

HBnB

Purpose of the Document:

This technical document serves as a comprehensive guide for the HBnB project, detailing the system's architecture and design decisions. It aims to provide a clear reference for the implementation phase and serves as a blueprint for the development stage.

Scope of the Document:

The document includes key diagrams representing the architecture of the HBnB application, including:

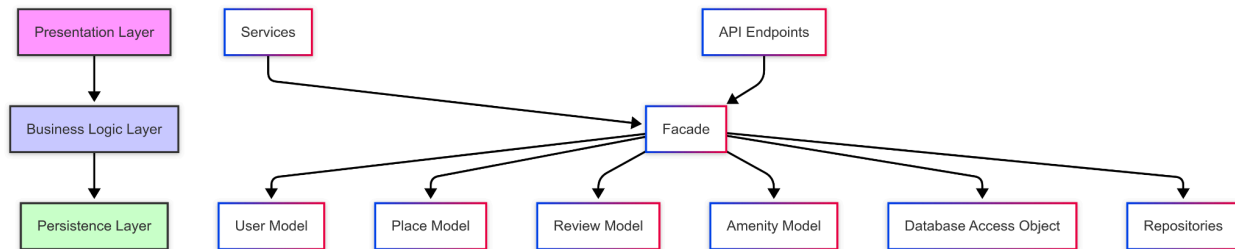
- A high-level package diagram outlining the system's layered architecture.
- A detailed class diagram for the Business Logic layer.
- Sequence diagrams that illustrate the flow of interactions for key API calls.

Each diagram is accompanied by explanatory notes to clarify design decisions, highlight key components, and explain how the system components interact.

Overview of the HBnB Project:

HBnB is an online platform similar to AirBnB, allowing users to list, book, review, and manage properties. The project is designed with scalability and modularity in mind, using a layered architecture to separate concerns and facilitate maintainability. This document focuses on the core components of the system, namely the **User Management**, **Place Management**, **Review Management**, and **Amenity Management** features, providing a thorough explanation of their interaction and structure.

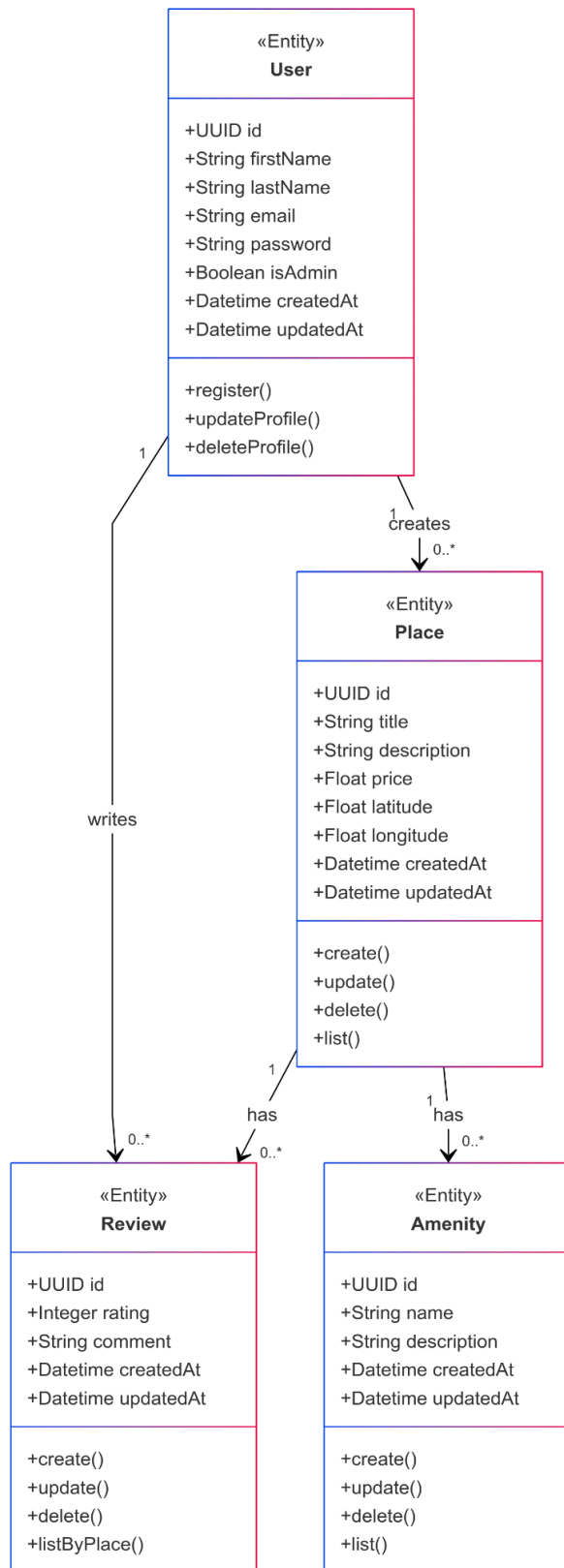
Package Diagram:



Explanation Notes:

- **Purpose:** The high-level package diagram provides an overview of the application's architecture. It breaks the system into three primary layers: presentation, business logic, and persistence. Each layer handles distinct responsibilities in the application's overall functionality.
- **Key Components:**
 - **Presentation Layer:** Handles the interaction between the user and the application. It includes APIs and services that expose the functionality of the system to the user.
 - **Business Logic Layer:** Contains the core functionality and models that define how the application works. This is where the logic for entities like users, places, and reviews resides.
 - **Persistence Layer:** Manages data storage and retrieval, interacting with databases or other data stores.
- **Design Decisions:**
 - The **Facade Pattern** is used to provide a unified interface to access the different layers. This simplifies interactions between layers by hiding complexity and allowing for a clear, consistent interface for the Presentation Layer to interact with.
 - The layered architecture enhances modularity, making the system easier to maintain and extend.

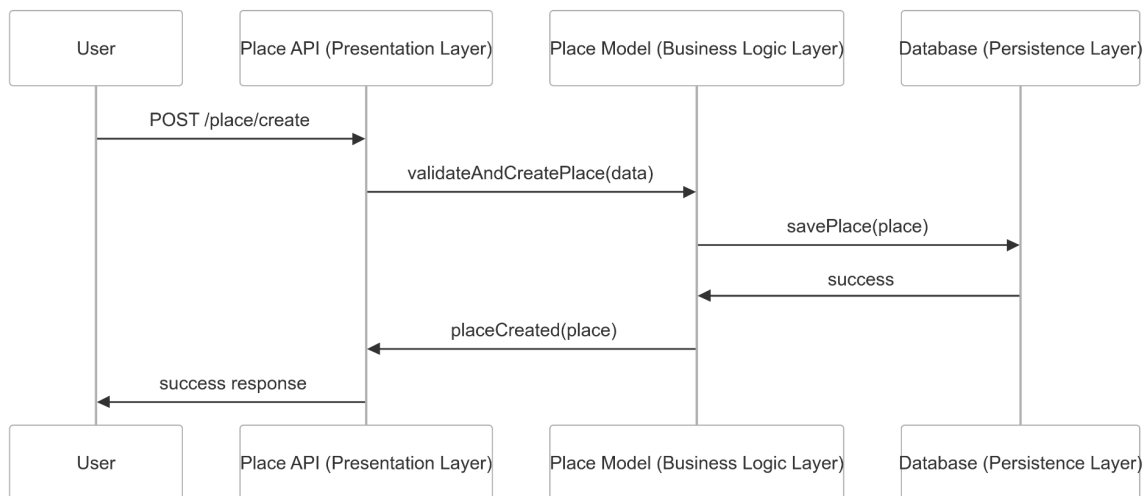
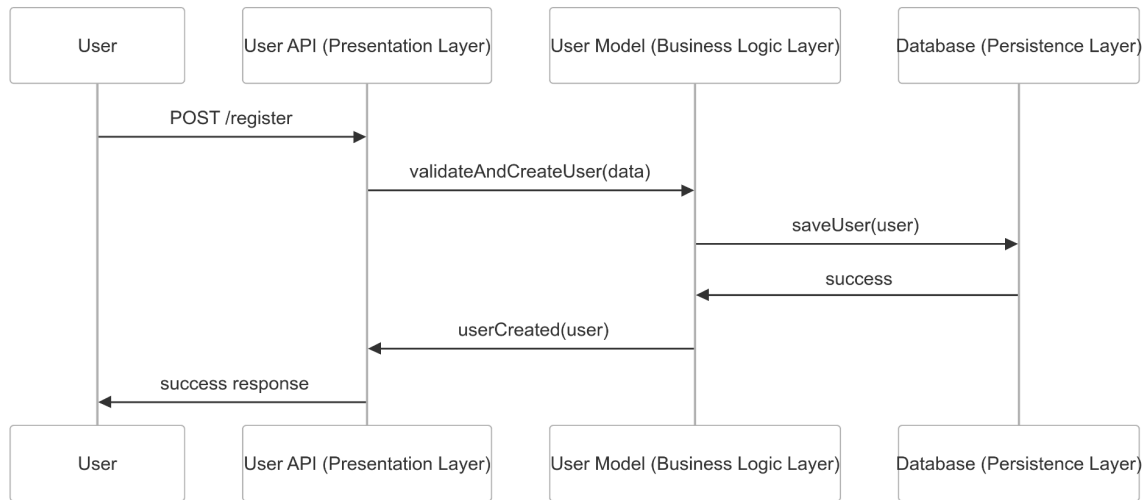
Class diagram:

