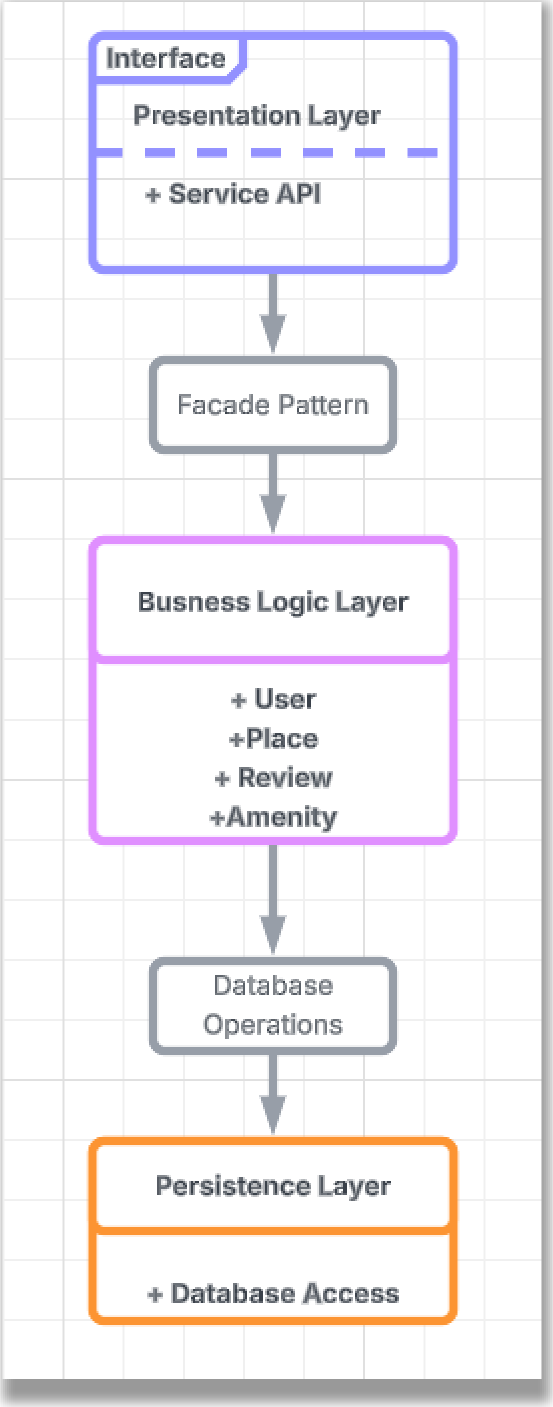
_HBnB Evolution application

By Laura Aupetit, Emi Hatano

Introduction

This document presents the system design of HBnB, a web application inspired by Airbnb. HBnB allows users to list places, book stays, and write reviews.

It includes a high-level architecture diagram, a class diagram for the business logic layer, and sequence diagrams for key API calls. These diagrams help explain how the system is structured and guide future development.



1.High-Level Package Diagram

Purpose

The purpose of this package diagram is to illustrate the high-level architecture of the HBnB Evolution application, showcasing the three main layers: Presentation, Business Logic, and Persistence. It uses the facade pattern to simplify the interactions between these layers, providing a clear visual guide for developers to understand the structure and communication flow within the application. This aids in efficient design and implementation while ensuring maintainability and scalability.

•Presentation Layer:

This includes the services and API through which users interact with the system. (Services, API endpoints)

•Business Logic Layer:

This contains the models and the core logic of the application. (Core models: User, Place, Review, Amenity).

