantages, [elles exigent en contrepartie un investissement significatif en termes de développement et de maintenance pour garantir la sécurité et la cohérence de l'architecture](#). Ces exigences pourraient rendre plus complexe la gestion à long terme de notre plateforme, en particulier vis-à-vis de nos impératifs de créer un espace sécurisé et aisément évolutif pour les utilisateurs de HomeSkolar.

Choix de Vue.js pour le front-end

[Vue.js a été préféré à Angular](#) et [React](#) en raison de sa légèreté, sa simplicité d'intégration, et sa courbe d'apprentissage douce. Sa modularité et sa performance assurent une expérience utilisateur réactive. La compatibilité avec Django permet une architecture claire et une maintenance simplifiée, facilitant le développement de fonctionnalités complexes.

PostgreSQL : La Base de Données de Choix

Pour la gestion des données, PostgreSQL a été choisi pour sa robustesse, sa conformité aux standards SQL, et ses capacités avancées, comme le support des données géospatiales. Sa compatibilité naturelle avec Django via l'ORM (Object-Relational Mapping) facilite la modélisation et manipulation des données, offrant une fondation solide pour la scalabilité de la plateforme.

Intégration de Technologies Complémentaires

Afin d'enrichir l'expérience utilisateur et de soutenir une communication en temps réel, l'intégration de [WebSocket avec Django Channels](#) est prévue, permettant des interactions instantanées via une communication bidirectionnelle. [FullCalendar](#) sera utilisé pour la gestion des rencontres et tâches, offrant une interface riche pour la planification d'événements, facilement intégrable avec Vue.js et les données gérées par Django.

Défis et Solutions avec les Technologies Choisies

Malgré les nombreux avantages offerts par Django, Vue.js, et PostgreSQL, nous sommes conscients des défis potentiels associés à ces technologies.

Avec Vue.js, la complexité de gestion des états dans des applications larges peut représenter un défi. Pour y remédier, l'adoption de [Vuex](#) nous permettra une gestion d'état centralisée et simplifiée, assurant ainsi une réactivité et une performance optimales, même dans des scénarios d'utilisation intensifs.

Pour Django, tout en profitant de sa robustesse en matière de sécurité, nous resterons attentifs à l'application des meilleures pratiques de développement sécurisé. Cela inclut des audits de

sécurité réguliers et la mise à jour constante des dépendances pour parer aux vulnérabilités émergentes. Concernant l'utilisation de l'ORM de Django, nous sommes conscients des défis potentiels, comme les performances suboptimales pour certaines requêtes complexes. Pour y faire face, nous prioriserons la formation sur les meilleures pratiques ORM et l'optimisation des requêtes.

Quant à PostgreSQL, sa complexité et sa puissance requièrent une expertise spécifique pour optimiser les schémas de base de données et les requêtes, afin de garantir les meilleures performances. Notre équipe s'engage à suivre une formation continue et à partager les meilleures pratiques en interne pour maîtriser ces aspects.

En anticipant ces défis et en mettant en place des stratégies pour les surmonter, nous assurons la création d'une plateforme HomeSkolar non seulement robuste et sécurisée mais également capable de fournir une expérience utilisateur sans faille.

Conclusion

Ces choix - notre trio technologique et ses compléments - sont dictés par la volonté de fournir une plateforme évolutive, sécurisée et engageante, capable de s'adapter aux besoins futurs et de soutenir efficacement l'apprentissage des élèves. En adoptant une approche basée sur des technologies éprouvées et flexibles, HomeSkolar est prêt à devenir un acteur majeur dans le soutien scolaire en ligne.

Liens docs :

django doc : <https://docs.djangoproject.com/fr/5.0/>

Vues.js : <https://fr.vuejs.org/guide/introduction.html>

PostgreSQL : <https://www.postgresql.org/docs/>

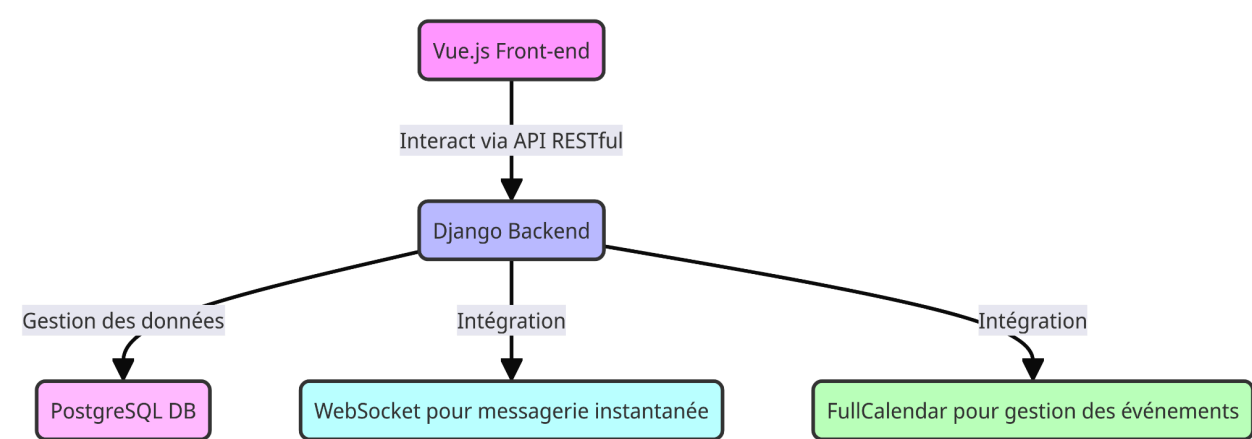
WebSocket django Channels : <https://github.com/django/channels>

