- ABDULAZIZ ALRASHDI
- ABDULRAHMAN ALGHAMDI
- ABDULLAH ALSALEM

2025

Holberton Project Hbnb

Documentation compilation

Table of Contents

Introduction

Technical Documentation of the HBnB Project!

HBnB Evolution is a simplified, AirBnB-like application designed to demonstrate the core concepts behind a property rental platform through a clean, layered architecture and well-defined business logic. The system enables four main capabilities: user management (registration and profile updates with support for administrator accounts), place management (creating and maintaining property listings with pricing and geographic coordinates), review management (users submitting ratings and comments for places), and amenity management (maintaining a catalog of amenities that can be associated with places).

This document provides the foundational technical blueprint for the HBnB Evolution application before implementation. It focuses on describing the overall architecture, the responsibilities of each layer (Presentation, Business Logic, and Persistence), and how these layers interact through the Facade pattern. It also defines the main domain entities—User, Place, Review, and Amenity—along with their attributes, relationships, and required audit information such as unique identifiers and creation/update timestamps.

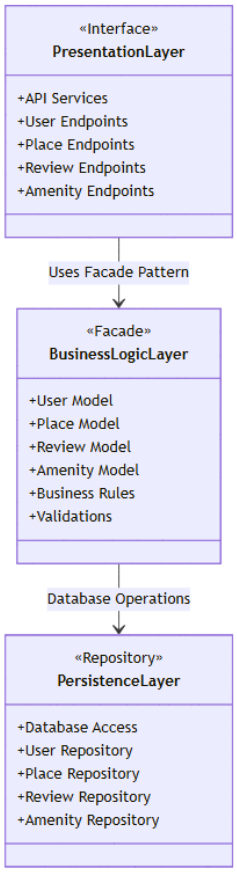
The objective of this documentation is to ensure that the design is clear, consistent, and implementation-ready for the next project phases. By formalizing the system using UML package, class, and sequence diagrams, the document helps the development team align on structure, responsibilities, and data flow, reducing ambiguity and serving as a reference throughout development and testing.

High-Level Architecture

The application is based on a well-defined three-layer architecture:

- Presentation
- Business Logic
- Persistence

This organization ensures a clear separation of responsibilities and makes long-term maintenance easier.
We also use the Facade design pattern, which provides a simplified interface to the upper layers while hiding internal complexity.

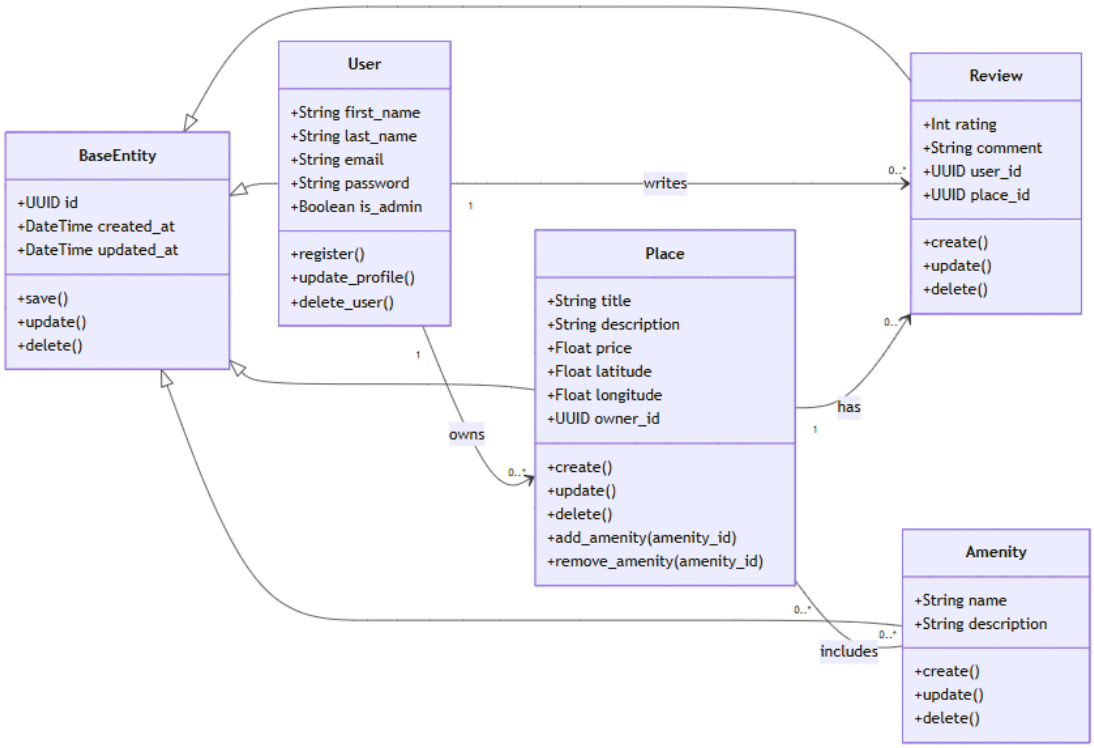


Each layer plays a specific role:

- PresentationLayer: manages the interface and data collection.
- BusinessLogicLayer: contains the business logic (validation, rules, interactions).
- PersistenceLayer: handles saving to and reading from the database.

Business Logic Layer

Business Logic. This is where we define our main entities (users, places, reviews, amenities) and their relationships.



The main design choices:

- BaseProfile serves as the parent class to factorize common fields.
- User represents our users with their personal information and role.
- Place manages the accommodations published and linked to users.
- Review allows users to leave feedback. Amenity describes the amenities associated with the places.

Business Logic Layer