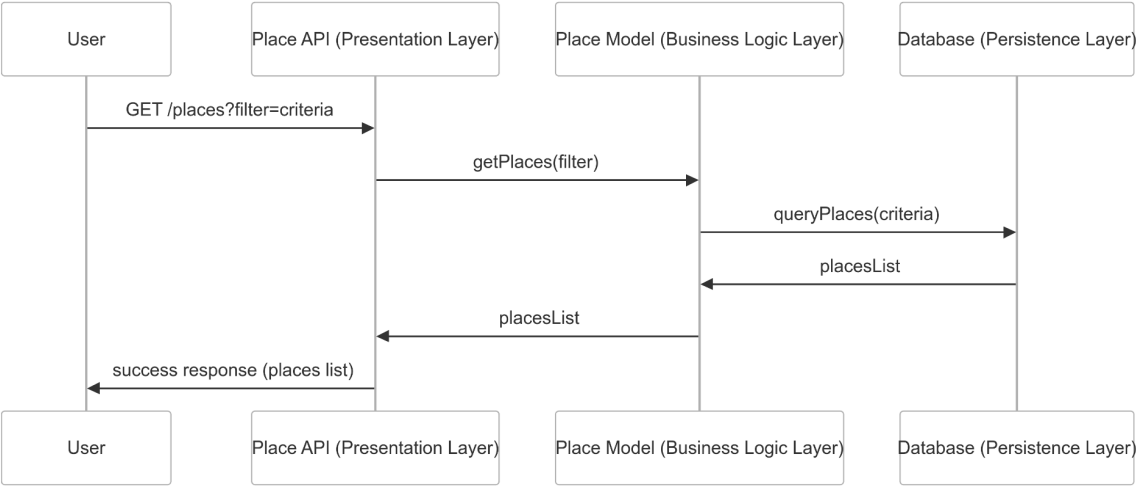
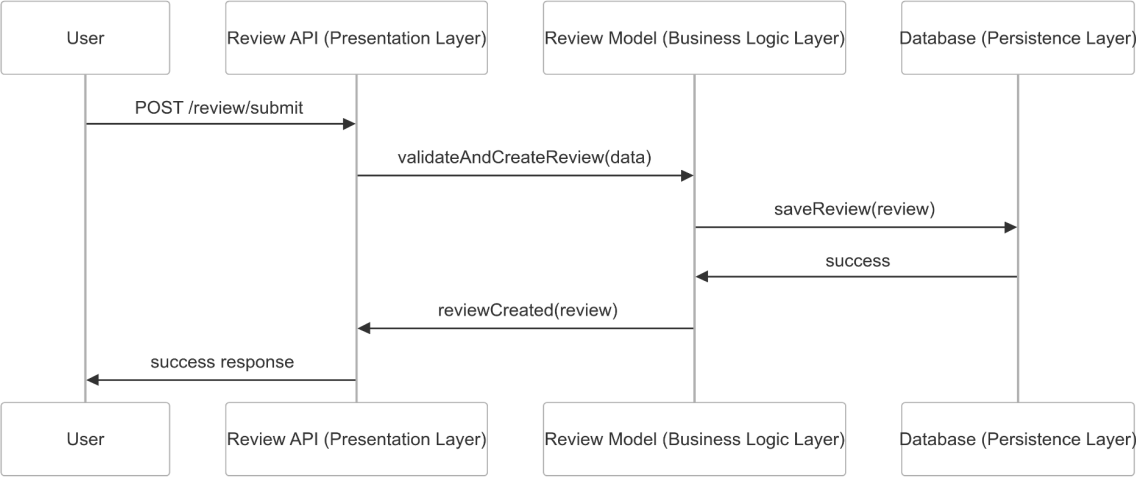