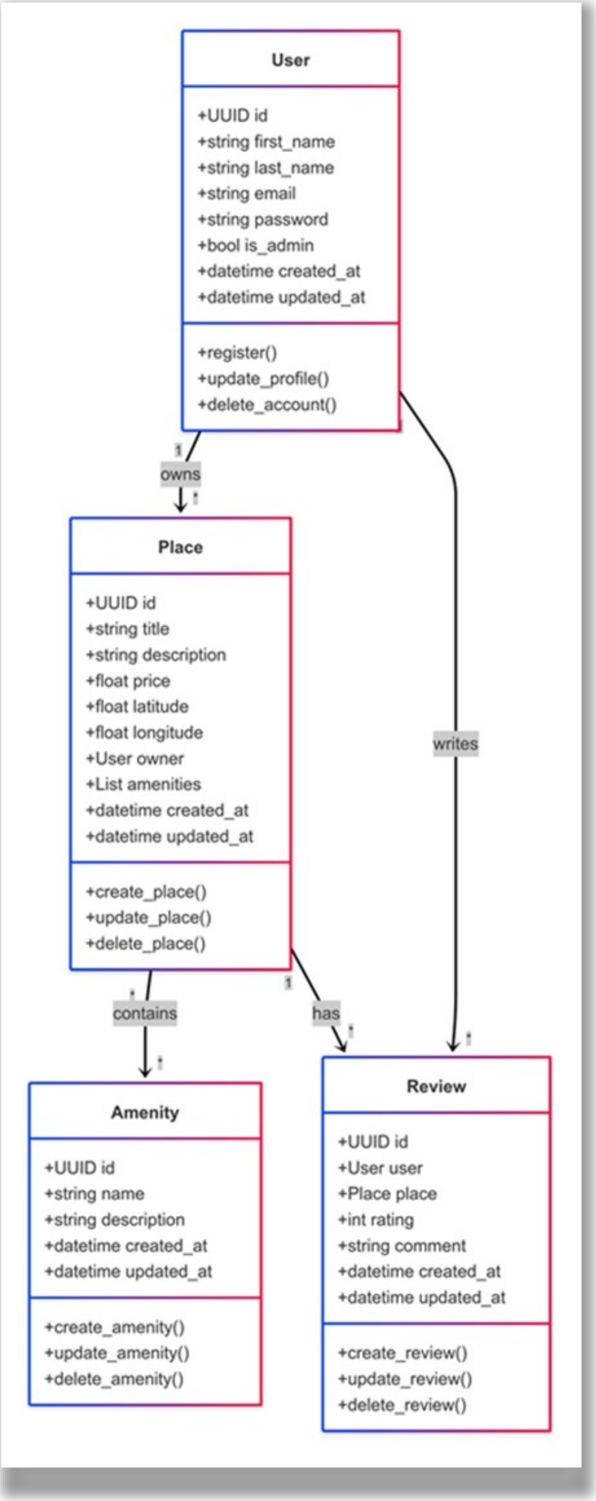
•Persistence Layer:

This is responsible for storing and retrieving data from the database. Database access objects or repositories.

•The facade pattern:

The HBnB application uses the Facade Pattern to simplify communication between the system’s layers. In this architecture, the Presentation Layer does not interact directly with the internal components of the Business Logic Layer or the Persistence Layer. Instead, it communicates through well-defined interfaces that act as a facade.

2.Detailed Class Diagram for Business Logic Layer



Purpose

The purpose of this class diagram is to define the core entities and relationships in the Business Logic Layer of the HBnB application. It provides a clear overview of key components such as User, Place, Review, and Amenity, along with their attributes, methods, and associations. This diagram serves as a foundational reference for implementing and maintaining the system's business rules and data flow.

Entities

•User:

A user has a first name, last name, email, and password, with an optional administrator status. Users can register, update their profile, and be deleted.

•Place: A place has a title, description, price, latitude, and longitude, and is owned by a user. It can include multiple amenities and supports creation, update, deletion, and listing.

•Review:

A review belongs to a specific user and place, containing a rating and comment. Reviews can be created, updated, deleted, and listed by place.

•Amenity:

An amenity has a name and description, and can be created, updated, deleted, and listed.

Relationships

• User↔Place

One user can own multiple places, representing a one-to-many relationship.

