



CodeIguanas

Cahier des charges

Application HomeSkolar

Spécifications fonctionnelles

Présentez ce que le produit va réaliser. Quelles tâches l'utilisateur pourra-t-il faire (soit élève, soit tuteur) ? Assurez-vous de les découper en sous-fonctionnalités.

A. CRÉATION ET CONNEXION

1) Création

Les utilisateurs pourront se créer un compte par mail et mot de passe sur l'interface d'inscription en remplissant les champs pré définis. Nous devons justifier d'une connexion sécurisée. Seule différence, C'est au moment de la création du compte qu'on définit si nous sommes un élève ou un tuteur. il faudra que tous les champs soient correctement remplis pour que l'enregistrement réussisse. les champs pré définis seront :

- nom et prénom
- date de naissance
- élèves ou tuteur
- niveau recherché pour les élèves ou niveau exercés pour les tuteurs.
- Une photo de profil pourra être ajoutée mais est optionnel (avatar si aucune photo définis)

2) Connexion

Les utilisateurs pourront se connecter par mail et mot de passe afin de pouvoir accéder à leur espace personnel

B. GESTION DES UTILISATEURS

1) Gestion des mots de passe

Chaque utilisateur pourra modifier son mot de passe via le compte ou via une vérification par mail (mot de passe oublié).

2) Modification du compte utilisateur

Un utilisateur pourra aussi modifier leur donnée personnelle via leurs comptes. Pour le moment, ce sera les photos de profil, ainsi que le niveau recherché pour les élèves.

C. COMMUNICATION

1) Messagerie entre élève et tuteur

Il faudra une messagerie intégrée pour pouvoir communiquer entre l'élève et son tuteur désigné. Il s'agit d'une simple messagerie, afin de pouvoir échanger des messages textes.

2) Notification de message non lus

Si un message n'a jamais été ouvert une notification sous forme de badge devra apparaître sur l'icône de messagerie de l'utilisateur

3) Épinglé un message

Enfin, si un élève ou tuteur le souhaite, il pourra épingler un message particulier.

D. LA PAGE CALENDRIER

1) Affichage des rendez-vous sous forme de calendrier.

Il faut que cette page affiche un calendrier présentant ses différents rendez-vous. Ce calendrier est disponible pour les élèves et tuteurs.

2) Ajouter ou modifier une rencontre au calendrier

Un élève ou tuteur peut ajouter une rencontre à son calendrier, en définissant un titre, une date, un contenu en option.

3) Annulation d'une rencontre

Un tuteur ou un élève doivent pouvoir annuler/supprimer une rencontre

E. LES TÂCHES

Un visuel sur toutes les tâches à réaliser, classées par date et statut (en cours / terminé).

Une tâche contient un nom et des actions à effectuer. Des dates de début et de fin sont possible mais optionnelle

1) Assignment de tâche en fin de cours

Pendant la session le tuteur a la possibilité de noter les tâches qui seront automatiquement assignées à l'élève. A la fin de chaque cours, pour la semaine d'après, l'élève pourra retrouver ces tâches assignées dans sa liste de tâches et dans son calendrier au moment de la rencontre entre le tuteur et l'élève.

2) Gestion des tâches personnelles

Les élèves et tuteurs ont la possibilité d'ajouter leurs propres tâches par eux-mêmes via un bouton plus par exemple. ces tâches personnelle pourront aussi être mis à jour ou supprimer en cas de besoin

3) Notification

Un badge rouge comme pour l'application sera sur l'icône tâche comme pour la messagerie afin d'indiquer une action en attente ainsi que la tâche non lue.

Veille technologique

Fournir au moins trois sources variées et pertinentes que vous surveillez avec une explication de l'intérêt de chacune. Justifier les choix technologiques retenus dans la spécification technique. Prendre compte des ressources où les technologies se trouvent, par exemple, le site web officiel ou la documentation officielle d'une technologie.

1. Documentation officielle de Spring Boot

<https://docs.spring.io/spring-boot/docs/current/reference/html/>

- Spring Boot est le framework principal utilisé pour le développement back-end de notre projet.
- La documentation officielle fournit des explications détaillées sur l'architecture, les fonctionnalités et les bonnes pratiques.

- Elle est constamment mise à jour, garantissant que nous utilisons les versions les plus stables et sécurisées.
2. **Stack Overflow (Communauté de développeurs)** <https://stackoverflow.com/>
 - Stack Overflow est un forum incontournable où les développeurs partagent des solutions aux problèmes techniques rencontrés.
 - Cette source permet de trouver rapidement des solutions aux défis rencontrés lors du développement.
 3. **Site officiel de Thymeleaf** - <https://www.thymeleaf.org>
 - Cela permet de comprendre les nouvelles fonctionnalités, les concepts clés, ainsi que les meilleures pratiques pour implémenter thymeleaf de manière optimale.
 4. **Medium** - <https://medium.com>
 - Medium fournit des articles, tutoriels et retours d'expérience détaillés sur l'utilisation de Java dans des projets réels, permettant de mieux comprendre comment l'utiliser dans un contexte de développement quotidien et d'explorer des cas pratiques.
 5. **Baeldung** - <https://www.baeldung.com/>
 - Baeldung est aujourd'hui une référence dans le monde du développement logiciel, notamment pour tout ce qui touche à **Java**, **Spring**, **API REST**, **backend**, **sécurité** et **Linux**.
 - Une mise à jour continue et rapide des contenus et tutoriel concret : Baeldung est une ressource incontournable pour tout développeur souhaitant rester à jour sur les évolutions des technologies Java, Spring, et du développement backend en général. Grâce à ses articles pédagogiques, pratiques et actualisés, il me permet de renforcer mes compétences techniques tout en assurant une veille efficace sur les nouveaux standards du secteur.

