**Premier plan de votre conception.**

**Etape 1 :**

Définir les acteurs principaux.

Etablir le diagramme de classe : relation entre les entités.

**Etape 2 :**

**Les fonctionnalités :**

**Gestion des utilisateurs**

Création et gestion des comptes étudiants, enseignants et administrateurs.

Attribution des rôles et permissions (ex: étudiant, enseignant, responsable des événements).

Authentification.

**Gestion des cours et des emplois du temps**

Consultation et gestion des emplois du temps.

Réservation des salles pour les cours et examens.

**Messagerie et notifications**

Système de messagerie interne entre étudiants et enseignants (email).

Envoi de notifications sur les modifications d’emplois du temps, nouveaux devoirs, notes, etc.

Alertes pour les événements et annonces importantes.

**Gestion des événements et clubs étudiants**

Création et gestion des événements académiques et culturels (conférences, séminaires, ateliers, etc.).

Gestion des inscriptions aux événements et suivi des participants.

Gestion des clubs étudiants (création, adhésion, activités).

**Etape 3 :**

Les services peuvent être 4 service s’appliquant directement à partir des fonctionnalités sinon, vous pouvez regrouper les deux premiers services et en laisser que 3.

**Etape 4**

Cette étape peut commencer maintenant en faisant un petit programme de teste pour commencer à se mettre dans le bain. Une fois c’est fait, vous devez être décisifs sur :

Choix des langages et framework :

Back-end : Spring Boot (Java), .NET Core, Node.js (NestJS), Django

Front-end : React, Angular, Vue.js

Communication entre services : REST, gRPC, Kafka

Base de données : PostgreSQL, MySQL, MongoDB

Stockage des médias : AWS S3, MinIO

Déploiement : Docker, Kubernetes, CI/CD (GitHub Actions, GitLab CI)

4. Conception de l’architecture

API Gateway : Kong, Nginx, Traefik

Service Discovery : Eureka, Consul

Circuit Breaker : Resilience4J pour gérer les pannes de services (on en reparlera sur ça).