

Module : **ASI II (Spring)**

Enseignant(s) : Spring Team

Nombre de pages : **03**

Documents autorisés: **OUI**

Calculatrice autorisée: **NON**

Internet autorisée: **NON**

Date : 14/12/2022

Heure : 09H00

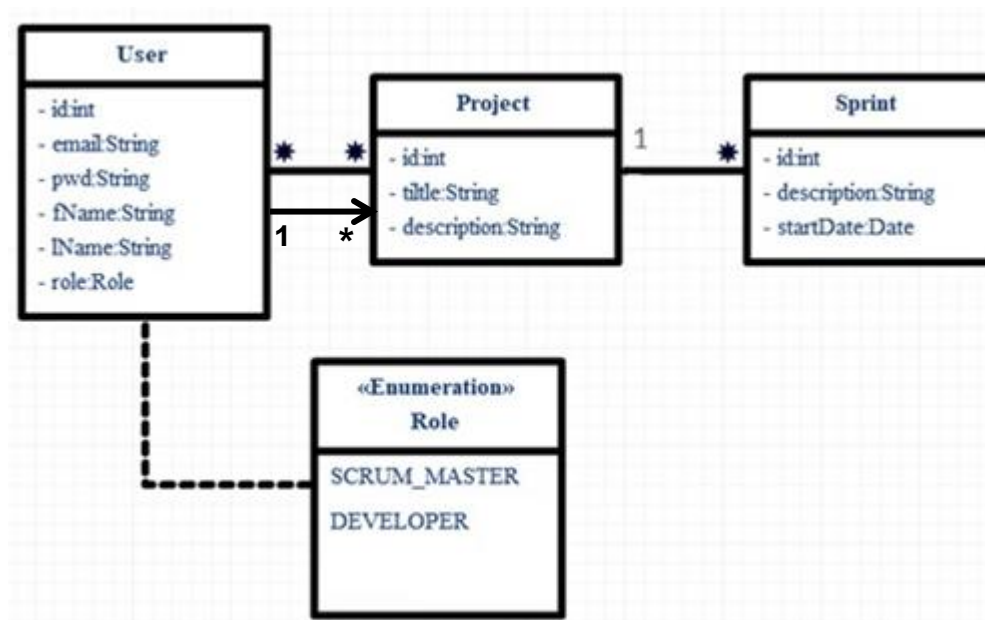
Durée : 01H30

**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 d'implémenter une application simplifiée de gestion des équipes qui utilisent la méthode Agile Scrum à Esprit.

- Un Utilisateur (**User**) de notre application peut avoir l'un des rôles (**Role**) suivants au niveau des projets (**Project**) sur lesquels il intervient : **Scrum Master ou Developer**.
- Un utilisateur (**Developer**) peut travailler sur plusieurs projets. Un projet peut avoir plusieurs intervenants.
- Un projet est composé de plusieurs Itérations (**Sprint**). Un Sprint est lié à un unique projet.
- Un **Scrum Master** peut être affecté à plusieurs projets, et un **Project** peut être piloter par un seul **Scrum Master**.
- Les autres intervenants (Product Owner et Client) et les User Stories ne seront pas traitées dans notre étude de cas.

Ci-dessous le diagramme de classes :



## Partie I (5points) :

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 des trois entités sont auto-générés avec la stratégie «IDENTITY».
- L'association User-Project représente une relation bidirectionnelle. Le Project est le Child (Slave).
- Les énumérations doivent être stockées en tant que chaînes de caractères dans la base de données

## Partie II (15 points) :

Développer le code nécessaire dans une classe annotée par **@RestController** qui fait appel aux différents services. (Exposition des services avec Spring REST MVC).

- a) Ajouter les deux utilisateurs ci-dessous, en respectant la signature suivante **(1pt)** :
- public User addUser (User user)**

role	fName	IName	email	pwd
DEVELOPER	Mohamed	BOUHLEL	ab@esprit.tn	dev1
DEVELOPER	Kais	ALLANI	ka@neoxam.tn	dev2
SCRUM_MASTER	Ahmed	Farah	fa@esprit.tn	SM

- b) Ajouter les deux projets ci-dessous, en respectant la signature suivante **(2pts)** :
- public Project addProject (Project project)**

title	description
MAP	Gestion de Mandats, Assignations et Projets
OTDAV	Gestion des Œuvres

NB : Pour l'ajout d'un projet, il faut ajouter en même temps au moins un sprint.

Sprint (description)	Sprint (startDate)	Projet
Sprint Projets	2023-05-27	OTDAV
Sprint Mandat	la date d'aujourd'hui	MAP

- c) Affecter le projet MAP aux deux utilisateurs Mohamed BOUHLEL et Kais ALLANI, en respectant la signature suivante **(1pt)** :

**public void assignProjectToDeveloper (int projectId, int devId)**

- d) Affecter le projet MAP au Scrum\_master Ahmed Farah, en respectant la signature suivante **(2pts)** :

**public void assignProjectToScrumMaster(int projectId, String fName, String IName)**

- e) Afficher les projets pilotés par le scrum master Ahmed Farah, comme indiqué ci-dessous, et en respectant la signature suivante (2pt) :

**public List<Project> getProjectsByScrumMaster(String fName, String lName)**

- f) Ajouter un Sprint et l'affecter au projet MAP, comme indiqué ci-dessous, et en respectant la signature suivante (2pt) :

**public void addSprintAndAssignToProject(Sprint sprint, int idProject)**

Sprint (description)	Sprint (startDate)
Sprint Mandat 2	2023-12-27

- g) Créer un service planifié automatiquement qui permet d'afficher au niveau de console en utilisant des logs, chaque 30 secondes, les projets qui ont des sprints en cours (start date < = date d'aujourd'hui), en respectant la signature suivante : (2pts)

**public void getProjectsCurrentSprints()**

- h) Créer un **Aspect** qui permet d'afficher, **après le bon déroulement**, le nom de chaque méthode appelée dont le type de retour est **void** (2pt)

**Bon Travail !**