# HBnB - UML

## Introduction

The HBnB project is a simplified version of an AirBnB-like application. In **Part 1**, we do not write code — we focus only on planning the system. The goal is to create clear technical documentation that explains how the application will work.

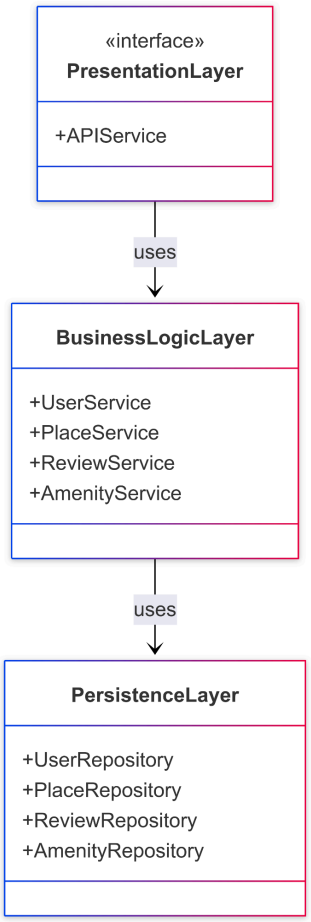
This document includes:

- A high-level view of the system architecture (3-layer structure)
- A detailed class diagram for the core logic (users, places, reviews, amenities)
- Sequence diagrams that show how different parts of the system talk to each other when users call the API

# Part 1: Technical Documentation

## 0. High-Level Package Diagram

This document provides a high-level package diagram illustrating the three-layer architecture of the HBnB application. It demonstrates how different components are organized and interact using the Facade Pattern.
Three-Layer Architecture: The application is structured into three primary layers:




1. Presentation Layer (Services, API)
    - Responsible for handling user interactions.
    - Exposes API endpoints that communicate with the Business Logic Layer.
    - Calls the Facade to simplify interactions with underlying components.

2. Business Logic Layer (Models)
    - Contains the core application logic.
    - Defines models representing system entities (e.g., User, Place, Review, Amenity).
    - Uses the Facade to abstract interactions with the Persistence Layer.

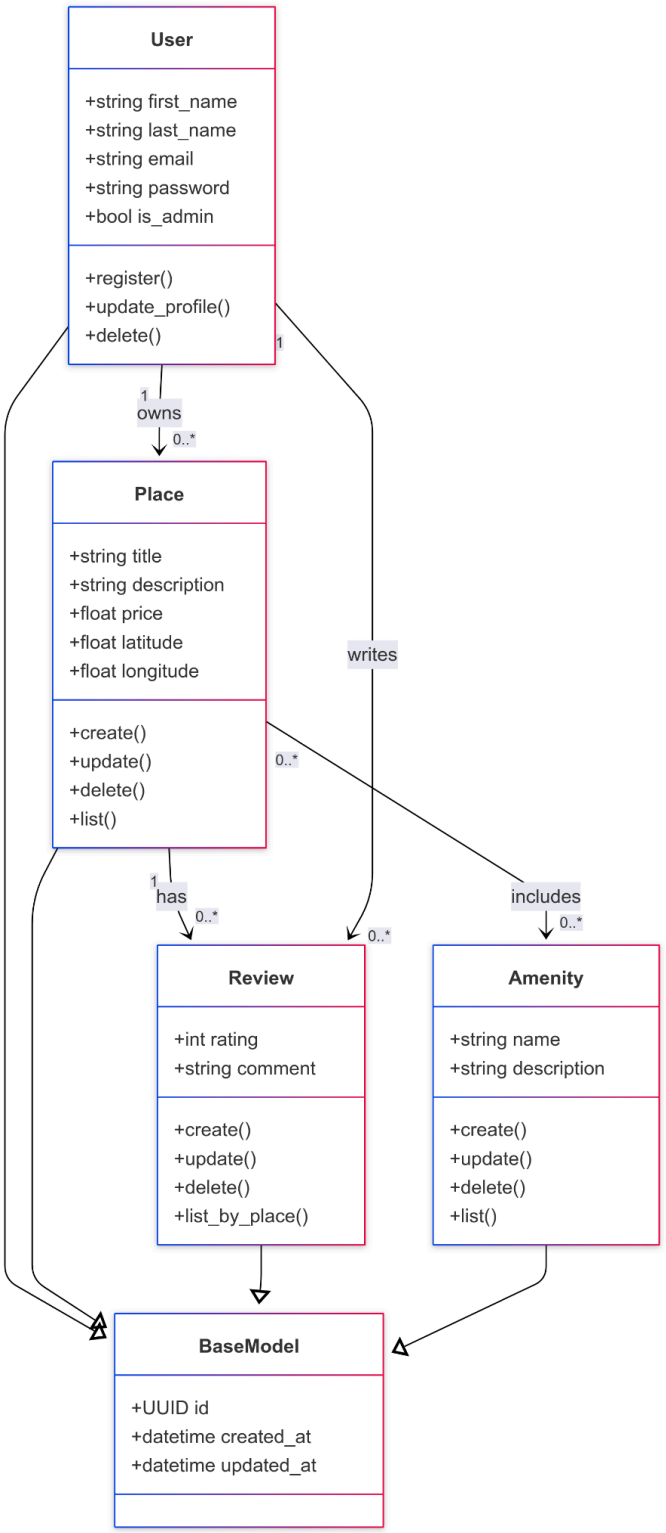
3. Persistence Layer (Database Access)
    - Manages data storage and retrieval operations.
    - Provides an interface for CRUD operations on the database.
    - Business Logic Layer interacts with this layer via the Facade.

