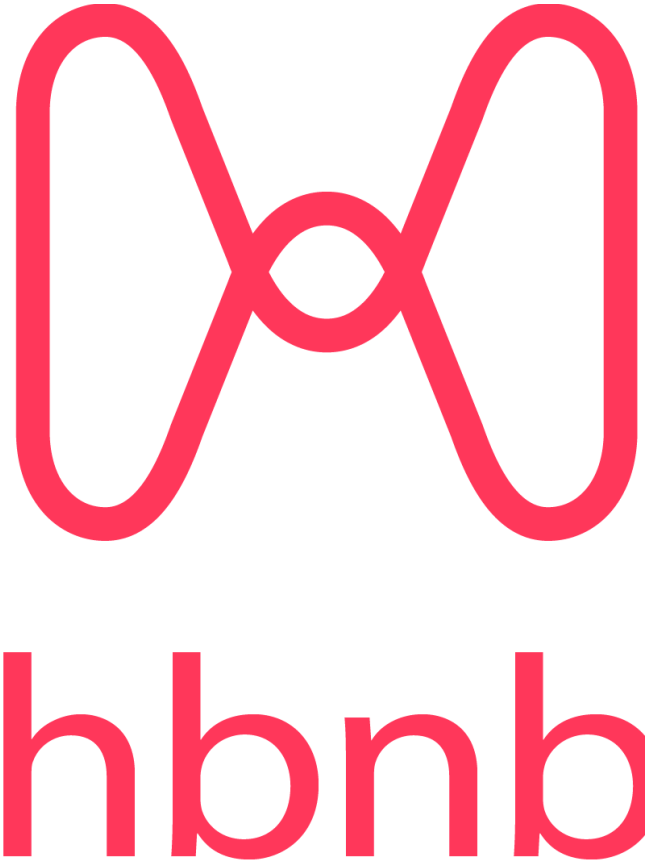


HBnB Evolution

Documentation Compilation

Gwendal Minguy-Pèlerin and Ingrid Mornac
June 2025

Summary

Introduction

The purpose of this initial part is to create a comprehensive technical documentation that will serve as the foundation for the development of the HBnB Evolution application.
This will help understanding the high-level architecture, the detailed design of the business logic and the interactions within the system.

The aim of this project is to create a clone of Airbnb, to realize a simplified version of it, named HBnB Evolution.

The Mermaid Chart Editor was used to create all the diagrams.

1. High-level Architecture

High-level Package Diagram

The purpose of this section is to illustrate the three-layered architecture of the HBnB application and the communication between these layers via the facade pattern.
This is an overview of how the different components of the application are organized and how they interact with each other.

It is a software design pattern that organizes the application into 3 main layers, where each layer has a specific responsibility and communicates only with its adjacent layers.

The package diagram represents the structure of the application with its 3 main layers :

- **Presentation Layer**
  - responsibilities: handles the interaction between the user and the application
  - includes : services and APIs exposed to the users
  - functions: display information

- **Business Logic Layer**
  - responsibilities: contains the core business logic and the models that represent the entities in the system
  - includes: user entity, place entity, review entity and amenity entity
  - functions: change entities

