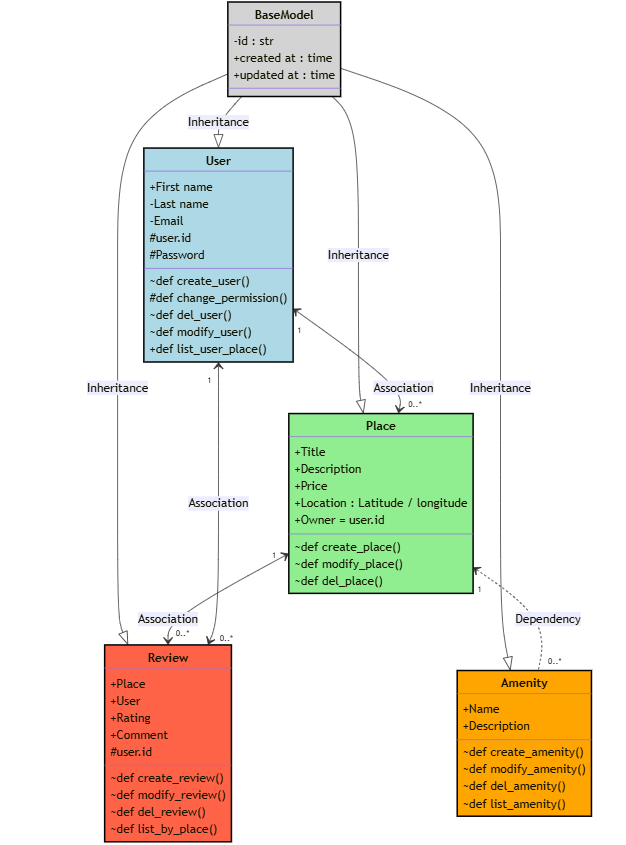
HBnB Class Diagram Documentation

# 1. Class Diagram Overview



## 1.1 Description

The Class Diagram provides a structured representation of the system’s data model, illustrating how different entities interact within the application.

## 1.2 Main Classes and Relationships

* BaseModel: A parent class that provides common attributes (ID, timestamps) for all entities.
* User: Represents registered users with authentication and management functionalities.
* Place: Stores property details such as location, price, and owner.
* Amenity: Represents features available at a property.
* Review: Allows users to rate and comment on places.

## 1.3 Design Decisions

* Inheritance: The BaseModel class ensures consistency across all entities.
* Associations: User, Place, and Review classes maintain clear relationships to support data integrity.

# 2. Detailed Explanation of Classes

## 2.1 BaseModel

* The superclass that provides common fields (id, created\_at, updated\_at) to all entities.
* Ensures uniformity and reusability across different data models.

## 2.2 User

* Attributes: first\_name, last\_name, email, password, user.id
* Methods:
* - create\_user(): Registers a new user.
* - change\_permission(): Modifies user roles.
* - del\_user(), modify\_user(): Manage user profiles.
* - list\_user\_place(): Retrieves places owned by the user.
* Relationships:
* - Can create Places (One-to-Many relationship).
* - Can submit Reviews (One-to-Many relationship).

## 2.3 Place

* Attributes: title, description, price, latitude, longitude, owner = user.id
* Methods:
* - create\_place(), modify\_place(), del\_place(): CRUD operations for places.
* Relationships:
* - Owned by a User.
* - Can have multiple Amenities (Many-to-Many relationship).
* - Can receive multiple Reviews (One-to-Many relationship).

## 2.4 Amenity

* Attributes: name, description
* Methods:
* - create\_amenity(), modify\_amenity(), del\_amenity(): CRUD operations for amenities.
* Relationships:
* - Linked to Places through a Many-to-Many relationship.

## 2.5 Review

* Attributes: place, user, rating, comment, user.id
* Methods:
* - create\_review(), modify\_review(), del\_review(): CRUD operations for reviews.
* Relationships:
* - Associated with a User (Many-to-One relationship).
* - Associated with a Place (Many-to-One relationship).

# 3. Class Diagram Structure and Integrity

* The inheritance structure ensures all classes share common properties via BaseModel.
* One-to-Many and Many-to-Many relationships ensure logical data storage and retrieval.
* Encapsulation and abstraction principles are applied to keep operations modular and maintainable.

# 4. Conclusion

This document provides a detailed analysis of the HBnB Class Diagram, explaining the relationships and functionalities of each entity. Understanding these components ensures a clear and scalable implementation of the system’s business logic.