Facade Pattern Explanation:

- The Facade Pattern is used to streamline interactions between layers:
- The Presentation Layer does not directly access database logic; instead, it communicates through the Facade in the Business Logic Layer.
- The Facade provides a simplified interface to interact with the models, ensuring separation of concerns.
- The Persistence Layer remains hidden behind the Business Logic Layer, ensuring direct database interactions are encapsulated.

## 1. Detailed Class Diagram for Business Logic Layer

This document provides a class diagram of the different class of the HBnB application. It demonstrates how different class are organized and interact with each other.



BaseModel

- BaseModel is a common class that all other classes inherit from.
- It gives each object a unique ID and stores the date of creation and last update.
- This helps us keep track of when each object was created or modified.

User

- Represents a person who uses the app.
- Each user has a name, email, password, and a flag to know if they are an admin.
- Users can register, change their profile, or delete their account.
- A user can own many places and write reviews.

Place

- A place is a property that users can list for rent.
- It has a title, description, price, and location (latitude and longitude).
- Users can create, update, delete, and list places.
- A place can have many reviews and many amenities.

Review

- A review is a comment or rating a user gives to a place.
- It includes a score (rating) and a comment.
- Users can create, update, delete reviews, and list reviews by place.

Amenity

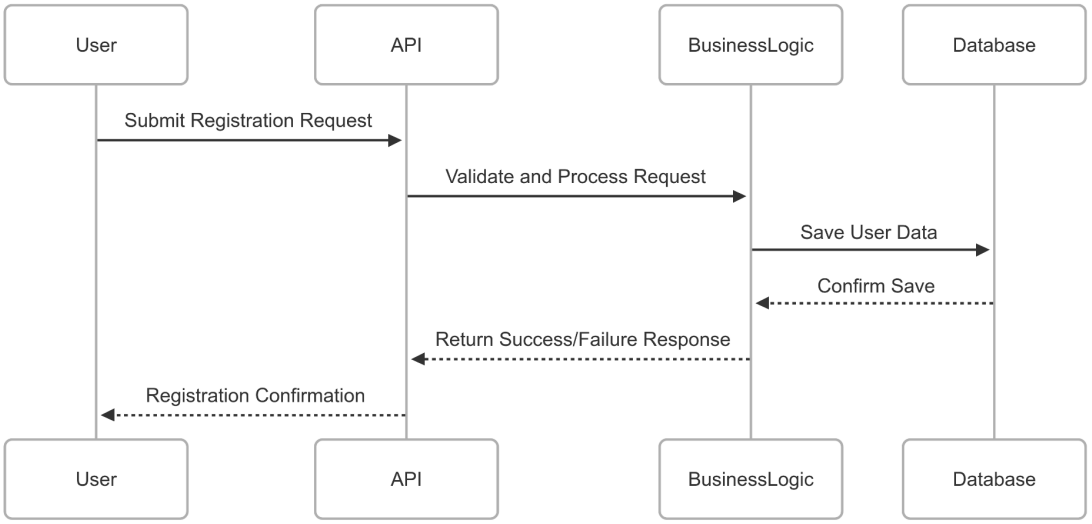
- Amenities are things that make a place better (like Wi-Fi, TV, etc.).
- Each amenity has a name and a description.
- Users can create, update, delete, and list amenities.A place can include many amenities, and an amenity can belong to many places.

