



Université Mohammed V - Rabat
École Nationale d'Informatique et
d'Analyse des Systèmes



PREMIER LIVRABLE DU PROJET FÉDÉRATEUR

FILIÈRE

GÉNIE LOGICIEL

SUJET:

Conception et mise en œuvre d'un moteur de
Recommandation adaptatif pour l'orientation académique
Basé sur une architecture micro services

Réalisé par :

M.Ayman ANNA

M.Ismail AHAKAY

Encadré par :

Prof Mme. Bouchra EL ASRI

Mme Zineb

ELKAIMBILLAH

Année Universitaire 2024-2025

Table des matières

| | |
|---|----------|
| Introduction | 2 |
| 0.1 Système de recommandation pour l'orientation académique | 2 |
| 0.2 Adaptativité : | 2 |
| 0.2.1 Cas d'utilisation : | 2 |
| 0.3 Architecture microservices | 3 |
| 0.3.1 Définition | 3 |
| 0.3.2 Avantages détaillés | 3 |
| Conception | 4 |
| 0.3.3 Représentation Graphique | 4 |
| Technologies Utilisés | 7 |

Introduction

L'objectif de ce projet est de concevoir et mettre en œuvre un moteur de recommandation adaptatif pour guider les étudiants dans leurs choix académiques, en s'appuyant sur une architecture microservices moderne. Ce système vise à répondre à des enjeux clés de l'orientation académique tout en offrant une solution flexible et évolutive.

0.1 Système de recommandation pour l'orientation académique

- **Problématique adressée** : Les étudiants, notamment les élèves de baccalauréat (1ère et 2ème années au Maroc), font face à une multitude d'options (filières, cours, certifications) souvent difficiles à appréhender sans accompagnement personnalisé. Le système propose des recommandations pertinentes basées sur leurs profils (intérêts, compétences, performances) et les exigences académiques, afin de les guider efficacement dans leurs choix post-bac.
- **Fonctionnalités principales** : Suggestion de cours adaptés, identification des prérequis manquants, proposition de parcours pour atteindre des objectifs professionnels ou académiques spécifiques (ex. : préparation à une licence ou un master), avec un focus particulier sur l'accompagnement des élèves de 1ère et 2ème années de baccalauréat au Maroc dans leur transition vers l'enseignement supérieur.

0.2 Adaptativité :

Capacité du système à ajuster ses recommandations en fonction de changements dans les données d'entrée ou le contexte utilisateur.

0.2.1 Cas d'utilisation :

- **Évolution des performances** : Si un étudiant améliore ses notes dans une matière, le système peut suggérer des cours plus avancés.
- **Gestion des candidatures** : Faciliter la soumission et le suivi des candidatures par les étudiants et les entreprises.
- **Changement d'intérêts** : Si un étudiant passe d'un intérêt pour les sciences aux arts, les recommandations s'ajustent en conséquence.
- **Contraintes externes** : Le système adapte les recommandations en intégrant de nouvelles données entrantes, telles que l'apparition de nouveaux diplômes ou de nouveaux débouchés professionnels, pour refléter les opportunités émergentes.

0.3 Architecture microservices

L'architecture microservices a été choisie pour structurer le système en unités indépendantes, chacune dédiée à une fonctionnalité spécifique. Cette approche contraste avec les architectures monolithiques traditionnelles, où toutes les composantes sont étroitement couplées.

0.3.1 Définition

Chaque microservice est une application autonome qui exécute une tâche précise (ex. : gérer les profils, calculer les recommandations) et communique avec les autres via des interfaces standardisées (API REST ou messages asynchrones).

0.3.2 Avantages détaillés

- **Scalabilité** : Si le nombre d'utilisateurs augmente, le Service Recommandation peut être déployé sur plus de serveurs sans toucher aux autres services.
- **Résilience** : Une défaillance dans le Service Données (ex. : panne de la base) n'empêche pas le Service Utilisateur de fonctionner pour afficher les profils existants.
- **Flexibilité** : Ajouter une nouvelle fonctionnalité (ex. : un module de prédiction des taux de réussite) se fait en développant un nouveau service, sans modifier l'existant.

