High-Level Diagram

Introduction

A high-level overview of the HBnB Evolution application's structure is depicted in Fig. 1. Serving as a foundational blueprint for development, this high-level diagram illustrates the overall architecture of the HBnB Evolution application, showcasing its distinct layers and their interconnected communication.

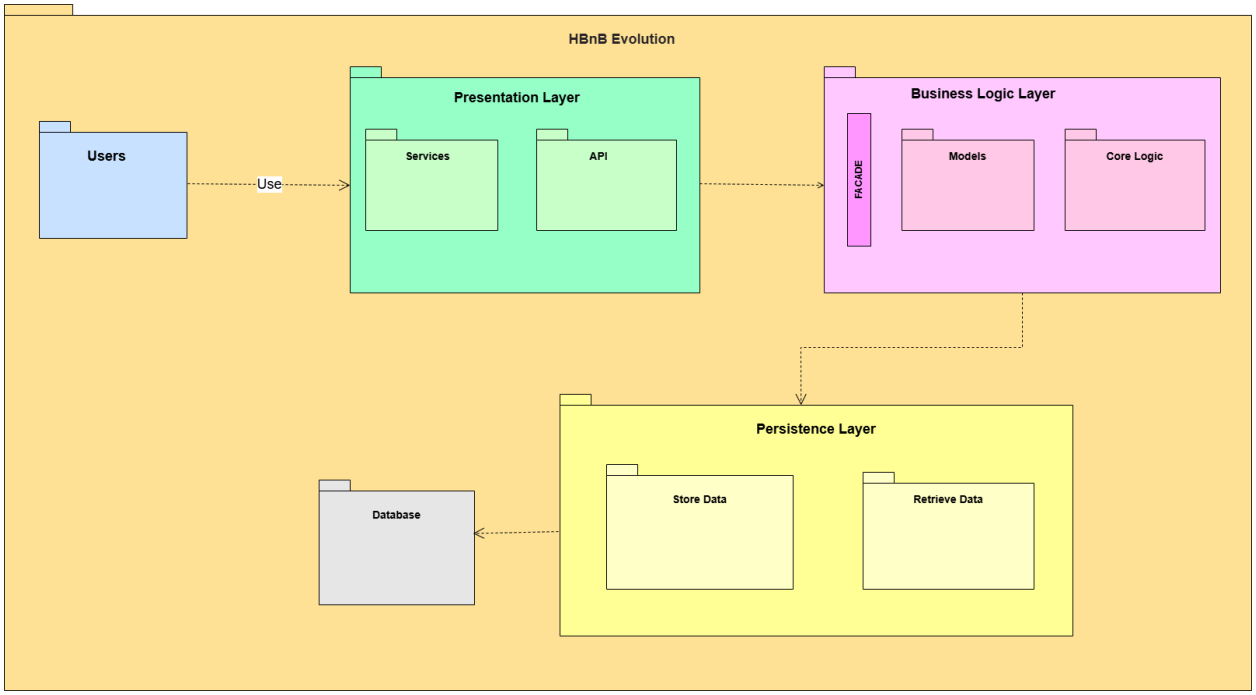

Fig. 1: HBnB Evolution high-level diagram.

Presentation Layer

The Presentation Layer is the part responsible for direct user interaction and the direct display of information.

- **Responsibilities**
  - **At API level**
    - Checks whether the user is authenticated and authorized to perform the requested action
  - **At services level**
    - Receives user requests (clicks, keystrokes)
    - Deserialization of incoming data (conversion from JSON format to usable object format)
    - Basic (simple) validation of data (valid email format, non-empty field)

        - Calls the Business Logic Layer
        - Receives the response of the Business Logic Layer
        - Serialization of response data (conversion from usable object format to JSON format)

Business Logic Layer (BLL)

The BLL is the layer that contains the specific rules that define how the system operates. It's independent of the way data is displayed (Presentation Layer) and the way it's stored (Persistence Layer). Through the BLL, there is a facade between the Presentation Layer and the BLL.

- **Responsibilities**
    - Contains and applies all the rules and constraints that define the application's behavior
    - Performs complex validations
    - Defines the structure and relationships of business objects and their behavior
    - Requests the Persistence Layer to store, retrieve, update or delete data required for BLL execution
    - Ensures that complex operations are performed in an atomic (all-or-nothing) manner

Persistence Layer

This Layer is exclusively responsible for data retention and retrieval. It's responsible for interaction with the data storage system, hiding the technical details of the database from the higher layers.

- **Responsibilities**
    - Performs the basic operations of creating, reading, updating and deleting data in the database
    - Converts business objects received from the BLL into database-comprehensible format
    - Converts data retrieved from the database into BLL-comprehensible object format
    - Independence of the database management system, allowing database changes without impacting the BLL

Cross-layer communication and Facade Pattern

In this layered architecture, specific patterns are employed to ensure clear and efficient communication between layers while managing complexity.

Communication flow:

From Presentation Layer to BLL: the Presentation Layer initiates requests to the BLL to perform business operations.

From BLL to Persistence Layer: the BLL orchestrates the flow of data and business rules, then requests the Persistence Layer to store or retrieve necessary data.

Response flow:

Responses travel back through the layers in the reverse order.

Role of the Facade Pattern:

The Facade pattern is applied in the BLL. Its purpose is to provide a simplified interface. It offers a single, unified, and simplified entry point for the Presentation Layer to access the complex underlying business logic and data operations.

The Facade Pattern hides away the internal workings of the BLL, such as how different business objects interact or how the data is retrieved or stored via the Persistence Layer.

The Facade Pattern facilitates the service execution. By presenting a clear and simplified set of operations (e.g. “register_user”, “list_places”), it makes it easy for the Presentation Layer to call the right service without having to know the complexity of their execution.