

Cahier des charges

Projet : Smart Student Networking Platform

Objectif : Créer une plateforme de mise en réseau intelligente pour connecter étudiants, professeurs, lauréat et parents.

1. Avant-projet résumé

La **Smart Student Networking Platform** vise à :

- Faciliter les échanges académiques et professionnels entre **étudiants, professeurs, lauréat** et **parents**.
- Proposer un espace sécurisé pour :
 - Partager des conseils,
 - Trouver des opportunités (stages, emplois, mentorat),
 - Créer des groupes d'étude,
 - Recevoir des informations sur la scolarité des étudiants (pour les parents).
- Utiliser des **algorithmes intelligents** pour :
 - Recommander des contacts pertinents,
 - Suggérer des événements ou des groupes d'intérêt,
 - Permettre une meilleure interaction personnalisée.

2. Contraintes techniques

- **Architecture :**
 - Application Web (backend) + Application Mobile (Flutter).
 - Backend développé avec **Spring Boot** pour garantir :
 - Cohérence technologique avec le projet intégré.
 - Développement rapide de **REST APIs** pour communiquer avec Flutter.
 - Possibilité d'évoluer vers une application web complète.
 - Base de données relationnelle gérée par **Spring Data JPA** (ex: PostgreSQL).
- **Sécurité :**
 - Authentification OAuth 2.0 (Google, Microsoft) avec Spring Security.

- Gestion fine des rôles : étudiant, professeur, alumni, parent.
- Chiffrement des données sensibles.
- **Scalabilité :**
 - Architecture modulaire basée sur Spring Boot.
 - Prête pour un hébergement cloud scalable.
- **Intégration :**
 - API de messagerie instantanée (par WebSocket ou par intégration externe).
 - Notifications Push via Firebase Cloud Messaging intégrées avec Flutter.
- **Intelligence :**

Recommandations basées sur l'activité des utilisateurs (algorithmes simples dans un premier temps).
- **Environnement :**

Déploiement sur AWS, Azure ou GCP.
- **Méthodologie :**

Projet en **microservices** possible à long terme.

3. Contraintes réglementaires

- **Conformité RGPD :**
 - Collecte minimale des données.
 - Consentement explicite des utilisateurs.
 - Droit d'accès, de rectification et de suppression.
- **Mentions légales, CGU, Politique de confidentialité** obligatoires.
- **Vérification d'âge** pour garantir un usage légal de la plateforme (accord parental pour mineurs).

4. Contraintes financières

- **Budget estimé :**
 - Développement initial : **20 000 € – 50 000 €**.
 - Hébergement cloud : **100 € – 300 €/mois** selon la charge.
 - Licences et outils tiers éventuels : ~10% du budget initial.

- **Optimisations :**
 - Usage d'outils open-source (Spring Boot, Flutter, PostgreSQL).
 - Déploiement progressif (MVP d'abord, puis version enrichie).

5. Contraintes de calendrier

- **Phase de conception** (2 semaines) :
Finaliser le cahier des charges + maquettes UI/UX.
- **Développement du MVP** (4 mois) :
 - Backend Spring Boot : 2 mois.
 - Application mobile Flutter : 2 mois.
- **Tests utilisateurs et ajustements** (1 mois).
- **Lancement officiel** :
Dans un délai de **6 mois maximum**.
- **Méthode de travail** :
 - **Methodologie Agile** : sprints de 2 semaines, démo à chaque fin de sprint.
 - Utilisation d'outils de gestion comme **Jira**.

6. La manière de répondre au cahier des charges

- **Équipe projet** :
 - Product Owner, Scrum Master, Développeurs Backend (Spring Boot), Développeurs Mobile (Flutter), UX/UI Designer, Testeurs.
- **Livrables attendus** :
 - Backend fonctionnel exposant des API REST sécurisées.
 - Application mobile Flutter connectée et fonctionnelle.
 - Documentation technique (API REST, architecture, modèle de données).
 - Politique RGPD et mentions légales validées.
- **Tests** :
 - Tests unitaires backend et mobile.
 - Tests d'intégration API.
 - Tests fonctionnels et tests de charge.
- **Déploiement** :

- Mise en production sur une infrastructure cloud sécurisée.
- Surveillance des erreurs et performances dès le lancement.
- **Suivi projet :**
 - Points quotidiens rapides ("daily stand-ups").
 - Démonstrations intermédiaires de l'avancement du produit.
 - Adaptation continue selon les retours utilisateurs.