Conception

Diagramme de Cas d'Utilisation

Acteurs

- **Étudiant** : Utilisateur principal qui s'inscrit, met à jour son profil et consulte les recommandations.
- **Système** : Backend qui traite les données et génère les recommandations.

Cas d'Utilisation

- **S'inscrire** : L'étudiant entre ses données initiales (nom, année, intérêts, matières) et reçoit des recommandations initiales en retour.
- **Mettre à jour le profil** : L'étudiant modifie ses informations (par exemple, ajoute une matière), ce qui impacte les recommandations.
- **Consulter les recommandations** : L'étudiant visualise ses programmes d'enseignement supérieur et ses parcours professionnels.

0.3.3 Représentation Graphique

Le diagramme de cas d'utilisation peut être représenté comme suit :

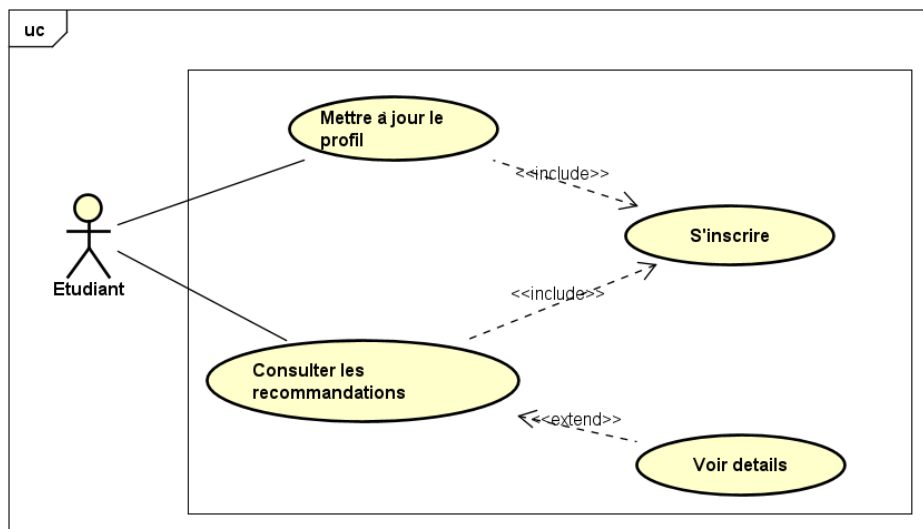


FIGURE 1 – Diagramme de Cas d'Utilisation

Note : Le diagramme inclut :

- Un acteur "Étudiant" (stick figure) relié aux cas d'utilisation par des flèches.
- Un acteur "Système" (rectangle) interagissant avec les cas backend.
- Des ovales pour chaque cas d'utilisation : "S'inscrire", "Mettre à jour le profil", "Consulter les recommandations".

Diagramme entite-relation

La conception de notre système de recommandation académique pour les étudiants marocains en première et deuxième années du baccalauréat, illustrée dans la figure 5, repose sur une architecture basée sur un graphe de connaissances modélisé dans Neo4j. Ce diagramme représente les entités clés et leurs relations, telles que le profil de l'étudiant connecté à son nom, son email, son baccalauréat, ainsi qu'à ses sujets d'intérêt et aspirations de carrière.

