

# Explanatory Notes for Task1 : Class Diagram

## Entity Descriptions

### 1. User

**Role in the System:** Represents individuals who interact with the HBnB platform. Users can register, update their profiles, and are identified as regular users or administrators.

**Key Attributes:**

- ID: A unique identifier for each user (UUID4).
- First\_name: The user's first name.
- Last\_name: The user's last name.
- Email: The user's email address, used for communication and identification.
- Password: Encrypted password for user authentication.
- Is\_admin: A boolean flag to distinguish between regular users and administrators.
- created\_at, updated\_at: Timestamps for record creation and updates, for auditing purposes.

**Key Methods:**

- register\_profile(): Allows new users to register on the platform.
- update\_profile(): Enables users to update their information.
- delete\_profile(): Allows users or administrators to delete a user profile.

### 2. Place

**Role in the System:** Represents properties or accommodations listed by users. These places can be rented or visited by other users.

**Key Attributes:**

- ID: A unique identifier for each place (UUID4).
- Title: A brief title or name for the property.
- Description: Detailed information about the property.
- Price: Cost per night or stay.
- Latitude, Longitude: Geographic coordinates to locate the place.
- created\_at, updated\_at: Timestamps for record creation and updates.

**Key Methods:**

- created\_place(): Allows users to list a new place.
- updated\_place(): Enables updating information about an existing place.
- deleted\_place(): Removes a place from the listings.
- list\_place(): Lists all the available places on the platform.

### 3. Review

**Role in the System:** Allows users to leave feedback on places they have visited, helping others in making informed decisions.

**Key Attributes:**

- ID: A unique identifier for each review (UUID4).
- Rating: A numeric rating (e.g., 1 to 5 stars).
- Comment: A text-based review of the place.
- created\_at, updated\_at: Timestamps for record creation and updates.

**Key Methods:**

- created\_review(): Allows users to submit a review for a place.
- deleted\_review(): Removes a review from the platform.

- `listed_by_place_review()`: Lists all reviews associated with a specific place.

#### 4. **Amenity**

**Role in the System:** Represents services or features that can be associated with a place, such as Wi-Fi, parking, or a pool.

**Key Attributes:**

- **ID:** A unique identifier for each amenity (UUID4).
- **Name:** The name of the amenity.
- **Description:** A description of the amenity.
- **created\_at, updated\_at:** Timestamps for record creation and updates.

**Key Methods:**

- `created_amenity()`: Allows users to create a new amenity.
- `updated_amenity()`: Enables updates to existing amenities.
- `deleted_amenity()`: Removes an amenity from the list.
- `listed_amenity()`: Lists all available amenities on the platform.

## Relationships Between Entities

1. **User** and **Place**  
A **User** can own multiple **Places** (1 User -- \* Place). This association reflects the ownership of the property by the user, who can manage the place listings, such as adding, updating, or deleting their properties.
2. **User** and **Review**  
A **User** can leave multiple **Reviews** on places they have visited (1 User -- \* Review). This represents the user's ability to provide feedback for various properties, influencing other users' decisions.
3. **Place** and **Review**  
A **Place** can receive multiple **Reviews** (1 Place -- \* Review). This relationship highlights how each place can accumulate reviews from different users, affecting its reputation and popularity.
4. **Place** and **Amenity**  
A **Place** can have multiple **Amenities** (1 Place o-- \* Amenity). This composition signifies that a place can offer a variety of services, making it more attractive to potential renters. However, if a place is deleted, its associated amenities remain intact in the system.

## Contribution to Business Logic

These relationships allow the platform to function cohesively. **Users** create and manage **Places**, which attract other users based on the **Reviews** and **Amenities** offered. **Reviews** add credibility and help rank **Places** based on user experiences, while **Amenities** make the **Places** more appealing by providing additional value, such as comfort and convenience. This interconnected logic ensures that the system provides a fluid user experience, from registration and profile management to property listing, booking, and reviewing.