Spécifications techniques

Présentez les technologies utilisées et comment la technique répond aux besoins du client. Focalisez-vous sur les parties front-end et back-end, mais pas besoin de spécifier un hébergeur ou un outil de déploiement.

Pour chaque technologie, précisez la raison pour laquelle vous l'avez choisie (plutôt qu'une autre) et son rôle pour le projet.

Notre projet repose sur une architecture moderne combinant des technologies performantes

et éprouvées pour répondre aux exigences du client. J'ai fait le choix d'un découpage clair entre le **front-end** et le **back-end pour la comparaison**, garantissant ainsi **modularité**, **scalabilité** et **maintenabilité**.

1) TECHNOLOGIES BACK-END

Langages possibles : JAVA + SPRING, PYTHON + DJANGO, EXPRESS.JS+ NODE.JS.

Nous partons du principe qu'il n'y aura pas de période d'apprentissage, nous aurons des développeurs compétents immédiatement.

Comparatif	Java/Spring	Python/Django	Express/Node
Sécurité	Très Forte	Forte	Moins sécurisé
Performance	Plus Forte	Moins Forte	Forte
Scalabilité	Excellente	Moins adapté	Bonne
Ecosystème	Très Riche	Forte mais moins orienté web	Large
Coût de développement	Plus élevé mais rentable à long terme	Modéré mais pas efficace à long terme	Moins chère à court terme
Utilisation générale	Entreprise / Grande Entreprise	Adapté à des projets moyens.	Petit projet web rapide moins adapté à l'évolution

Choix final du Back end :

Notre projet a des besoins précis, en utilisant Java et Spring Boot, nous pourrions :

1) Sécurité

- Gestion des comptes utilisateurs (élèves et tuteurs) avec des données sensibles : Java et Spring Boot offrent des outils avancés de sécurité (Spring Security).

2) Scalabilité et performance

- Le projet va évoluer (plus d'élèves, plus de tuteurs).
- Spring Boot permet de gérer une charge importante sans compromettre la rapidité.
- Utilisé par les grandes entreprises, donc fiable à long terme.

3) Communication & messagerie

- Un système de messagerie simple, avec notifications.
- Java/Spring offre des solutions robustes (WebSockets, RabbitMQ) pour éviter les

ralentissements sur les messages.

4) Gestion des tâches & calendrier

- La planification des tâches et rendez-vous demande une gestion efficace des événements.
- Spring Boot + une base de données bien optimisée permettra d'assurer une bonne gestion des disponibilités.

5) Expérience des développeurs & longévité du projet

- Java est un langage stable, avec beaucoup de développeurs qualifiés.
- Il est utilisé dans les systèmes critiques (banques, entreprises), ce qui montre sa fiabilité.

2) Base de données (back-end)

Maintenant que nous connaissons une partie de notre technologie Back-end, choisissons notre Base de données :

Critères	PostgreSQL	MySQL	MongoDB (NoSQL)	Firebase
Type	Relationnel (SQL)	Relationnel (SQL)	Documentaire (NoSQL)	NoSQL
Performance	Excellente pour des requêtes complexes	Bonne mais légèrement inférieure à PostgreSQL	Très rapide pour les données non structurées	Ultra rapide mais limité
Scalabilité	Très évolutif	Adapté aux sites web mais moins aux gros systèmes	Très évolutif pour des données massives	Très évolutif (cloud) mais moins contrôlable
Sécurité	Très sécurisé (transactions ACID, chiffrement)	Sécurisé mais moins performant que PostgreSQL	Moins sécurisé pour les transactions complexes	Dépend des restrictions Google
Flexibilité	Très structuré (bon pour les relations élèves/tuteurs)	Structuré mais rigide	Très flexible mais pas optimal pour des relations complexes	Très flexible mais dépend du cloud
Requêtes avancées	SQL puissant (très adapté aux statistiques)	Bon mais plus limité	Peu performant pour des requêtes complexes	Très limité
Coût	Peut-être plus	Moins cher	Coût dépendant	Dépend du

	coûteux en hébergement		du volume	volume et de Google
--	------------------------	--	-----------	---------------------

HomeSkolar gère des données complexes et interconnectées, comme :

- Relations élèves/tuteurs : chaque élève a un ou plusieurs tuteurs.
- Calendrier et rendez-vous : requêtes qui peuvent être très évolutives.
- Messagerie : stockage de messages et gestion des notifications.
- Tâches : suivi des devoirs et alertes pour les rappels.

