

Suivi des étudiants dans une formation à la carte.

L3 Miage, projet de programmation Java

1 Présentation générale

1.1 Contexte

L'application à réaliser se place dans le contexte d'une **Université** qui proposerait des formations à la carte dans lesquels les **étudiants** choisissent les enseignement qu'ils veulent suivre. Les enseignements sont régis par des notions de prérequis¹.

Un étudiant peut décrocher un **diplôme** lorsqu'il a validé tous les **enseignements** nécessaires pour faire un **parcours complet**.

1.2 Objectif

L'objet de l'application est de suivre les étudiants : les **inscrire aux enseignements**, **voir où ils en sont** de leur parcours et **saisir les enseignement validés**.

2 Le Logiciel

Cette section décrit l'application demandée. Cela constitue la base à partir de laquelle des évolutions seront demandées ultérieurement.

Dans un premier temps, nous définissons les concepts qui seront manipulés par l'application.

Une **année universitaire** est à cheval sur deux années civiles (par ex. 2021-2022). Elle est divisée en deux **semestre** : le semestre impair (début de l'année universitaire) et le semestre pair (fin de l'année universitaire)

2.1 Les formations et les UE

Les formations sont organisées par **mention** puis sous-divisées en **parcours** (par exemple mention MIASHS parcours Miage et parcours IO).

Les **Unités d'Enseignement (UE)** sont les cours proposés par les différentes mentions. Chaque UE est associée à une mention, à l'exception de certaines UE qui sont des *UE d'ouverture*. Une UE possède un code d'identification et rapporte des crédits ECT (3 ou 6).

Dans une formation à la carte, toutes les UE ont potentiellement un ou des prérequis qui sont d'autres UE et qui doivent être validées antérieurement (par exemple on peut imaginer que pour suivre "Prog Objet 2" il faut avoir validé "algorithmique" et "Prog Objet 1"). Les UE qui n'ont pas de prérequis peuvent être choisies dès l'arrivée en première année.

2.2 Validation du parcours

Un étudiant suit en principe des UE pour un total maximum de 30 ECTS chaque semestre. Il est possible d'avoir des dérogations pour suivre jusqu'à 39 ECTS.

1. Similaire à la LFlex mais avec plus de libertés de choix

Chaque parcours est défini par un certain nombre d'UE obligatoires pour sa validation. Un étudiant qui a validé ces UE obligatoires ainsi que 180 ECTS obtient alors son diplôme.

2.3 Les étudiants

Un étudiant est défini par son numéro d'étudiant. Il possède un prénom et un nom et il est inscrit à un parcours d'une mention.

Il faut effectuer le suivi pour chaque étudiants des UE qu'il a suivies, en quelle année universitaire et quel semestre (pair ou impair) il a suivi cette UE. Il faut de plus mémoriser si l'étudiant a validé ou non l'UE à l'issue de ce semestre. Par exemple un étudiant peut suivre une UE et échouer sur le semestre pair de 2021-2022 puis suivre et valider l'UE le semestre impair de 2021-2022.

Il faut également suivre les UE auxquelles l'étudiant est inscrit pour le semestre en cours.

2.4 Objectifs

À l'ouverture, l'application doit charger les données concernant les mentions, parcours, UE et la liste des étudiants à partir de fichiers .csv. Ces données doivent par défaut être chargées à partir d'une base de données.

L'application doit permettre de réaliser les opérations suivantes :

- Ajouter ou modifier un étudiant.
- Visualiser pour un étudiant, quelles sont les UE qu'il a validées, les UE qu'il suit actuellement ainsi que les UE pour lesquelles il a les prérequis.
- Inscrire un étudiants à des UE pour lesquelles il a les prérequis.
- Marquer une UE comme validée ou échouée pour le semestre en cours
- Passer au semestre suivant
- Enregistrer les changements effectués dans la base de données.

Il vous est également demandé de proposer, dès que c'est pertinent, de permettre la filtration des listes par mention et parcours.

Il n'est pas demandé de pouvoir modifier la structure des formations depuis le logiciel.

3 Consignes

L'application devra être réalisée en Java. Les données seront chargées à partir d'une base de données.

3.1 Interface et ergonomie

Il est important que votre logiciel propose une interface lisible et efficace. L'ergonomie sera particulièrement importante pour les traitements les plus fréquents et répétitifs. Le nombre d'actions nécessaire pour ces traitements devra être le plus faible possible.

3.2 Conception, Code et documentation

Il est recommandé de prévoir une conception modulaire de votre projet. Cela sera un atout lors de l'implémentation des évolutions du logiciel. Utilisez les interfaces et classe abstraites proposées par le langage.

L'utilisation de bibliothèques externes est autorisée. Il n'est pas judicieux de redévelopper des fonctionnalités qui existent déjà (par exemple la manipulation de fichiers .csv).

Toute utilisation de l'IA devra être documentée dans le détail.

Il vous est également demandé de produire une documentation sous forme de **javadoc**. Écrivez les commentaires javadoc à chaque fois que vous ajoutez une classe ou une fonctionnalité.

3.3 Rendez-vous intermédiaire

Un rendez-vous intermédiaire est prévu pour ce projet. Vous devrez y faire une démonstration de la première version du logiciel en 5 minutes. Suite à cette démonstration, il sera demandé des évolutions qui devront intégrer la version finale du logiciel.

Vous trouverez les heures de passage de chaque groupe sur le moodle avant la séance.

Préparez votre démonstration et la scriptant, avec un jeu de donné pertinent et conséquent. Vous devez présenter toutes les fonctionnalités de manière convaincante et montrer les limites de votre logiciel.

À l'issue de ce rendez-vous, de nouvelles fonctionnalités vous seront demandées.

3.4 Validation du projet

Pour la validation du projet, vous devrez faire une démonstration complète du logiciel comme pour le rendez-vous, avec un temps de 10 minutes maximum.

La présentation sera suivie d'une discussion sur les différents aspects du logiciel, l'organisation du travail et les difficultés rencontrées.

3.5 Rendus

Le projet fera l'objet de rendus qui seront réalisés sur moodle (un avant le premier rendez-vous et un avant la validation du projet). Les deadlines seront visibles sous moodle. La liste détaillée des éléments à déposer vous sera donnée.