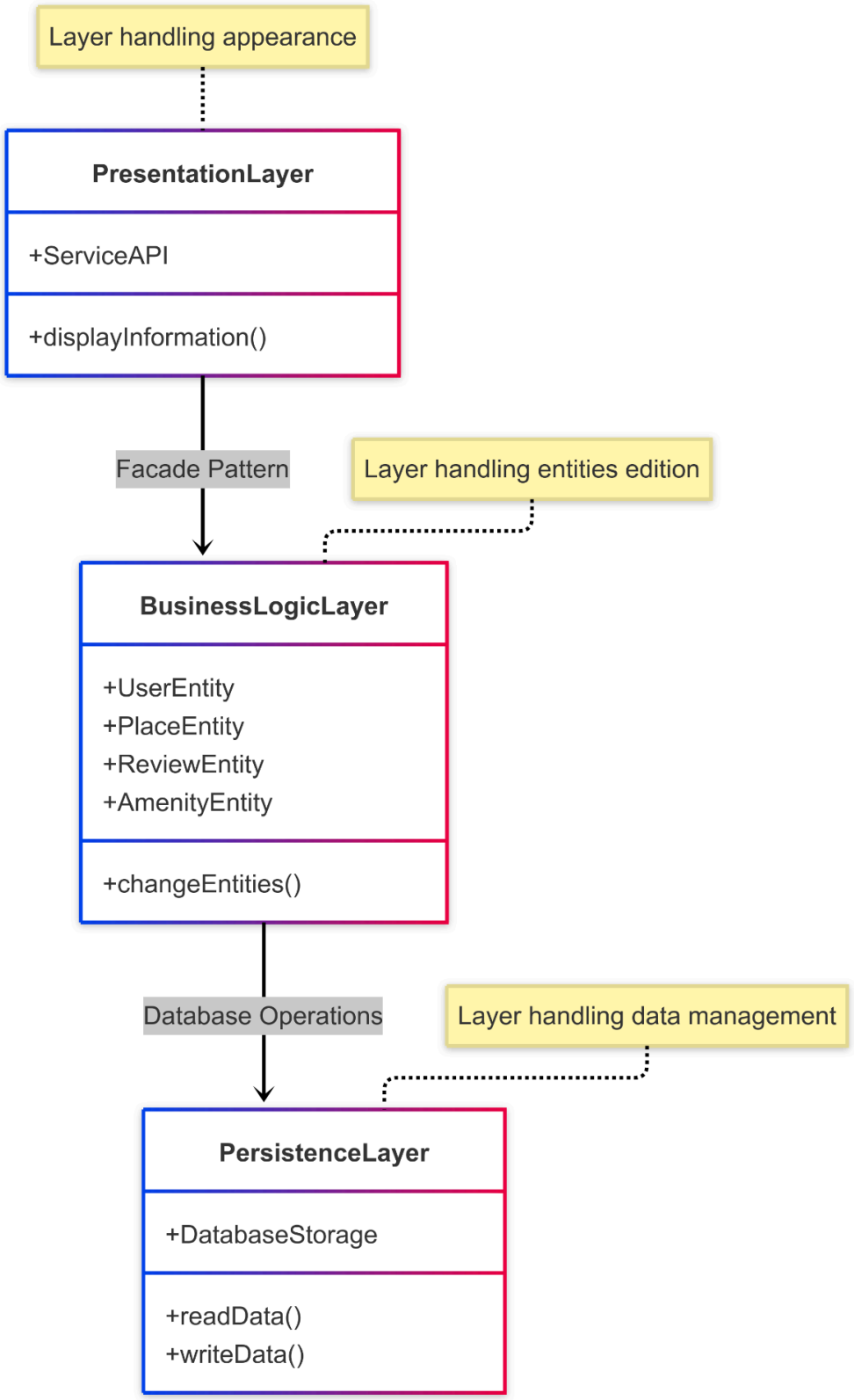
- **Persistence Layer**
  - responsibilities: handles database interaction for data storage and retrieval
  - includes: database storage
  - functions: read data, write data

It also has two interfaces:

- **Facade Pattern**, between Presentation Layer and Business Logic Layer:
  to simplify communication.
  to hide complexity by exposing a limited and clear interface.
  to tenfold layers by reducing direct dependencies.

- **Database Operations**, between Business Logic Layer and Persistence Layer:
  when the business logic layer needs to store, retrieve, or modify data, it asks the persistence layer to perform those operations.

Package Diagram



2. Business Logic Layer

Detailed Class Diagram for Business Logic Layer

The objective of this section is to create a detailed class diagram for the Business Logic Layer of the HBnB application. It includes the entities within this layer, their attributes and methods, and the relationship between them.

The detailed class diagram represents the structure of the Business Logic Layer:

- **class Model** (parent class)
  - attributes:
    - Id: identification of the object
    - CreatedAt: datetime of the creation
    - UpdatedAt: datetime of the updating
  - methods:
    - create() to create data
    - read() to read data
    - update() to update data
    - save() to save data
    - delete() to delete data
  - relationships: Inheritance with the 4 other classes as the Parent