Pour cela nous utiliserons PostgreSQL au lieu de MySQL par exemple, cela permettra d'avoir une base de données plus sécurisée et plus évolutive. mais ce choix n'est pas non plus des plus important.

PostgreSQL est parfaitement adapté à ces besoins car :

- Stocke des relations complexes efficacement.
- Gère de grandes quantités de données sans perte de performance.
- Sécurise les transactions (idéal pour éviter la perte de données).
- Permet des requêtes avancées (statistiques sur les progrès des élèves, activité des tuteurs...).
- Facilement scalable en cas de forte croissance.

3) TECHNOLOGIES FRONT-END :

- Langages Indispensable : Html5, Css3, et JavaScript.
- Outils complémentaire possibles : SASS/SCSS, TypeScript, Webpack (bundlers)
- Framework possibles : React.js, Angular, Vue.js.
- Moteur de templates : Thymeleaf (car Java Spring en back end)

React.JS

Avantages	Inconvénient
<ul style="list-style-type: none"> • Javascript ou Typescript • Grande communauté et beaucoup de documentation • Architecture basée sur des composants réutilisable • Performance très efficace grâce au virtual dom • Complexité d'apprentissage modéré • Un écosystème très riche 	<ul style="list-style-type: none"> • Nécessite SSR avec Next.js pour être optimal avec le seo • Nécessite des bibliothèques tierces pour la gestion avancée de l'État.

<ul style="list-style-type: none"> • Facilité d'intégration d'autre technologies • très populaire 	
---	--

Angular

Avantages	Inconvénients
<ul style="list-style-type: none"> • Le plus vieux des frameworks proposé, donc très complet • Architecture MVC robuste et complète. • Sécurisé et très structuré 	<ul style="list-style-type: none"> • Framework le plus complexe • Performance bonne mais moins performante à cause du dom réel • Convient aux grandes entreprises sur des grandes infrastructures généralement • Très complet mais aussi très rigide, plus structuré mais moins flexible • TypeScript Obligatoire

Vue.js

Avantages	Inconvénients
<ul style="list-style-type: none"> • Basé sur des composant réactifs • Virtual Dom très rapide • Bonne documentation et communauté active (<React) • Complexité très peu élevée. 	<ul style="list-style-type: none"> • Communauté plus restreinte que React • Si le projet évolue beaucoup cela pourrait être un inconvénient. • Moins utilisé en entreprise.

Thymeleaf

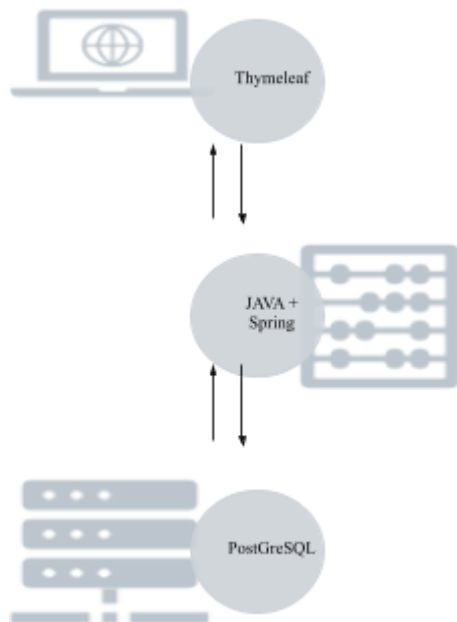
Avantages	Inconvénients
<ul style="list-style-type: none"> • Unification du langage : Utilisation exclusive du langage Java pour le back et le front • Réduction des dépendances : pas besoin de gérer un front en react/vue/angular • Facilité de déploiement : Tout est intégré dans un seul projet java, donc un seul serveur à déployer • Intégration natives à spring boot : fonctionne immédiatement avec spring facilite la configuration 	<ul style="list-style-type: none"> • Pas de rendu dynamique côté client : si il y a besoin d'y avoir un contenu dynamique (qui change en permanence par exemple), thymeleaf sera limité (chargement de la page obligatoire) • Interface plus statique : Les interactivités peuvent être limitées qu'avec un framework js, navigation fluide, animations...

Choix final du front end :

Nous avons opté pour une architecture full Java avec Spring Boot et Thymeleaf afin de garantir une cohérence technologique, une mise en œuvre rapide et un déploiement simplifié. Ce choix s'aligne parfaitement avec les besoins de notre projet, qui ne requiert pas de front-end ultra dynamique, mais une interface claire, fonctionnelle et facilement maintenable.

Nous utiliserons les outils HTML5 + CSS3 + thymeleaf avec Java qui offre un équilibre parfait entre performance requises, scalabilité et maintenabilité pour notre projet.

4) Résumé de notre pile technologique



En combinant ces technologies, HomeSkolar bénéficie d'un écosystème solide, prêt à répondre aux exigences d'un projet éducatif moderne. Ce choix nous permet de garantir une expérience utilisateur fluide, une gestion efficace des interactions, et surtout une capacité d'adaptation aux besoins futurs.

Diagramme de classes

Comment

Record video

