

EXAMEN

Semestre : 1 ☒ 2

Session : Principale

Module : **Architecture des SI II (Spring)**
 Enseignant(s) : **UP ASI (Spring Team)**
 Classe(s) : **4^{ème} SAE.**

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

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

Date : **04-01-2023**

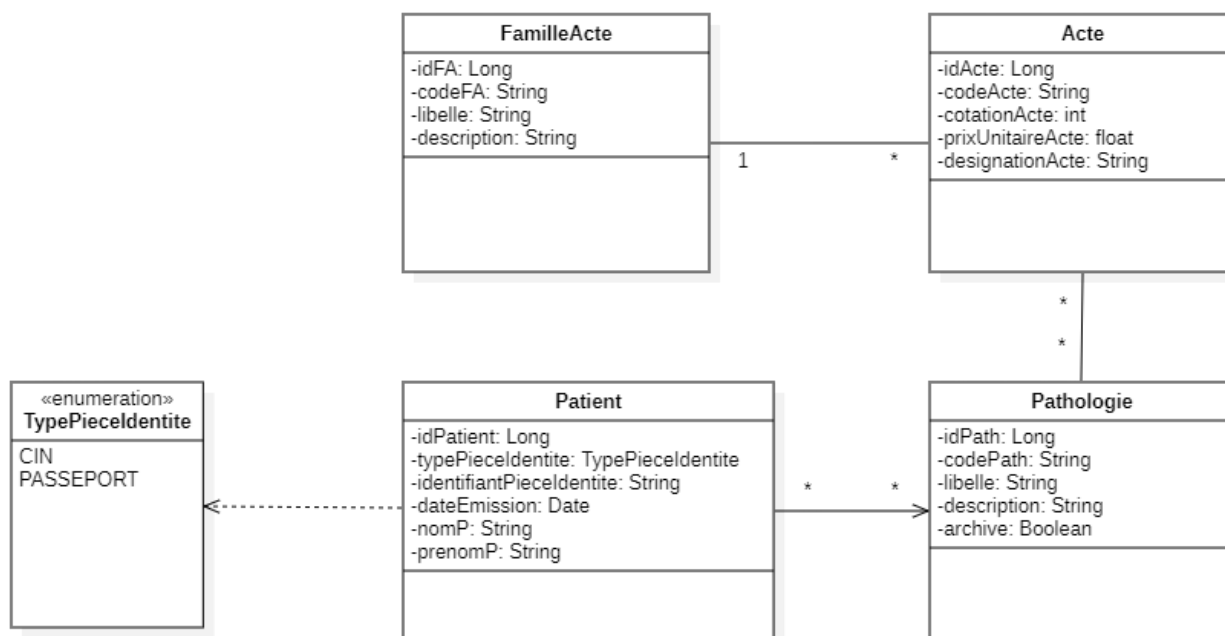
Heure **9h**

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 simplifiée de gestion des pathologies, des Actes et des Patients :

Ci-dessous, le diagramme de classes.



I. (6 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 tels que : id=1,2,3,4, ...etc.
- L'**énumération** doit être stockée en tant que chaîne de caractères dans la base de données.
- L'association unidirectionnelle Patient-Pathologie indique qu'un patient peut avoir plusieurs pathologies et qu'une pathologie peut être affectée à plusieurs patients.
- L'association bidirectionnelle Pathologie-Acte indique qu'une pathologie peut avoir plusieurs actes et qu'un acte peut être affecté à plusieurs pathologies.
- Dans l'association bidirectionnelle Pathologie-Acte, l'acte est le Child.

II. (14 points)

Pour chacune des questions suivantes, développer le code nécessaire dans une classe annotée par **@RestController** qui fait appel aux différents services.

N.B : Chaque question doit être testée sur Postman ou Swagger.

- 1) Ajouter **2 Pathologies** ayant les détails ci-dessous en respectant la signature suivante **(/1)** :

```
public Pathologie ajouterPathologie(Pathologie path);
```

| codePath | libelle | description |
|----------|----------------------|-----------------------|
| R10 | Douleurs abdominales | À vérifier |
| J20 | Bronchite aigue | Appareil respiratoire |

N.B : Par défaut, les pathologies sont non archivées.

- 2) Ajouter **2 patients**, ayant les détails ci-dessous, en les affectant aux pathologies correspondantes et en respectant la signature suivante **(/2)** :

```
public Patient ajouterPatientEtAffecterAPathologie(Patient p, String codePath);
```

| typePiece Identite | identifiant Pieceldentite | dateEmission | nomP | prenomP |
|-----------------------|------------------------------|--------------|-------|---------|
| CIN | 09887755 | 2022-08-22 | Ahmed | Atiya |
| PASSEPORT | W142516 | 2019-05-15 | Semia | Salem |

| Patient | Pathologie |
|-------------|----------------------|
| Ahmed Atiya | Douleurs abdominales |
| Semia Salem | Bronchite aigue |

- 3) Ajouter **2 Familles d'actes** et leurs actes associés ayant les détails ci-dessous en respectant la signature suivante. L'ajout des actes se fait par cascade. **(/2)** :

```
public FamilleActe ajouterFamilleActeEtActeAssocie(FamilleActe acte);
```

| codeFA | libelle | description |
|--------|--------------|-----------------------|
| R | Gastrologie | Appareil digestif |
| J | Pulmonologie | Appareil respiratoire |

| codeActe | cotationActe | prixUnitaireActe | designationActe | FamilleActe |
|----------|--------------|------------------|---------------------------------|--------------|
| MCC200 | 9 | 10 | Echographie simple en mode A | Gastrologie |
| BKA020 | 4 | 20 | Prélèvements mucus pour analyse | Pulmonologie |

- 4) Créer un service qui permet d'affecter un acte à une pathologie tout en s'assurant que cette pathologie n'est pas archivée en respectant la signature suivante (/2) :

```
public void affecterActeAPathologie(String codeActe , String codePathologie);
```

| codeActe | codePathologie |
|----------|----------------|
| MCC200 | J20 |
| BKA020 | R10 |

- 5) Calculer la facture associée à une pathologie non archivée. La facture est la somme des tarifs des différents actes pour la pathologie en question (/2.5) :

Formule pour une pathologie :

$$F = \text{CotationActe1} * \text{PrixUnitaireActe1} + \dots + \text{CotationActeN} * \text{PrixUnitaireN}$$

```
public float calculerFacture(String identifiant);
```

N.B : le paramètre « identifiant » est soit le codePath ou le libelle de la pathologie.

- 6) Créer un service qui se déclenche automatiquement permettant d'afficher le nombre d'actes pour chaque pathologie au niveau de la console en utilisant la journalisation « **logger** » tout en garantissant un écart de **30 secondes** entre 2 exécutions successives et respectant la signature suivante au niveau de **service** (/3) :

```
public void calculerNombreActesParPathologie();
```

- 7) Créer un **Aspect** qui permet d'afficher le nom des méthodes qui se trouve dans la couche service après l'exécution et qui ont un paramètre de type « String » (/1.5).

Bon travail

