

Se former autrement

EXAMEN

Semestre : 1 ☒ 2 ☐

Examen Blanc ☒

Module : **ASI II (Spring)**

Enseignant(s) : **Thouraya LOUATI**

Classe(s) : **4TWIN4**

Documents autorisés : OUI ☒ NON ☐ Nombre de pages : 3

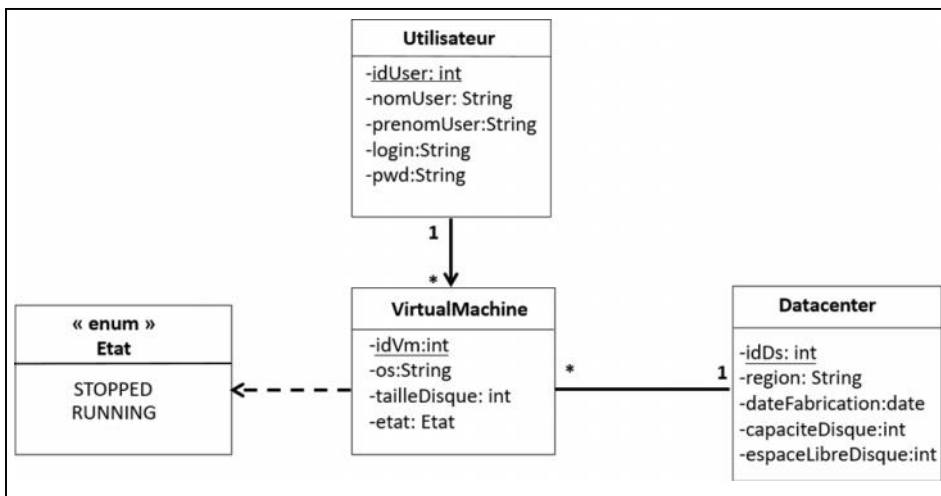
Calculatrice autorisée : OUI ☐ NON ☒ Internet autorisée : OUI ☐ NON ☒

Date : ...**Décembre 2021**..... Heure**09h00**..... Durée :**1h30**....

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 se propose de mettre en place une application de gestion des machines virtuelles sur un Datacenter.

Ci-dessous, le diagramme de classes.



Spring Boot - Spring Core – Spring Data JPA – Spring REST MVC – Log4J - Lombok

I.1 (5 points)

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

- Les identifiants sont auto-générés avec la stratégie «**IDENTITY**».
- L'association **unidirectionnelle** Utilisateur-VirtualMachine indique qu'un utilisateur peut avoir plusieurs machines virtuelles et une machine virtuelle appartient à un seul utilisateur.
- L'association **bidirectionnelle** VirtualMachine-Datacenter indique qu'un datacenter peut contenir plusieurs machines virtuelles et une machine virtuelle peut appartenir à un seul datacenter.
- L'**énumération** doit être stockée en tant que chaîne de caractères dans la base.

I.2 (10 points)

Développer le code nécessaire dans une classe annotée par @RestController qui fait appel

aux différents services. (Exposition des services avec Spring REST MVC, vous pouvez utiliser Postman ou Swagger).

a) Ajouter un utilisateur ayant les détails ci-dessous en respectant la signature suivante (I1.5) :

```
public void ajouterUser(Utilisateur u);
```

login	nomUser	prenomUser	pwd
allogin	Ben Salah	Ali	alipwd

b) Ajouter 2 datacenters ayant les détails ci-dessous en respectant la signature suivante (I1.5) :

```
public void ajouterDc(Datacenter datacenter);
```

capaciteDisque	dateFabrication	espaceLibreDisque	region
500	30-09-2019	100	USA Est
100000	11-11-2019	8000	USA West

c) Ajouter 3 machines virtuelles ayant les détails ci-dessous en respectant la signature suivante (I1.5) :

```
public int ajouterVm(VirtualMachine vm);
```

etat	os	tailleDisque
STOPPED	Windows	200
STOPPED	Linux	50
STOPPED	Mac	80

d) Affecter les 3 machines virtuelles que vous avez ajoutées à l'utilisateur Ali Ben Salah en respectant la signature suivante (I2) :

```
public void affecterVmuser(int idvm, int iduser);
```

e) Affecter chacune des machines virtuelles au premier Datacenter qui un **espace libre supérieur à la taille du disque de la machine virtuelle**. Après affectation, mettre à jour la taille de l'espace disque libre sur le datacenter (i.e. **Nouveau espaceLibreDisque = Ancien espaceLibreDisque – tailleDisque de la machine virtuelle**). Utiliser la signature suivante (I2).

```
public void affecterVm(int idvm);
```

- f) Démarrer la première machine virtuelle de Ali, i.e, changer l'état de la machine virtuelle de STOPPED à RUNNING en respectant la signature suivante **(/0.5)**.

```
public void demarrerInstanceUser(int idvm);
```

- g) Arrêter la première machine virtuelle de Ali, i.e, changer l'état de la machine virtuelle de RUNNING à STOPPED en respectant la signature suivante **(/0.5)**.

```
public void arreterInstanceUser(int idvm);
```

- h) Lister le ou les **datacenters (n'afficher que leur région)** qui ont une date de création ultérieure à 01-11-2019 ainsi que toutes les **machines virtuelles (n'afficher que leur os)** appartenant au datacenter en question tout en respectant la signature suivante **(/2)**.

```
public List<Datacenter> listerDatacenter(); // Avec Scheduler (/1)
```

- i) Créer un aspect qui permet de calculer et afficher dans les logs la durée d'exécution de chaque méthode appelée **(/1)**..

- j) Sécuriser l'accès à **listerDatacenter()** en utilisant un filtre d'authentification **(/1.5)**.

Bon travail.