Jeudi 19  Octobre 2023

# Projet : Système de Gestion Universitaire basé sur Microservices (SGUM)

**Objectif général :**

Créer un système universitaire flexible, évolutif et résilient pour la gestion intégrée des étudiants, des cours, du personnel et des ressources, en utilisant une architecture de microservices.

**Fonctionnalités principales :**

1. **Module d'Admission** :
    - Inscription des étudiants
    - Traitement des candidatures
    - Enregistrement des pièces justificatives
2. **Module de Gestion des Cours** :
    - Création et mise à jour des cours
    - Planification des sessions de cours
    - Attribution des salles et ressources
3. **Module de Gestion des Étudiants** :
    - Profils étudiants avec historique académique
    - Suivi des performances académiques
    - Gestion des absences
4. **Module de Gestion du Personnel** :
    - Profils des enseignants et du personnel administratif
    - Attribution des cours aux enseignants
    - Gestion des contrats et de la rémunération
5. **Module d'Évaluation** :
    - Création de tests et d'examens
    - Soumission des notes par les enseignants
    - Calcul des moyennes et des résultats finaux
6. **Module de Ressources** :
    - Gestion des salles de cours, laboratoires et équipements
    - Planification et réservation des ressources
7. **Module de Communication** :
    - Portail d'annonces pour les étudiants et le personnel

- Système de messagerie intégré

**Technologie & Architecture :**

- **Microservices** : Chaque module sera développé comme un microservice indépendant avec sa propre base de données.
- **API RESTful** : Pour l'interaction entre les microservices et les interfaces utilisateur.
- **Containerisation** : Utilisation de Docker pour empaqueter chaque microservice.
- **Orchestration** : Kubernetes pour déployer et gérer les conteneurs.
- **Message Broker** (comme Kafka ou RabbitMQ) pour la communication asynchrone entre les microservices.

**Stratégies de sécurité et de résilience :**

- **Authentification & Autorisation** : Utilisation d'un gateway API pour gérer l'accès aux microservices.
- **Circuit Breakers** : Pour éviter que les défaillances d'un microservice ne se propagent à d'autres.

**Déploiement & Monitoring :**

- **CI/CD** : Mise en place d'une chaîne d'intégration et de déploiement continue pour faciliter les mises à jour.
- **Monitoring & Logging** : Intégration de solutions comme Prometheus et Grafana pour la surveillance, et ELK Stack (Elasticsearch, Logstash, Kibana) pour la centralisation des logs.

**Équipe projet :**

- **Architectes** : Conception de l'architecture globale.
- **Développeurs** : Création des microservices et des interfaces.
- **DevOps** : Gestion de l'infrastructure, du déploiement et de la surveillance.
- **QA/Testeurs** : Assurance qualité et tests.

**Résultat attendu :**

Un système universitaire moderne, flexible et évolutif, capable de répondre rapidement aux besoins changeants d'une institution éducative tout en garantissant une haute disponibilité et une performance optimale.

Notez que cette description est basée sur une vision générale. Chaque université ou institution pourrait avoir des besoins spécifiques qui nécessitent des ajustements ou des ajouts à ce plan.