

Module : **Architecture des SI II (Framework Spring)**

Enseignante : **Sirine Naifar**

Classe : **4DS2**

Nombre de pages : **03**

Documents autorisés: **OUI**

Calculatrice autorisée: **NON**

Internet autorisée : **NON**

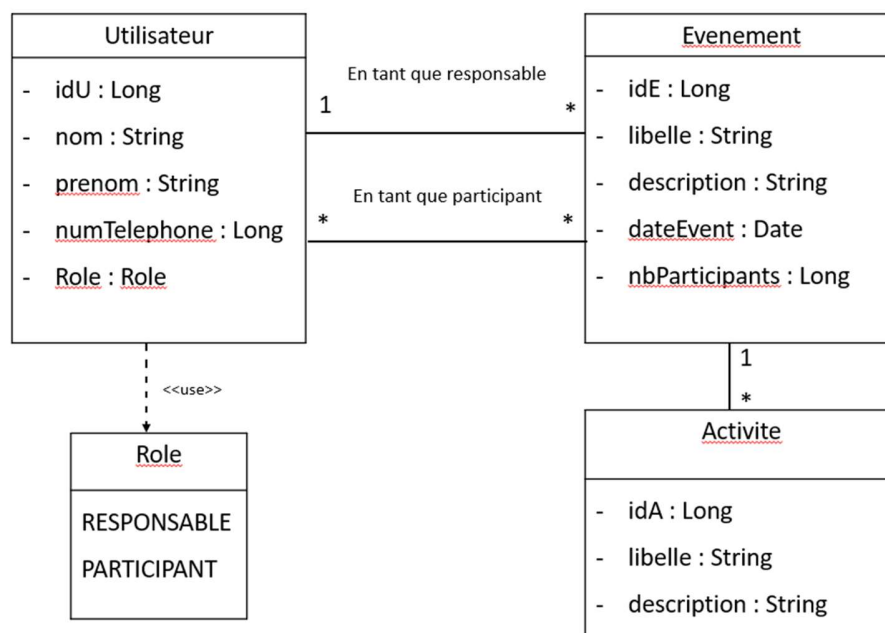
Date : **09/05/2022**

Heure : **13h30**

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 donc d'implémenter une application simplifiée de gestion des événements pour le compte d'une agence événementielle. Ci-dessous le diagramme de classes :



I.1. (05 points)

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 **Evènement - Activité** modélise le fait qu'un **évènement** peut avoir plusieurs **activités** et qu'une **activité** peut être affectée qu'à un seul **évènement**.
- La première relation bidirectionnelle **Utilisateur - Evènement** : Un **responsable** peut gérer plusieurs **évènements** or que l'**évènement** peut être géré qu'avec un seul **responsable**.
- La deuxième relation bidirectionnelle **Utilisateur - Evènement** : Un **participant** peut participer à plusieurs **évènements** et un **évènement** peut contenir plusieurs participants (**Evènement est le child**).

I.2. (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, et Tests avec Postman ou Swagger**). Voici les Services demandés :

- a) En respectant la signature de la méthode suivante, ajouter les deux événements ci-dessous (2pts) :

Evenement addEvenement(Evenement e);

	Libelle	Description	DateEvent	NbParticipants
Evènement	Halloween	Une fête folklorique	2022-05-09	2
	La journée de l'habit traditionnel tunisien	Une exposition d'art et un défilé de mode	2022-03-16	3

- b) En respectant la signature de la méthode suivante, ajouter les deux activités ci-dessous (2pts) :

Activite addActivite(Activite a);

	Libelle	Description
Activité	Activité 1	Dance
	Activité 2	Dessin

- c) Affecter les deux activités à l'évènement « Halloween » en respectant la signature suivante (2pts) :

void affectActiviteToEvent(Long activiteId, Long eventId);

- d) Ajouter le responsable et les participants ci-dessous à l'évènement « La journée de l'habit traditionnel tunisien » en respectant la signature suivante (5pts) :

String addUser(User u, Long idEvent);

	Nom	Prénom	NumTéléphone	Rôle
Utilisateur	Kadri	Hanin	66212121	RESPONSABLE
	Sallem	Salem	66989898	PARTICIPANT
	Mrabet	Mohsen	66121212	
	Khdiji	Firas	66585858	

NB : La méthode vérifie les conditions suivantes :

- 1- Si l'évènement n'a pas un responsable, l'affectation sera effectuée et la méthode retourne « **L'affectation du responsable est effectuée avec succès** »
 - 2- Si l'évènement a déjà un responsable, la méthode retourne « **Il y a déjà un responsable !** »
 - 3- S'il y a encore des places disponibles dans l'évènement, l'affectation sera effectuée et la méthode retourne « **L'affectation du participant est effectuée avec succès** »
 - 4- Si le nombre des participants est atteint, la méthode retourne « **l'évènement est complet !** »
- e) En respectant la signature suivante, afficher la liste des évènements d'un responsable donné (2pts) :

```
List<Evenement> getEventsByResponsable(Long responsableId);
```

- f) Nous souhaitons créer un service programmé automatiquement (scheduled) permettant d'afficher tous les événements d'aujourd'hui comme le montre la figure 1.

Créer le service qui permet d'afficher les événements concernés toutes les 30 secondes en respectant la signature suivante (2pts) :

```
List<Evenement> getEventsOfToday();
```

```
***** Evenements d'aujourd'hui *****
Libelle => Halloween, Description => une fête folklorique
```

Figure 1: Message à afficher

Bonne chance