Full Calendar : <https://fullcalendar.io/>

Spécifications techniques

Cette section offre une vue d'ensemble des spécifications techniques, détaillant les technologies sélectionnées pour le front-end, le back-end, la gestion des données, ainsi que l'intégration de services tiers. Ces choix sont stratégiquement alignés avec les besoins fonctionnels de HomeSkolar, visant à établir une architecture à la fois cohérente et performante.

Architecture Système de l'Application Web HomeSkolar



Vue d'ensemble des interactions entre l'utilisateur, le front-end, le back-end, la base de données, et les intégrations de services tiers

Spécifications Techniques Consolidées pour le Projet HomeSkolar

Front-end : Vue.js

- Rôle pour le Projet :

- **Interface Utilisateur** : Crée des interfaces pour l'inscription, la connexion et le tableau de bord utilisateur, enrichies de composants dynamiques pour la messagerie et la planification des sessions.
- **Interactivité** : Garantit une interaction utilisateur fluide, avec des notifications en temps réel via WebSocket et une gestion efficace des événements grâce à FullCalendar.

Back-end : Django

- **Rôle pour le Projet** :
 - **Authentification des Utilisateurs** : Utilise le système d'authentification de Django pour sécuriser l'inscription et la connexion.
 - **Logique Métier** : Implémente la logique nécessaire à la mise en relation des élèves et des tuteurs, ainsi qu'à la gestion des sessions de tutorat.
 - **Interaction avec la Base de Données** : Gère la communication avec PostgreSQL pour toutes les opérations de données et fournit une interface pour les API RESTful.

Base de Données : PostgreSQL

- **Rôle pour le Projet** :
 - **Stockage des Données** : Conserve de manière sécurisée toutes les informations cruciales, telles que les données utilisateurs, les profils des tuteurs et les historiques des sessions.
 - **Gestion des Tâches** : Facilite la création, la visualisation et le suivi des tâches et événements, intégrant étroitement FullCalendar pour une gestion transparente.