Relationships Summary

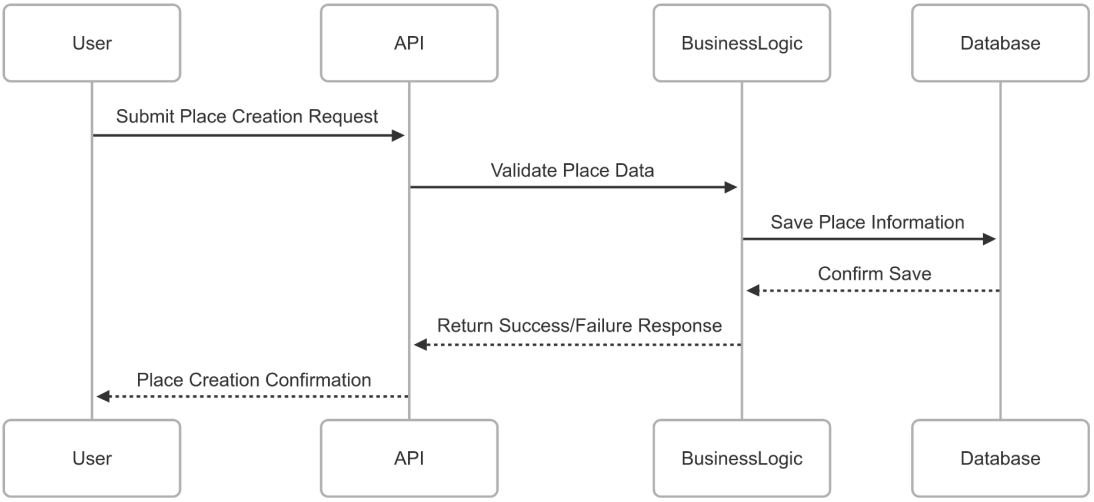
- A User can own multiple Places.
- A Place can have multiple Reviews.
- A User can write multiple Reviews.
- A Place can include multiple Amenities (and vice versa).

## 2. Sequence Diagrams for API Calls

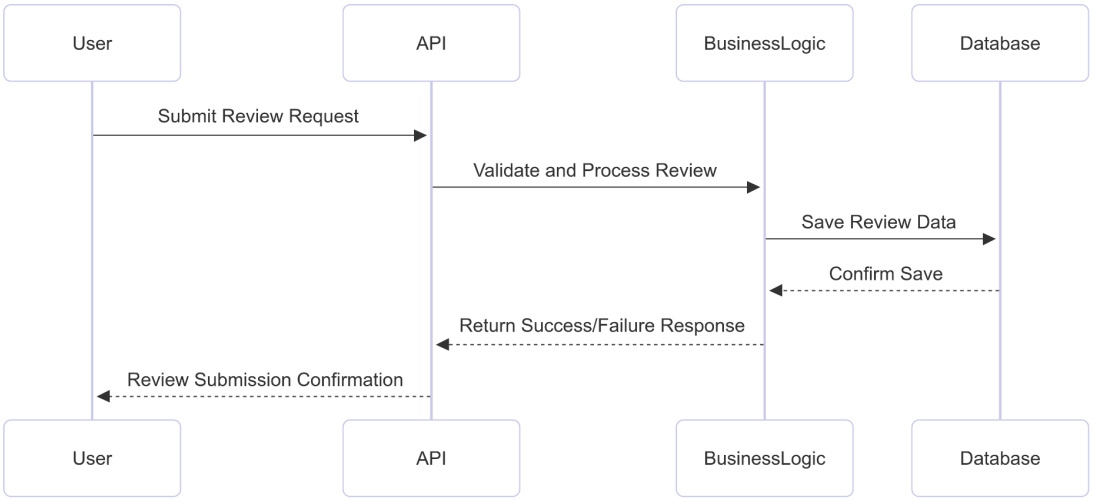
- User Registration
  A user submits registration details to create a new account.




- Place creation
  A registered user creates a new place listing



- Review submission
  A user submits a review for a specific place.




- Fetching a list of places

  A user requests a list of places based on specific criteria.