• Place ↔ Amenity

A place can have multiple amenities, and each amenity can be shared by multiple places, forming a many-to-many relationship.

• User ↔ Review

One user can write multiple reviews, which is a one-to-many relationship.

• Place↔ Review

A place can have multiple reviews, also representing a one-to-many relationship.

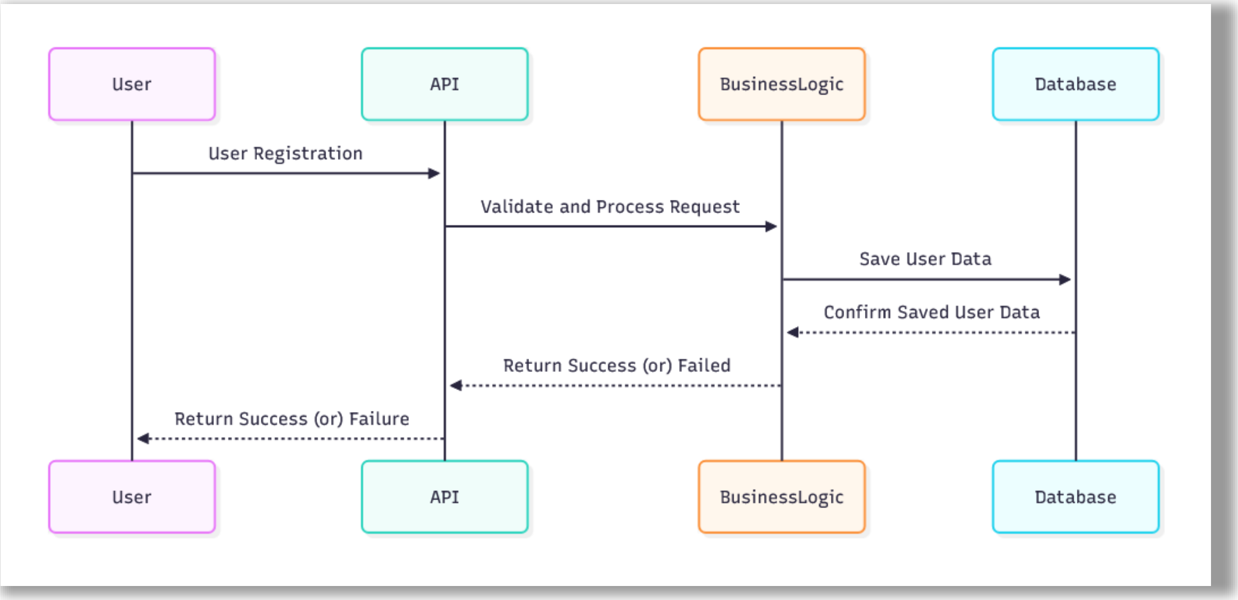
3.Sequence Diagrams for API Calls

Purpose

This section presents four sequence diagrams that illustrate how the HBnB application handles key API calls. Each diagram traces the flow of data and control between the three main layers of the system: the Presentation Layer (API), Business Logic Layer, and Persistence Layer (Database). These diagrams help visualize how each part of the system collaborates to fulfill specific user requests.

