

# Évaluation Technique Spring Boot - 1h30

## Système de commentaires et notations pour les profils

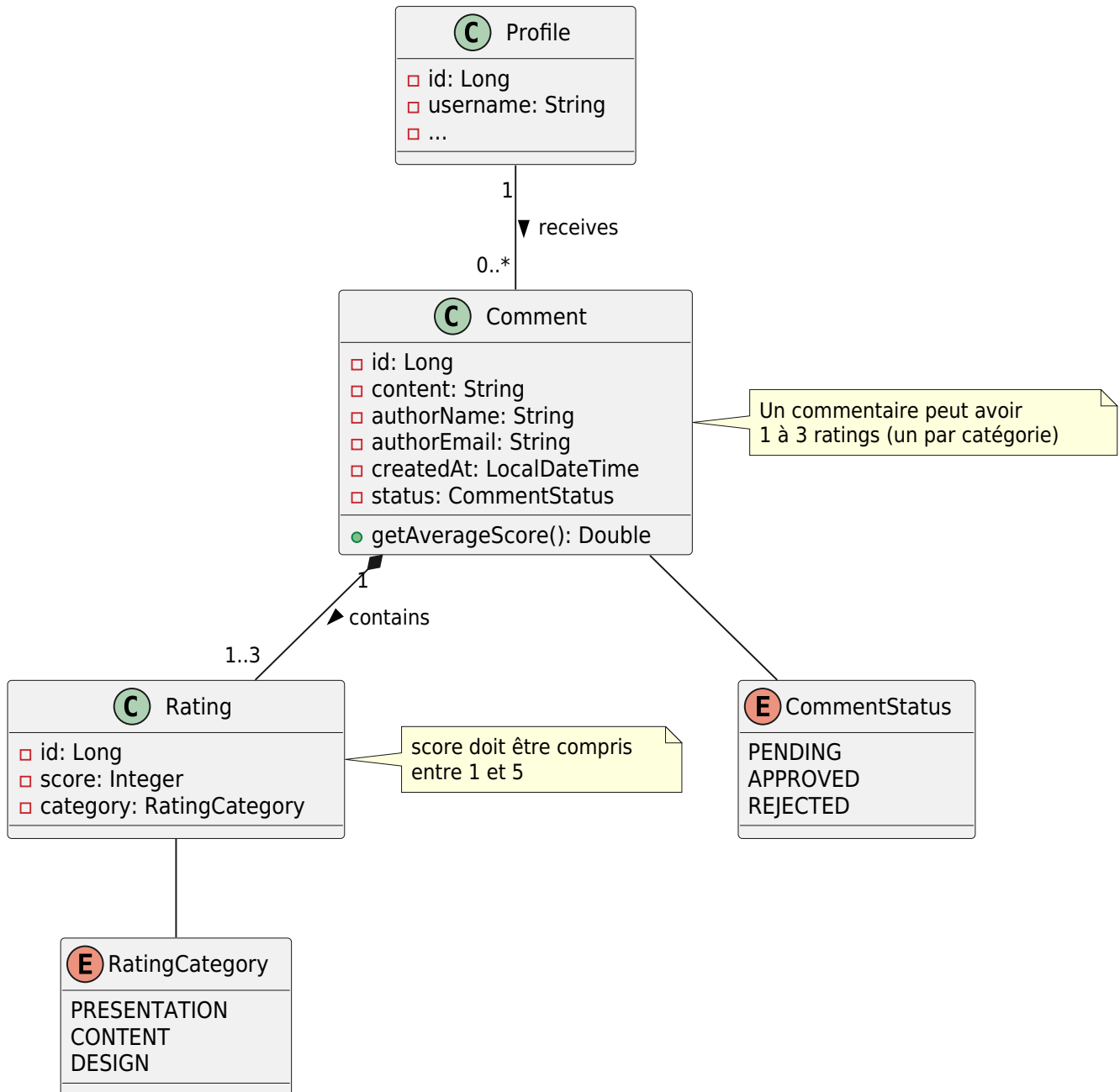
### Contexte

Les utilisateurs du Portfolio-builder souhaitent pouvoir recevoir des retours constructifs sur leurs profils de la part de visiteurs ou recruteurs. Vous allez implémenter un système de commentaires avec notation par catégories (présentation, contenu, design).

---

### Modèle de données

### Diagramme PlantUML



## Travail demandé

### Partie 1 : Modèle de données (30 min)

#### 1. Créer l'enum `CommentStatus`

- Valeurs : `PENDING`, `APPROVED`, `REJECTED`

#### 2. Créer l'enum `RatingCategory`

- Valeurs : `PRESENTATION`, `CONTENT`, `DESIGN`

#### 3. Créer l'entité `Comment`

Attributs :

- `id` : identifiant auto-généré

- content : texte de 500 caractères maximum, obligatoire
- authorName : nom du commentateur, obligatoire
- authorEmail : email du commentateur, obligatoire
- createdAt : date/heure de création
- status : statut du commentaire (enum)

Relations :

- un commentaire appartient à un profil
- un commentaire a plusieurs notes
- les ratings ne peuvent pas exister sans commentaire

#### 4. Créer l'entité Rating

Attributs :

- id : identifiant auto-généré
- score : note de 1 à 5, obligatoire
- category : catégorie de notation (enum)

Relations :

- une note appartient à un commentaire
- 

## Partie 2 : Repositories (10 min)

#### 5. Créer CommentRepository

- Hérite de JpaRepository<Comment, Long>
- Méthode personnalisée : trouver les commentaires par ID de profil ET statut

#### 6. Créer RatingRepository

- Hérite de JpaRepository<Rating, Long>
- Méthode personnalisée : trouver toutes les notes d'un commentaire

## Partie 3 : DTOs (15 min)

#### 7. Créer RatingDTO

Attributs :

- category
- score

Validations :

- category : non null
- score : entre 1 et 5

#### 8. Créer CreateCommentRequest

Attributs :

- content

- authorName
- authorEmail
- ratings (liste de RatingDTO)

Validations :

- content : non vide, max 500 caractères
- authorName : non vide
- authorEmail : format email valide
- ratings : non vide

### 9. Créer CommentDTO

Attributs :

- id
- content
- authorName
- createdAt
- status
- ratings
- averageScore

<note>Pas de validation nécessaire (DTO de réponse)</note>

## Critères d'évaluation

Critère	Points
<b>Entités</b> : annotations JPA correctes, relations bidirectionnelles cohérentes	/4
<b>Repositories</b> : déclarations correctes avec méthodes personnalisées	/3
<b>DTOs</b> : structure et validations appropriées	/3

From:  
<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:  
<http://slamwiki2.kobject.net/sio/bloc2/2a/td4?rev=1764549431>

Last update: **2025/12/01 01:37**

