Diagramme de classe

BaseEntity

Classe abstraite contenant les attributs de base communs à toutes les entités du système.

- id : identifiant unique au format UUID4.
- created_at : date de création.
- updated_at : date de dernière modification.

Toutes les entités héritent de cette classe pour garantir une structure uniforme et simplifier la sérialisation ou la traçabilité des objets.

User

Représente un utilisateur inscrit sur la plateforme HBNB.
Il peut créer des annonces, laisser des avis, et selon son statut (is_admin), accéder à des fonctions avancées.

Attributs :

- first_name, last_name, username, email, password : informations personnelles et d'identification.
- is_admin : indique si l'utilisateur est un administrateur (booléen).

Méthodes :

- register() : enregistrement de l'utilisateur.
- update_profile() / del_profile() : gestion du compte.
- is_admin() : vérification du statut d'administrateur.

Place

Représente un logement mis en ligne sur la plateforme.
Cette entité contient à la fois des informations descriptives, géographiques et économiques sur le bien proposé.

Attributs :

- country, city, address : localisation administrative du logement.
- housing_type, room_count : caractéristiques physiques du bien.
- description : présentation libre du logement.
- price : prix de location (corrigé en float pour plus de précision).
- latitude, longitude : coordonnées géographiques exactes du logement.

Méthodes :

- create_place(), update_place(), delete_place() : gestion CRUD du logement.
- list_places() : affichage ou filtrage des biens disponibles.

Review

Représente un avis laissé par un utilisateur sur un logement.

Attributs :

- place_id : identifiant du logement concerné.
- user_id : identifiant de l'auteur de l'avis.
- rating : note attribuée.
- commentary : commentaire textuel.

Méthodes :

- create_review(), update_review(), delete_review() : gestion des avis.
- list_reviews() : affichage des avis liés à un logement.

Amenity

Représente un équipement ou service proposé avec un logement.

Attributs :

- type : nom/type de l'équipement (ex. Wi-Fi, Piscine).
- description : détail de l'équipement.

Méthodes :

- create_amenity(), update_amenity(), delete_amenity() : gestion des équipements.
- list_amenity() : affichage des équipements disponibles.

Relations entre les entités

Les entités User, Place, Review et Amenity héritent toutes de la classe parente BaseEntity, qui définit une structure commune (identifiant unique, date de création, date de mise à jour).
Les relations entre ces entités reflètent les règles métier fondamentales du système HBNB.

1. BaseEntity → Tous les modèles (héritage)

Relation : Généralisation (<|--)

Description : Toutes les entités du système héritent de la classe abstraite BaseEntity, qui fournit les attributs communs id, created_at et updated_at. Cette relation structure l'ensemble du modèle métier en garantissant une base uniforme.

2. User → Place

Relation : Agrégation (◇)
Multiplicité : Un User peut posséder 1 ou plusieurs Place (1..*), chaque Place est associé à un seul User (1).
Justification : Les utilisateurs créent les logements. Un logement ne peut exister sans auteur, mais sa suppression ne nécessite pas la suppression de son créateur.

3. User → Review

Relation : Agrégation (◇)
Multiplicité : Un User peut créer plusieurs Review (1..*), chaque Review appartient à un seul User (1).
Justification : Un utilisateur peut laisser plusieurs avis. Si un utilisateur est supprimé, les avis peuvent être conservés (système de traçabilité, anonymisation).

4. Place → Review

Relation : Composition (●)
Multiplicité : Un Place peut contenir plusieurs Review (1..*), chaque Review concerne un seul Place (1).
Justification : Les avis n'ont de sens que s'ils sont liés à un logement existant. Si un logement est supprimé, ses avis sont également supprimés.

5. Place → Amenity

Relation : Agrégation (◇)
Multiplicité : Un Place peut avoir plusieurs Amenity (0..*), chaque Amenity peut être partagé par plusieurs Place (*).
Justification : Un logement peut être équipé de plusieurs commodités. Ces équipements sont réutilisables d'un logement à l'autre (ex. Wi-Fi, Jardin), sans être dépendants d'un logement spécifique.

6. User → Amenity

Relation : Agrégation (◇)
Multiplicité : Un User peut créer plusieurs Amenity (0..*), chaque Amenity a un seul créateur (1).
Justification : Certaines commodités peuvent être ajoutées manuellement par les utilisateurs (par exemple des équipements personnalisés). Leur durée de vie n'est pas liée à celle du créateur, car elles peuvent être réutilisées ailleurs.