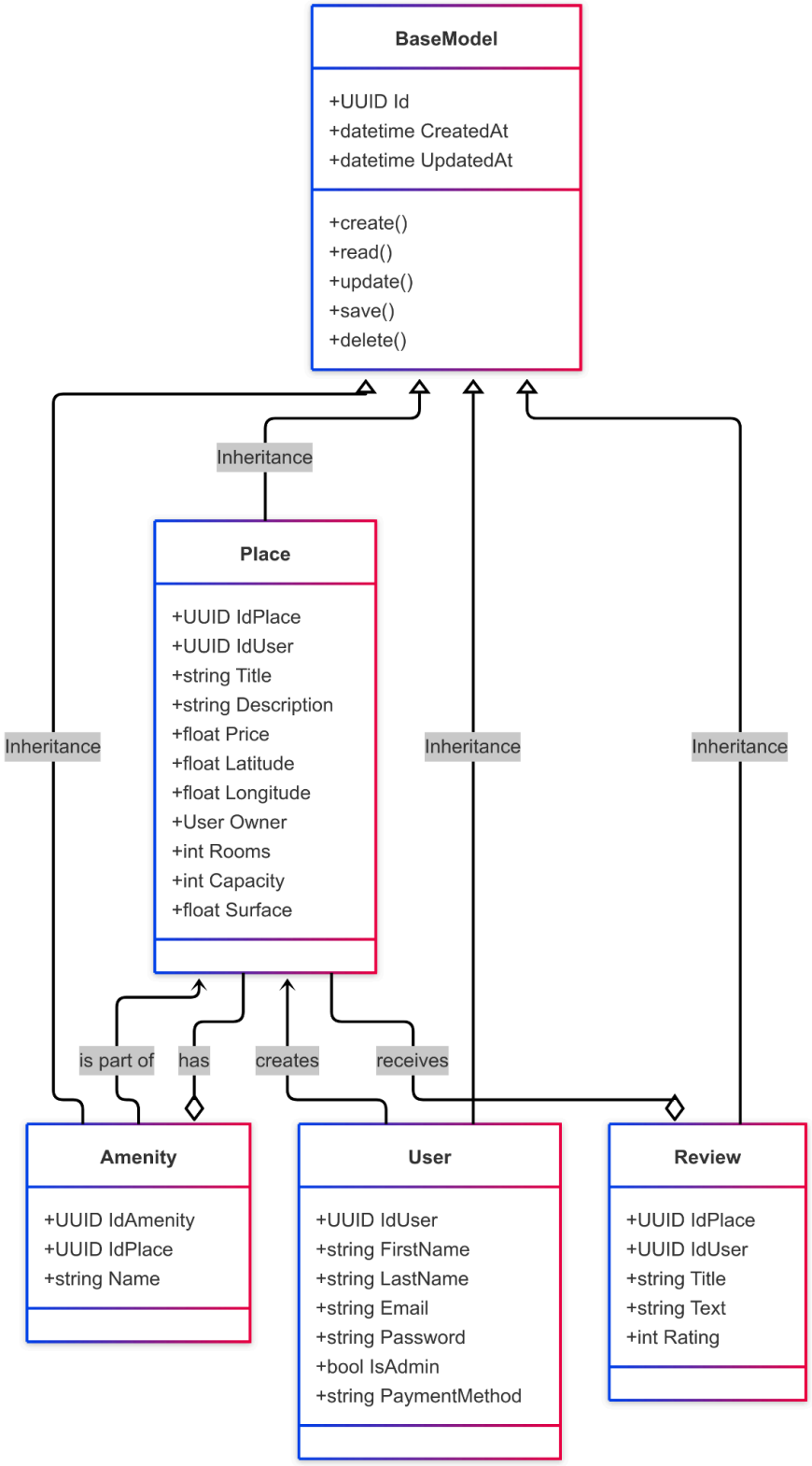
- **class User**
  - attributes:
    - IdUser: identification of the user
    - FirstName: identity of the user
    - LastName: identity of the user
    - Email: connexion information
    - Password: connexion information
    - IsAdmin: administrator status (boolean)
    - PaymentMethod: type of payment method
  - relationships: Association with Place

- **class Place**
  - attributes:
    - IdPlace: identification of the place
    - IdUser: key of class User to join it
    - Title: title of the place
    - Description: description of the place
    - Price: cost of the place
    - Location: latitude and longitude
    - Rooms: number of rooms in the place
    - Capacity: maximum number of people authorized
    - Surface: surface of the place
  - relationships: Association with Review

- **class Review**
  - attributes:
    - IdPlace: key of class Place to join it
    - IdUser: key of class User to join it
    - Title: title of the review
    - Text: review of the user
    - Rating: rating chosen by the user
  - relationships: Dependance with Place

- **class Amenity**
  - attributes:
    - IdPlace: key of class Place to join it
    - IdAmenity: identification of the amenity
    - Name: name of the amenity

Class Diagram



3. API Interaction flow

Sequence Diagrams for API Calls

The objective of this section is to develop sequence diagrams for 4 different API Calls, to represent the interaction between each of the layers. It allows us to understand the sequence of operations needed, from the request to the response.