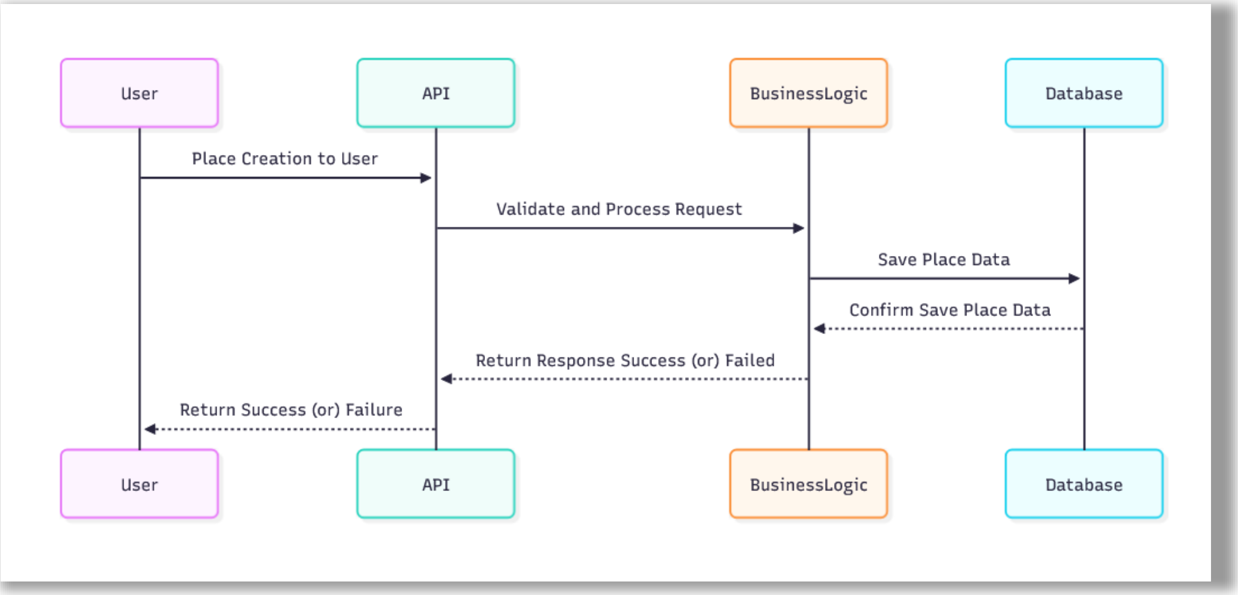
•User Registration

This diagram shows how a new user is registered. The API receives the signup request, validates the input, and passes the data to the business logic. The business logic handles user creation and stores the new user record in the database. A confirmation response is then sent back to the client.



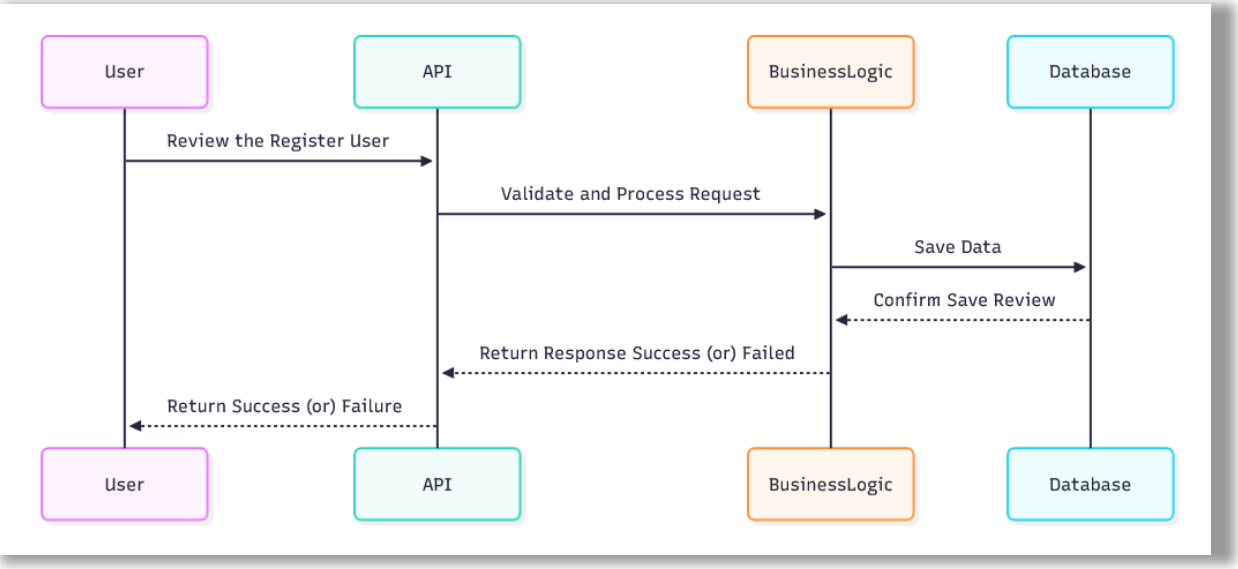
•Place Creation

This sequence represents the process of creating a new place listing. The user sends place details to the API, which forwards the data to the business layer. After validation and business rule checks, the place is saved in the database, and a success response is returned.