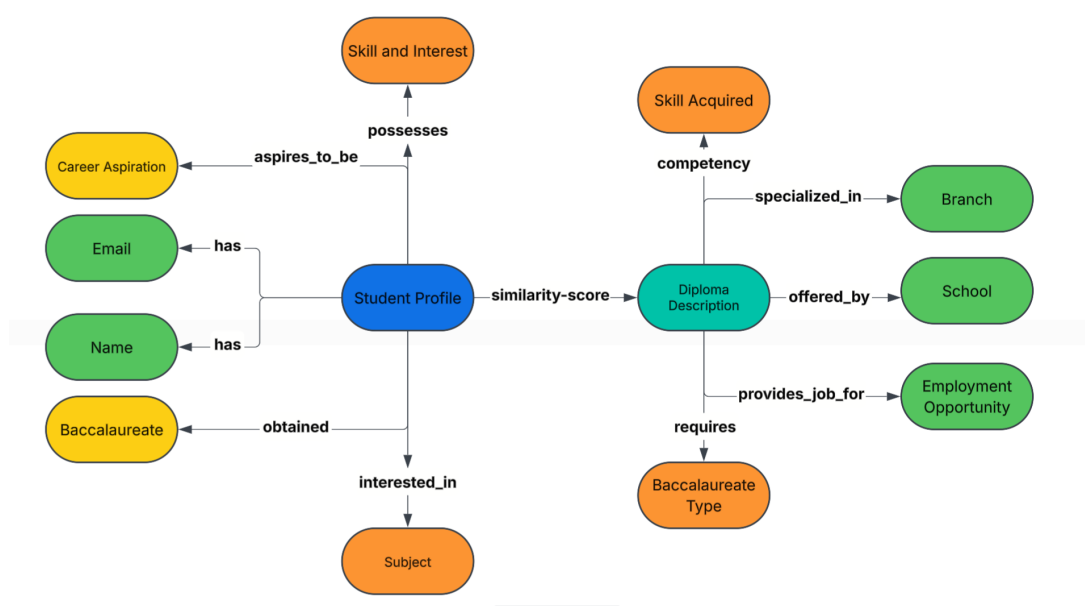


FIGURE 2 – Conception du graphe de connaissances pour le système de recommandation académique.

Cette structure permet de capturer les relations complexes entre les étudiants, leurs compétences, leurs parcours académiques et leurs opportunités professionnelles, facilitant ainsi des recommandations adaptatives à court terme (programmes d'études supérieures) et à long terme (chemins de carrière) via un modèle de réseau neuronal de graphe (GNN) intégré.

Architecture Design

La conception de l'architecture de notre système de recommandation académique pour les étudiants marocains en première et deuxième années du baccalauréat, illustrée dans la figure 3, repose sur une architecture microservices modulaire et scalable.

