

Module : **ARCHITECTURE DES SI II (SPRING)**

Enseignants : ghassen
hammouda

Classe : **4SE4**

Nombre de pages : 2

Documents autorisés: **OUI**

Calculatrice autorisée: **NON**

Internet autorisée: **NON**

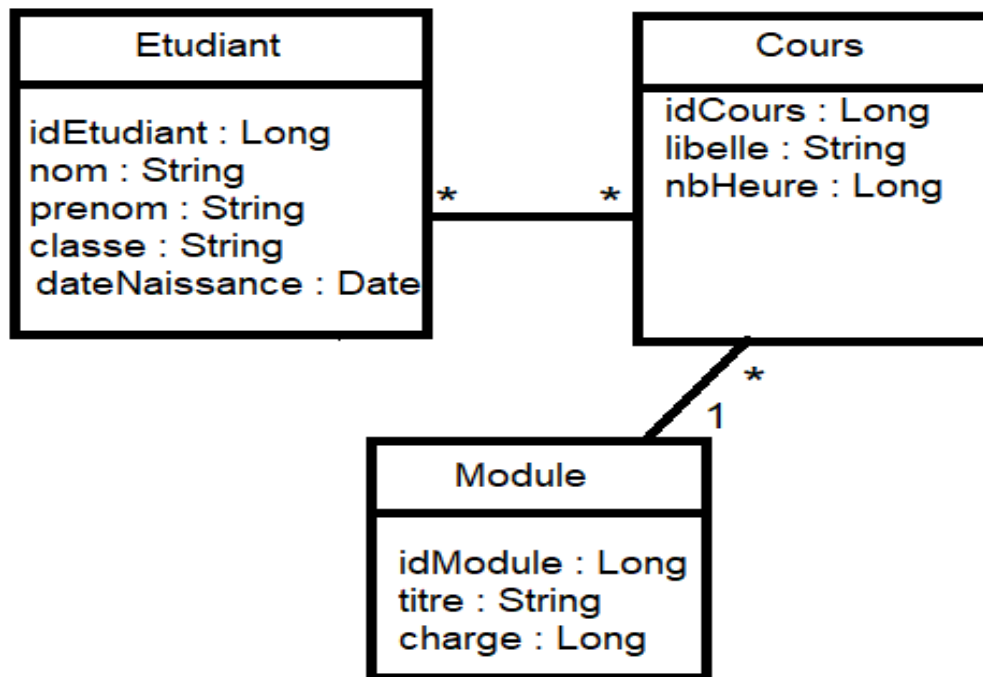
Date : **01/12/2021**

Heure : **10H30**

Durée : **00H40Min**

La validation de l'épreuve est appliquée sur la base d'un code source exécutable. Aucun code source non fonctionnel n'est comptabilisé lors de la validation.

On vous propose donc d'implémenter une application simplifiée de gestion de modules pour le compte d'un établissement universitaire. **Ci-dessous le diagramme de classes :**



I.1. (6points) :

Implémenter les entités qui permettent de générer le schéma de la base de données comme illustré dans le diagramme de classes sachant que :

Les identifiants sont auto-générés avec la stratégie «IDENTITY».

La relation bidirectionnelle **Cours - Etudiant** modélise le fait qu'un étudiant peut accéder à plusieurs cours et qu'un cours peut être suivi par plusieurs étudiants (Etudiant étant le child).

La relation bidirectionnelle **Cours - Module** : Un cours fait partie d'un seul module or que le module peut contenir plusieurs cours.

I.2. (14 points) :

Développer le code nécessaire dans une classe annotée par **@RestController** qui fait appel aux différents services. (Exposition des services avec Spring REST MVC, et Tests avec **Postman** ou **Swagger**). Voici les Services demandés :

- a) En respectant la signature de la méthode suivante, ajouter l'étudiant ci-dessous (4pts) :

void ajouterEtudiant(Etudiant etudiant);

Etudiant	nom	prenom	classe	dateNaissance
	Salhi	Ahmed	4SE4	1996-10-03

- b) Ajouter manuellement un cours et un module dans la base de données

Cours	Libelle	nbHeure
	Spring Batch	2
Module	Titre	charge
	ASI II Spring	42

- c) Affecter l'étudiant Ahmed Salhi au cours Spring Batch en respectant la signature suivante (5pts) :

void affecterEtudiantAuCours (Long idEtudiant, Long idCours)

- d) En respectant la signature suivante, afficher la liste des cours d'un module donné (5pts) :

List<Cours> listeCoursParModule(Long idModule) ;