Explanation Notes:

- **Purpose:** This diagram represents the core business entities in the HBnB application, along with their attributes and relationships. It provides a detailed view of how the Business Logic layer is structured, focusing on key entities and their interactions.
- **Key Components:**
 - **User:** Represents a user of the system. Each user has basic profile attributes like name, email, and password, and can be identified as an administrator.
 - **Place:** Represents a property listed by a user. It contains information like the title, description, price, and location, and can have associated amenities.
 - **Review:** Represents a review left by a user for a place. It includes a rating and comment and is tied to both a user and a place.
 - **Amenity:** Represents amenities that can be associated with places, such as pool, wifi, etc.
- **Design Decisions:**
 - Each entity includes essential attributes, such as a unique identifier (UUID4) and timestamps for creation and updates, to ensure data integrity and auditing capabilities.
 - Relationships such as **one-to-many** (a user can have many places) and **many-to-many** (a place can have many amenities) are represented as associations between entities.
 - The **methods** of each class are designed to support CRUD (Create, Read, Update, Delete) operations for each entity, ensuring the system can manage user, place, review, and amenity data effectively.

Sequence Diagrams for API Calls





Explanation Notes for Each Sequence Diagram:

- **User Registration:**
 - **Purpose:** This sequence diagram shows the flow of interactions when a user registers a new account in the system. The flow starts with the user sending a registration request to the API, which then interacts with the Business Logic layer to validate and create the user. Finally, the system stores the user's information in the database.
 - **Key Components:** User, User API (Presentation Layer), User Model (Business Logic Layer), Database (Persistence Layer).
 - **Design Decisions:** The diagram highlights how the layers interact with each other using RESTful API calls and how data is passed between the layers for processing and storage.
- **Place Creation:**
 - **Purpose:** This diagram illustrates the flow for creating a new place listing. The user sends a request to create a place, and the API interacts with the Business Logic layer to validate and store the place in the database.
 - **Key Components:** User, Place API (Presentation Layer), Place Model (Business Logic Layer), Database (Persistence Layer).
 - **Design Decisions:** The diagram demonstrates how the system ensures that place data is validated before it is saved to the database, maintaining the integrity of the data.
- **Review Submission:**
 - **Purpose:** This sequence diagram outlines the process for submitting a review for a place. The user's review is validated and then saved to the database.
 - **Key Components:** User, Review API (Presentation Layer), Review Model (Business Logic Layer), Database (Persistence Layer).
 - **Design Decisions:** The review is linked to both the user and the place, emphasizing the system's relational data model. The sequence highlights the API's interaction with both the Business Logic and Persistence layers.
- **Fetching a List of Places:**
 - **Purpose:** This diagram shows the flow for fetching a list of places based on user-defined criteria. The API queries the Business Logic layer, which then retrieves the data from the database.
 - **Key Components:** User, Place API (Presentation Layer), Place Model (Business Logic Layer), Database (Persistence Layer).
 - **Design Decisions:** The diagram showcases the system's ability to efficiently query the database and return a list of places, highlighting the use of filtering criteria.