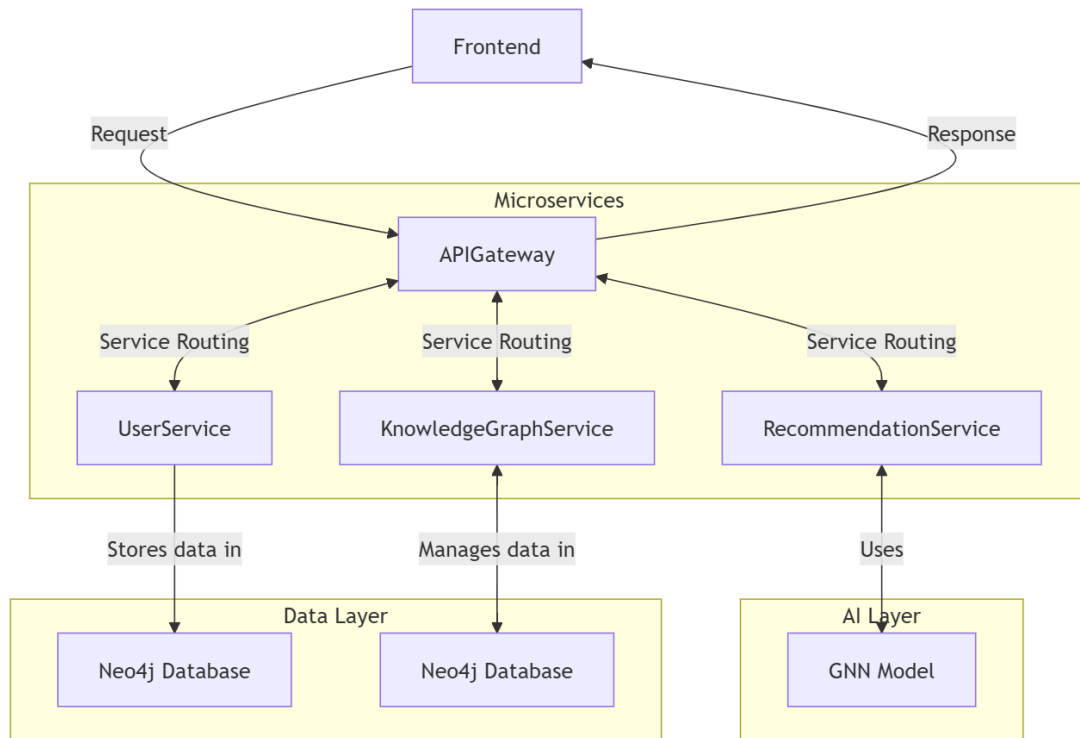


FIGURE 3 – Conception de l'architecture microservices pour le système de recommandation académique.

Comme montré dans le diagramme, le frontend interagit avec une couche de microservices à travers une API Gateway, qui route les requêtes vers trois services principaux : le *UserService*, le *KnowledgeGraphService*, et le *RecommendationService*. Le *UserService* gère les profils des étudiants et stocke les données dans la base de données Neo4j, tandis que le *KnowledgeGraphService* administre les relations graphe dans Neo4j, représentant les étudiants, leurs sujets, programmes et aspirations professionnelles. Le *RecommendationService* exploite un modèle de réseau neuronal de graphe (GNN) pour générer des recommandations adaptatives, en s'appuyant sur les données extraites de Neo4j. Cette architecture permet une gestion efficace des utilisateurs, une persistance des données structurées, et des prédictions intelligentes en temps réel.

Technologies Utilisés

Neo4j

Rôle : Base de données graphe pour représenter les relations complexes entre étudiants et parcours académiques post-bac.

Avantages : Permet des requêtes rapides sur des structures relationnelles complexes, essentielles pour modéliser les différentes possibilités de parcours académiques.

Pourquoi Neo4j a été adopté : Neo4j excelle dans la gestion des relations entre différents parcours et profils d'étudiants, facilitant l'analyse et la visualisation des parcours possibles basés sur des préférences et des compétences individuelles.

Exemple d'utilisation : Utilisation de requêtes Cypher pour identifier les parcours académiques suivis par les étudiants ou pour recommander des parcours adaptés à leurs aspirations et compétences.



FIGURE 4 – Neo4j

GNN (Graph Neural Networks)

Rôle : Analyse des graphes pour générer des recommandations de parcours académiques en exploitant les relations complexes entre les profils d'étudiants.

Avantages : Utilise la topologie du graphe pour des recommandations contextuelles et précises sur les parcours académiques.

Pourquoi les GNN ont été adoptés : Les GNN permettent de modéliser avec précision les interdépendances entre différents parcours académiques et les profils étudiants, offrant des recommandations personnalisées basées sur des analyses de réseau.

Exemple : Prédiction des parcours académiques appropriés pour un étudiant en fonction des parcours réussis par des profils similaires.

Spring Boot + Outils Microservices

Rôle : Développement de microservices robustes pour gérer différentes fonctionnalités du système de recommandation de parcours.

Avantages : Facilite le développement rapide d'API REST et l'intégration avec d'autres services et technologies.

Pourquoi Spring Boot et les outils microservices ont été adoptés : Le framework permet une intégration et une évolutivité facile avec les composants existants, crucial pour une architecture basée sur les microservices où chaque service peut être développé, déployé et mis à jour indépendamment.

Exemple : Un microservice de recommandation utilisant un endpoint `/recommend` qui intègre des modèles de GNN pour fournir des recommandations dynamiques aux étudiants concernant les parcours académiques post-bac adaptés à leurs profils.



FIGURE 5 – Spring boot