

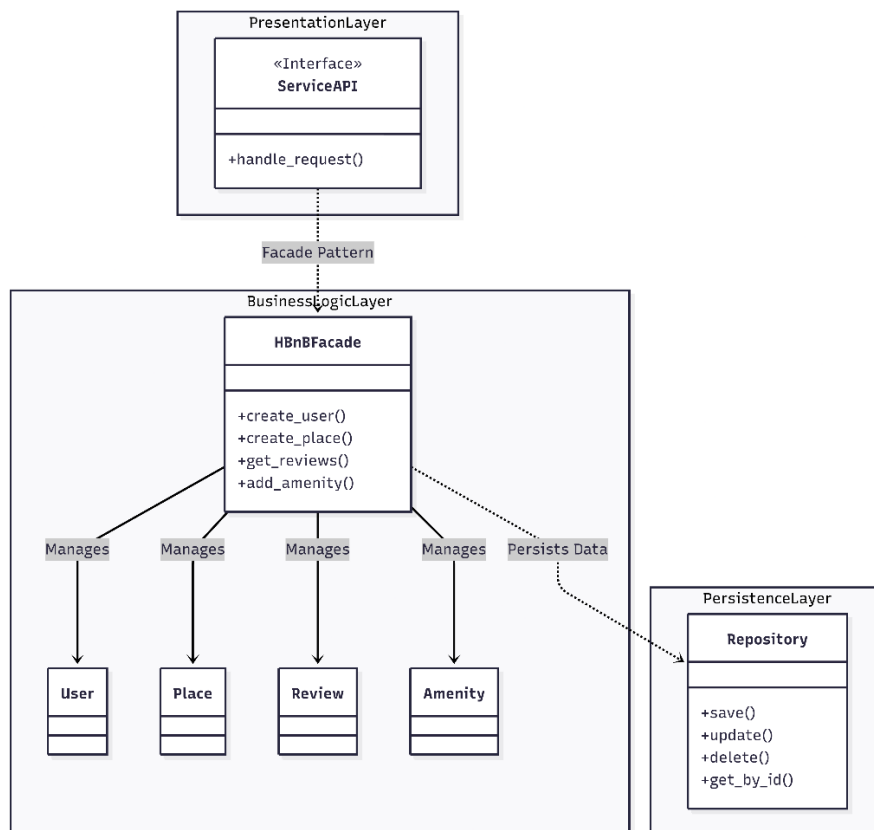
HBnB Evolution - Technical Documentation

1. Introduction

This document serves as the technical blueprint for the **HBnB Evolution** project. The goal of this application is to create a simplified AirBnB clone that allows users to manage properties (places), amenities, and reviews. This documentation outlines the system's three-layer architecture, detailed business logic design, and the flow of data through API interactions.

2. High-Level Architecture

The application follows a **Layered Architecture** pattern to enforce separation of concerns and modularity. Interaction between layers is managed via the **Facade Pattern**.



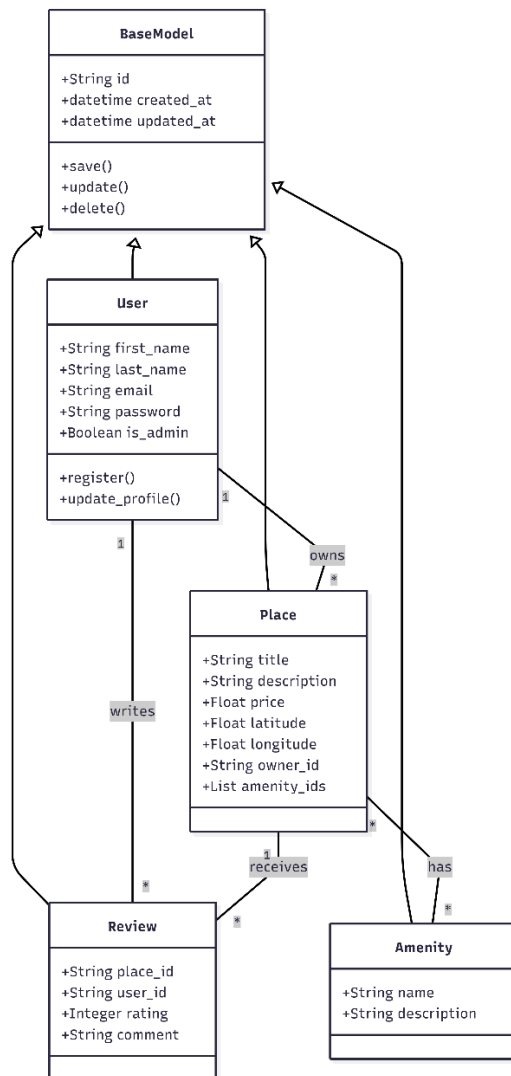
Presentation Layer: Handles API requests and user interaction. It forwards logic to the Facade.

- **Business Logic Layer:** Contains the HBnBFacade and core models (User, Place, etc.). The Facade acts as the single entry point for logic.
- **Persistence Layer:** Handles data storage and retrieval (Database/Repository).