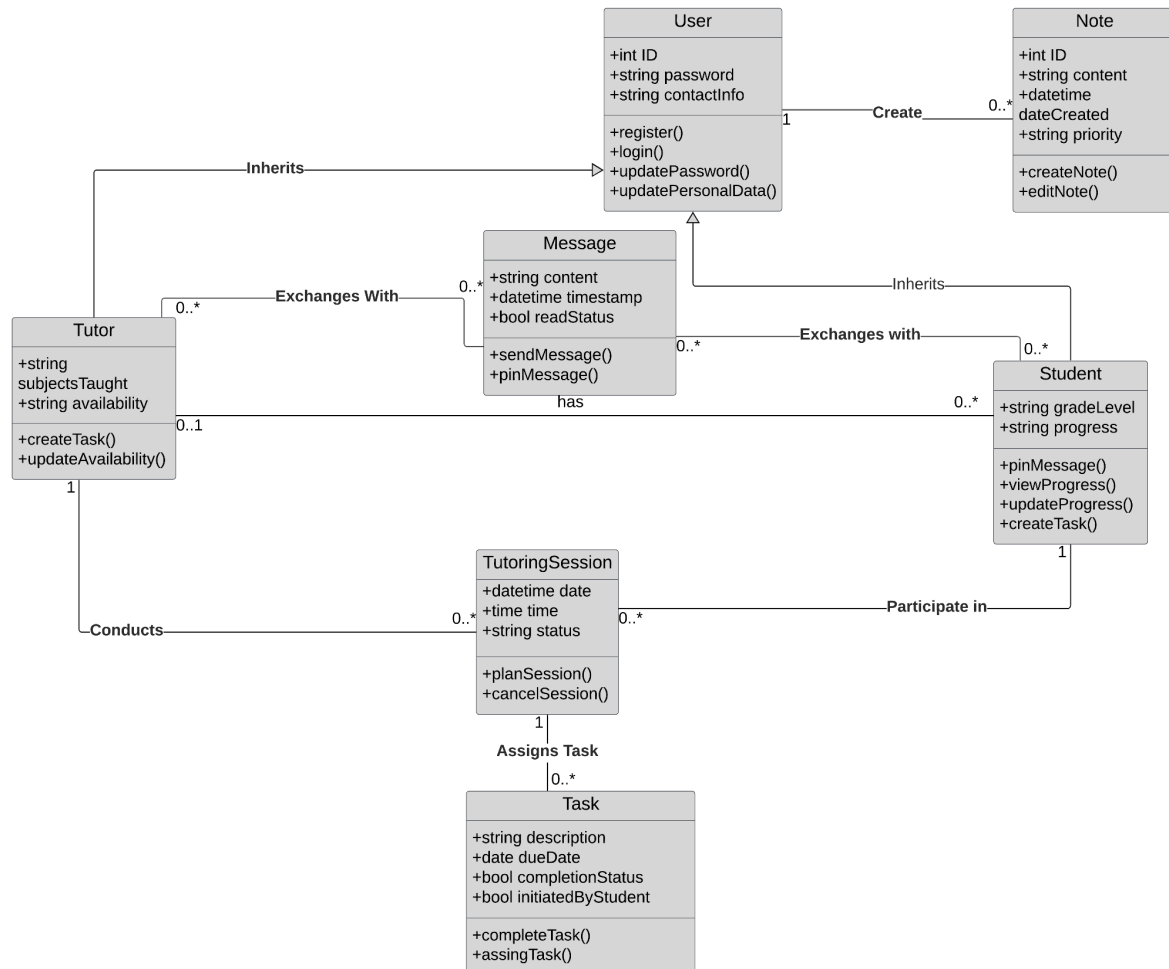
Communication et Notifications

- **Intégration avec WebSocket** : Permet une communication en temps réel entre élèves et tuteurs, dépassant la simple messagerie pour englober une interaction web dynamique.
- **Intégration avec FullCalendar** : Propose une interface utilisateur intuitive pour organiser les sessions de tutorat et visualiser les événements, améliorant la planification et l'engagement.

Alignement des Choix Technologiques avec les Besoins Fonctionnels

- **Authentification et Gestion des Utilisateurs** : Django assure une authentification solide et une gestion sécurisée des comptes, répondant aux normes élevées de protection des données.
- **Communication Instantanée** : La combinaison de Vue.js et WebSocket enrichit l'expérience utilisateur grâce à une communication instantanée, essentielle pour l'interaction directe entre élèves et tuteurs.
- **Planification et Gestion des Sessions** : L'intégration de Vue.js avec FullCalendar simplifie la planification et la visualisation des sessions, facilitant l'organisation et augmentant l'engagement.
- **Coordination des Tâches** : L'écosystème formé par Vue.js, Django, et PostgreSQL offre une gestion fluide et intégrée des tâches, de leur création à leur suivi, assurant une expérience utilisateur cohérente et sans accroc.

Diagramme de classes



Description

Classe "User" (Utilisateur) :

- **Attributs** : Chaque utilisateur a un identifiant, un mot de passe et des informations de contact.
- **Méthodes** : Un utilisateur peut s'inscrire, se connecter, changer son mot de passe et ses infos personnelles.
- **Relations** :
 - "Crée" des "Note".
 - "Crée" des "Message".

Classe "Note" :

- **Attributs** : Une note a un identifiant, un contenu, une date et heure, et une priorité.
- **Méthodes** : On peut créer et éditer une note.
- **Relations** :
 - Est une extension de "User".
 - Peut échanger des "Message" avec "Student".

Classe "Message" :

- **Attributs** : Un message contient du texte, une date et heure d'envoi, et un indicateur de lecture.
- **Méthodes** : On peut envoyer un message ou l'épingler.
- **Relations** :
 - Peut être envoyé entre "Tutor" et "Student".

Classe "Tutor" (Tuteur) :

- **Attributs** : Un tuteur a des matières enseignées et des créneaux de disponibilité.
- **Méthodes** : Un tuteur peut créer des "Task" et mettre à jour sa disponibilité.
- **Relations** :
 - Peut envoyer et recevoir des "Message" de "User".
 - Dirige des "TutoringSession".
 - Donne des "Task" aux "Student".

Classe "Student" (Étudiant) :

- **Attributs** : Un étudiant a un niveau scolaire et un suivi de progression.
- **Méthodes** : Un étudiant peut épingler un message, suivre sa progression, l'actualiser et créer des "Task".
- **Relations** :
 - Est une version spécialisée de "User".
 - Peut interagir avec des "Note" via des "Message".
 - Prend part aux "TutoringSession".

Classe "Task" (Tâche) :

- **Attributs** : Une tâche à une description, une date limite, un statut de complétion et peut être initiée par un étudiant.
- **Méthodes** : On peut finir une tâche ou l'assigner.
- **Relations** :
 - Est donnée par un "Tutor" à un "Student".
 - Est liée à "TutoringSession" où elle peut être créée.

Classe "TutoringSession" (Session de Tutorat) :

- **Attributs** : Une session a une date, une heure et un statut.
- **Méthodes** : On peut planifier ou annuler une session.
- **Relations** :
 - Est animée par un "Tutor".
 - Est suivie par un "Student".

- Est associée à l'assignation des "Task".

Chaque "User" peut être soit un "Tutor" soit un "Student", et chacun interagit à travers des "Message". Les "TutoringSession" sont des moments clés où les "Tutor" et les "Student" se rencontrent et où les tâches sont attribuées pour approfondir l'apprentissage.