These sequence diagrams allows us to describe and visualize the way components of the system interact for four specific use cases:

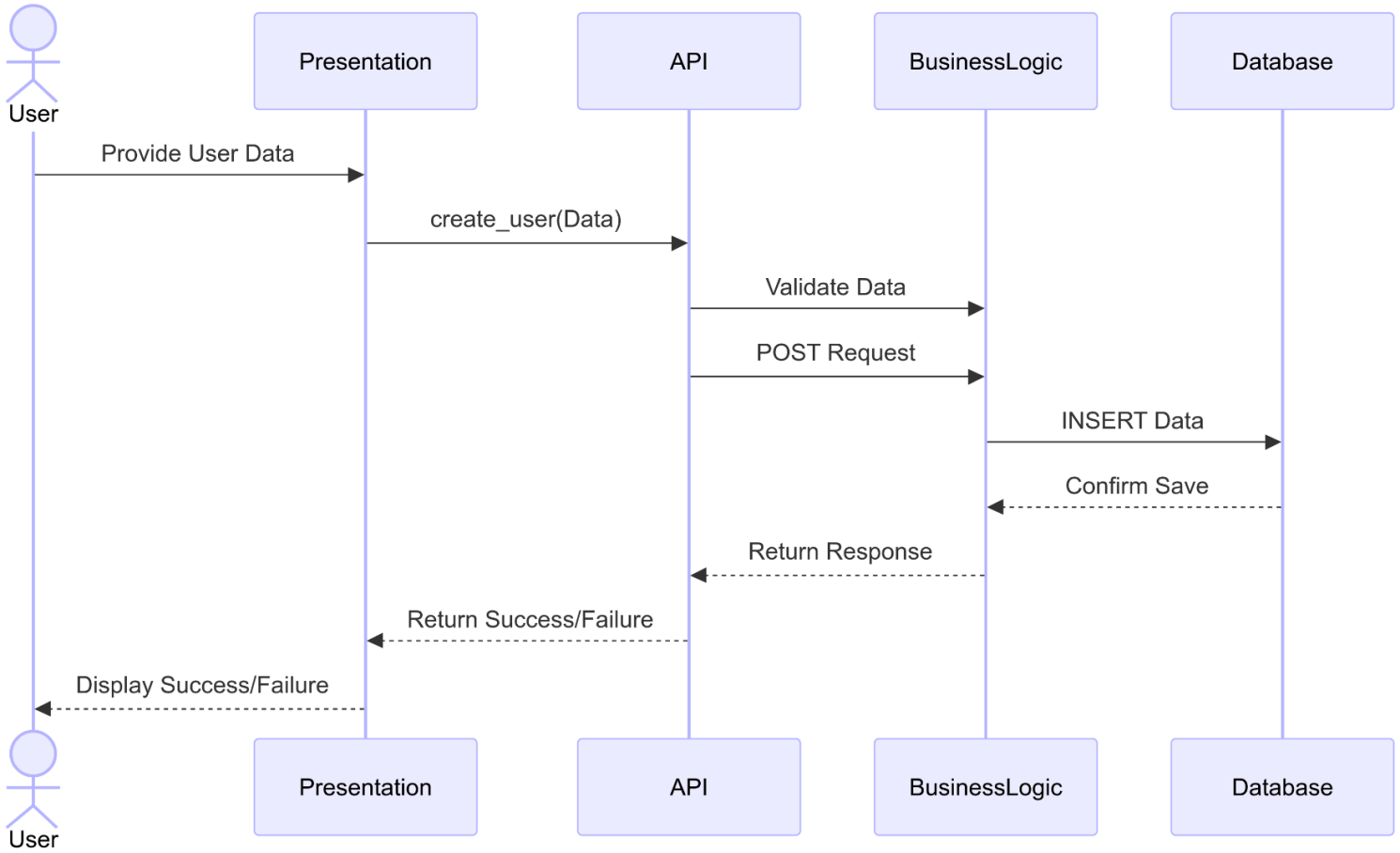
- **User Registration** (user signs up for a new account)
  - fields:
    - FirstName
    - LastName
    - Email & Password
    - PaymentMethod
  - return: success or failure