3. Business Logic Layer

The Business Logic Layer is structured around a central BaseModel. All entities (User, Place, Review, Amenity) inherit from this base to ensure they have unique IDs and timestamps.

-



User: Can register, login, and own places.

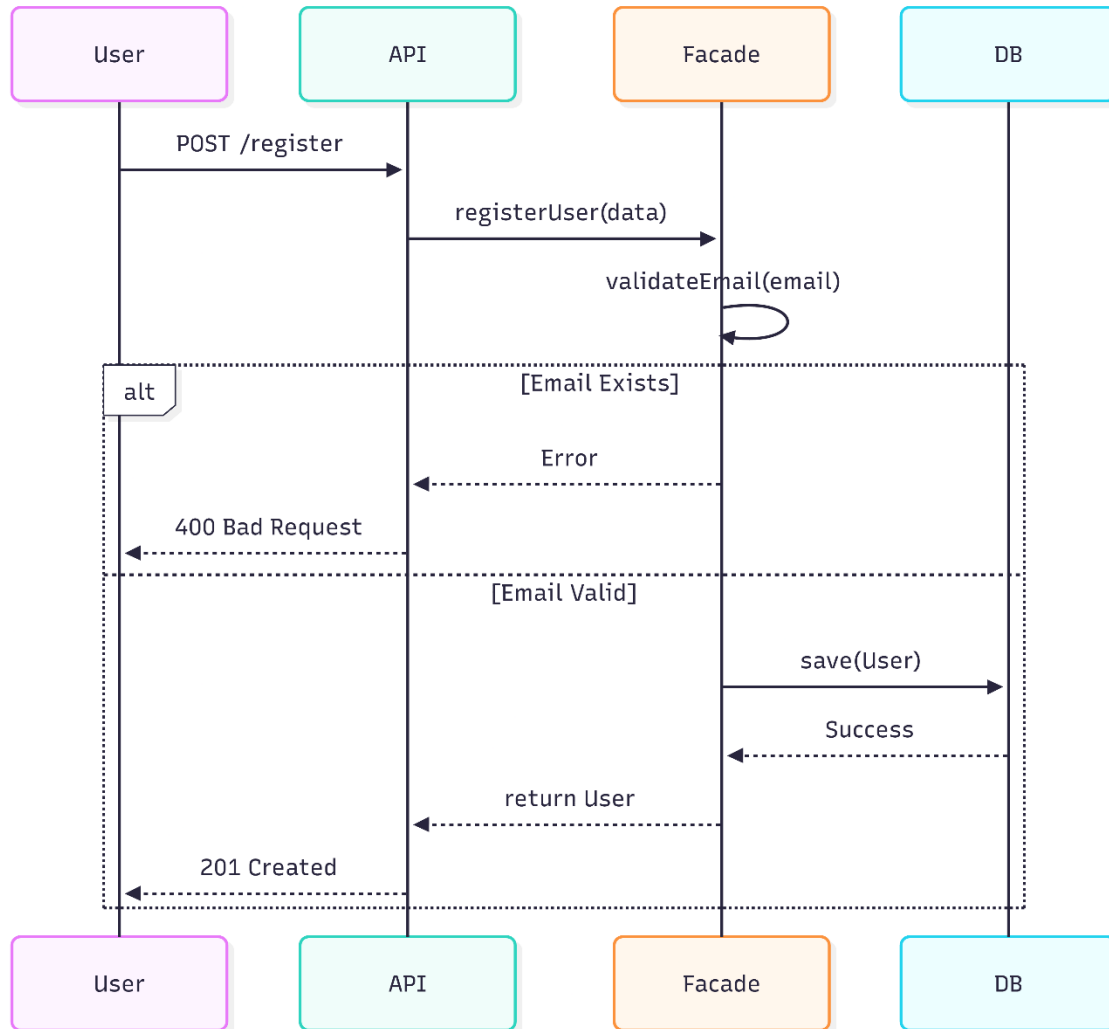
- **Place:** Properties listed by users. Can have multiple amenities.
- **Review:** Feedback left by users on specific places.
- **Amenity:** Features (like WiFi, Pool) linked to places.

4. API Interaction Flow

The following sequence diagrams illustrate how data moves through the layers for key operations.