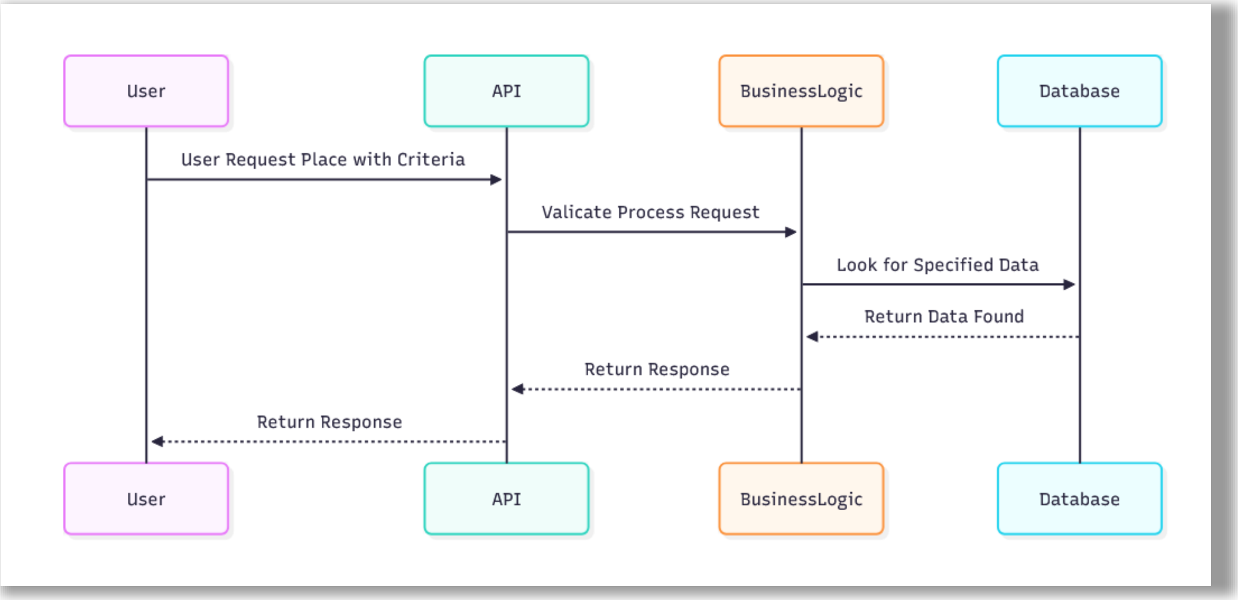
•Review Submission

This sequence shows how a user writes a review for a place. The user sends the review (including rating and comment) to the API. The API checks the data and sends it to the business logic. The business logic connects the review to the right place and user, and then saves it to the database. After saving, the API returns a confirmation to the user.



•Fetching a List of Places

This diagram outlines the flow for retrieving a list of places based on specific criteria. The user sends a request to the API, possibly including filters such as location, price range, or amenities. The API forwards the request to the business logic layer, which applies the criteria to query the database. The matching places are retrieved and returned through the API to the user.



Conclusion

This technical document outlines the architecture and design of the HBnB system. By combining a high-level overview, detailed class structure, and step-by-step API interactions, it provides a clear understanding of how different components of the application work together. This documentation serves as a solid foundation for development, ensuring consistency, scalability, and maintainability throughout the project lifecycle.