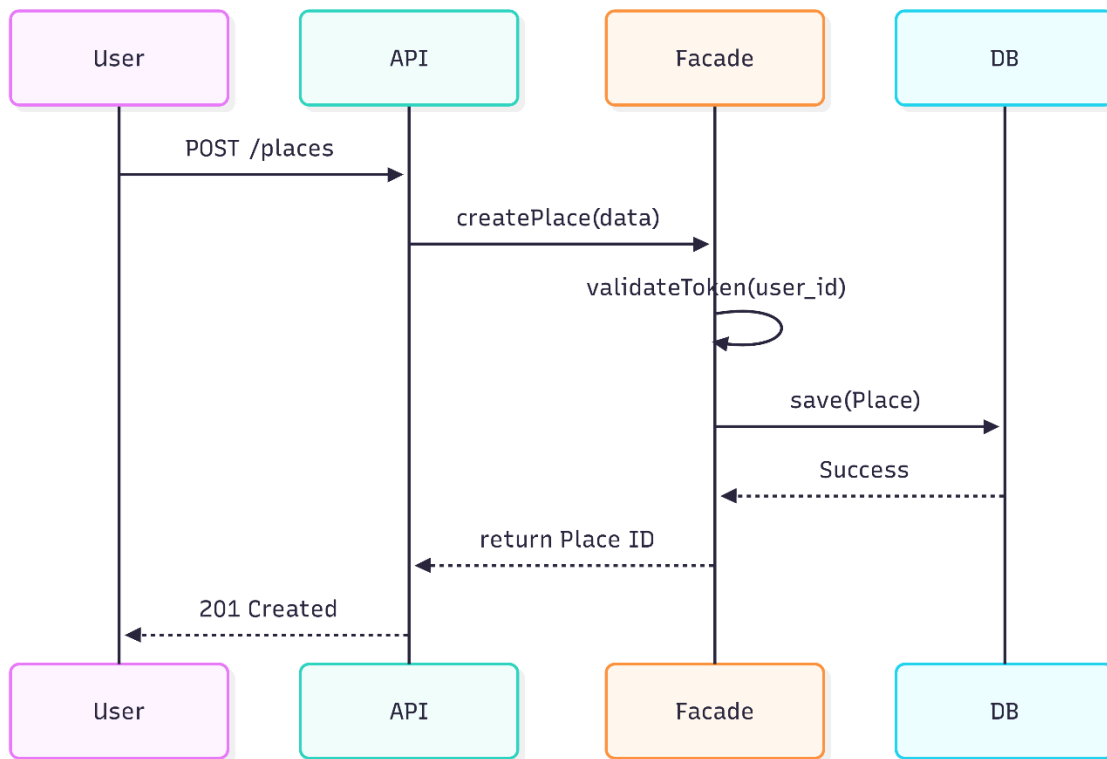
4.1 User Registration

When a user signs up, the system validates that their email is unique before creating the account.



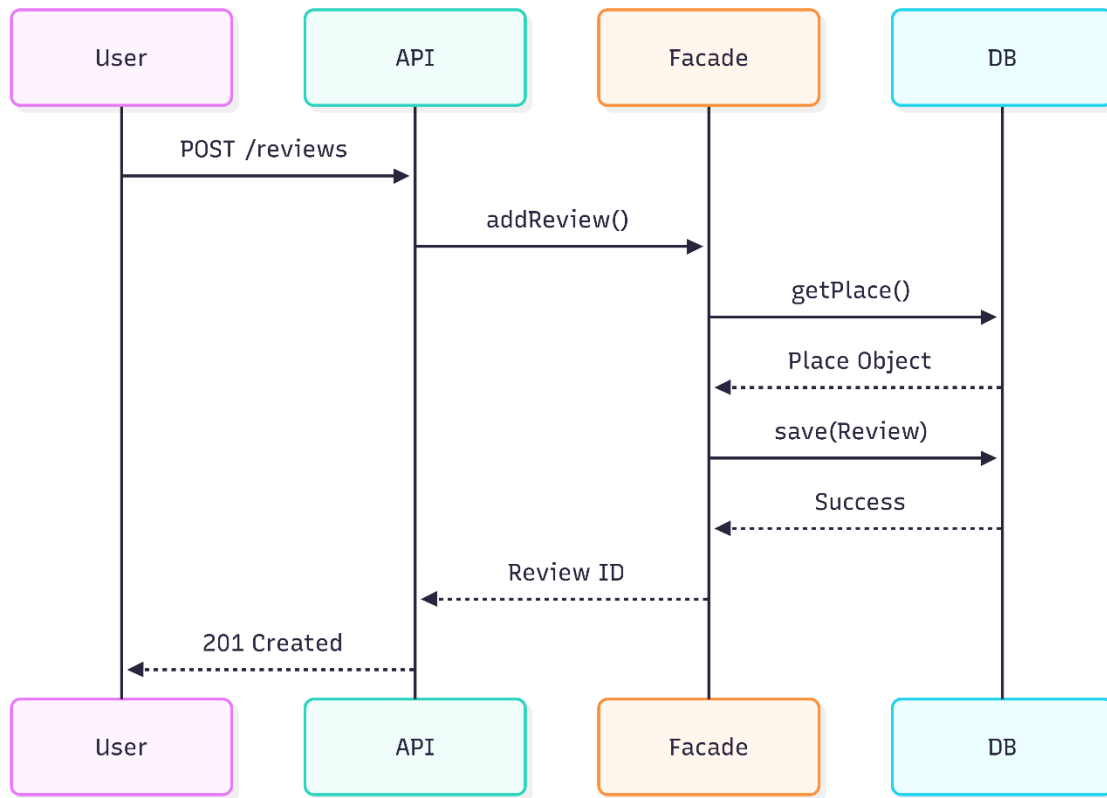
4.2 Place Creation

Authenticated users can create new listings. The system links the new place to the user's ID (owner).



4.3 Review Submission

Users can leave reviews. The system checks if the place_id exists before allowing the review to be saved.



4.4 Fetching a List of Places

This operation retrieves all available property listings from the database and returns them to the user.