- **Place Creation** (user creates a new place listing)
  - fields:
    - Title
    - Address
    - Location
    - Rooms
    - Capacity
    - Price
    - Description
  - return: success or failure

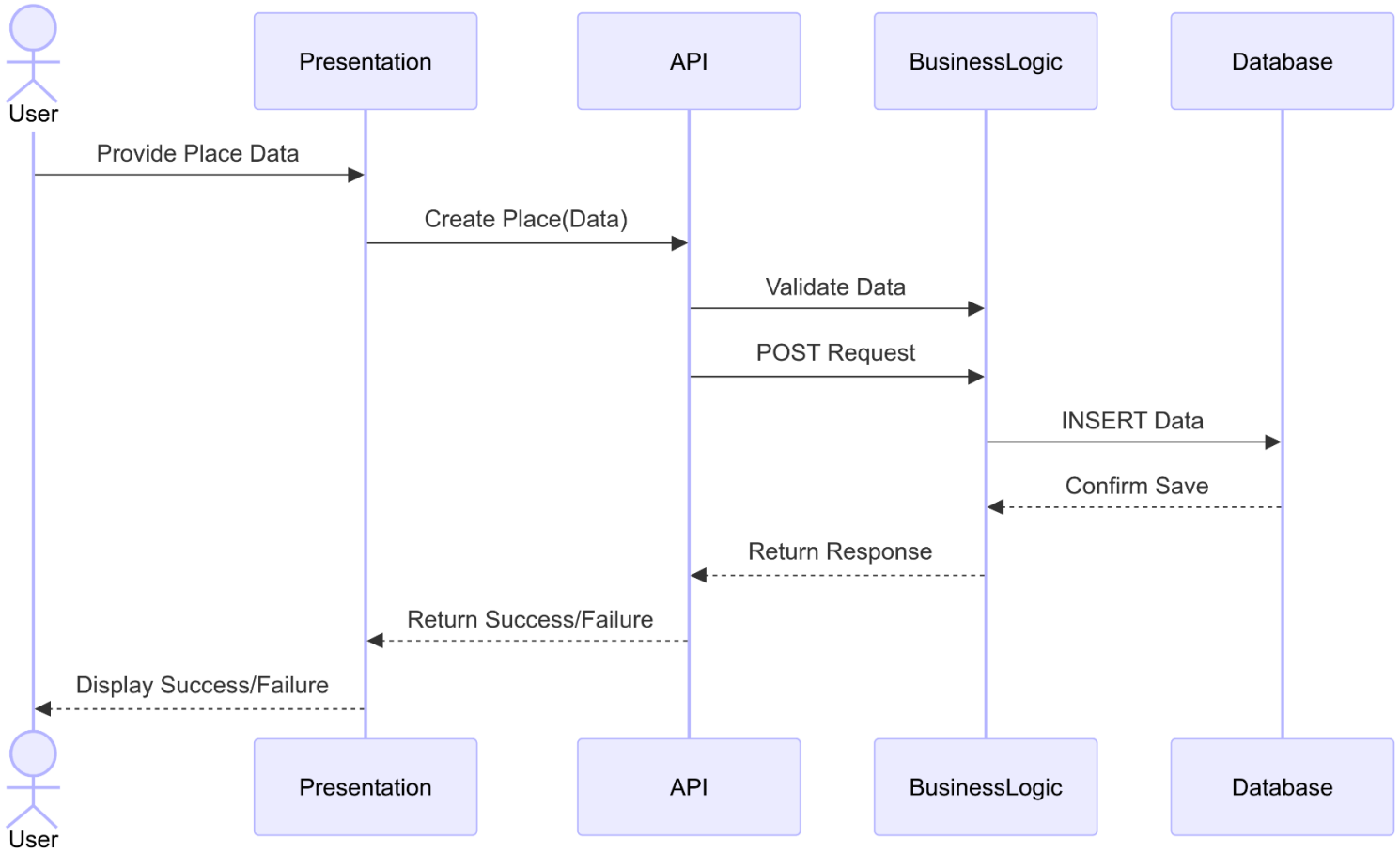
- **Review Submission** (user submits a review for a place)
    - Title
    - Text
    - Rating
  - return: success or failure

- **Places List Fetching** (user requests a list of places based on certain criteria)
  - fields:
    - Place
    - Capacity
    - Price
  - return: list of results

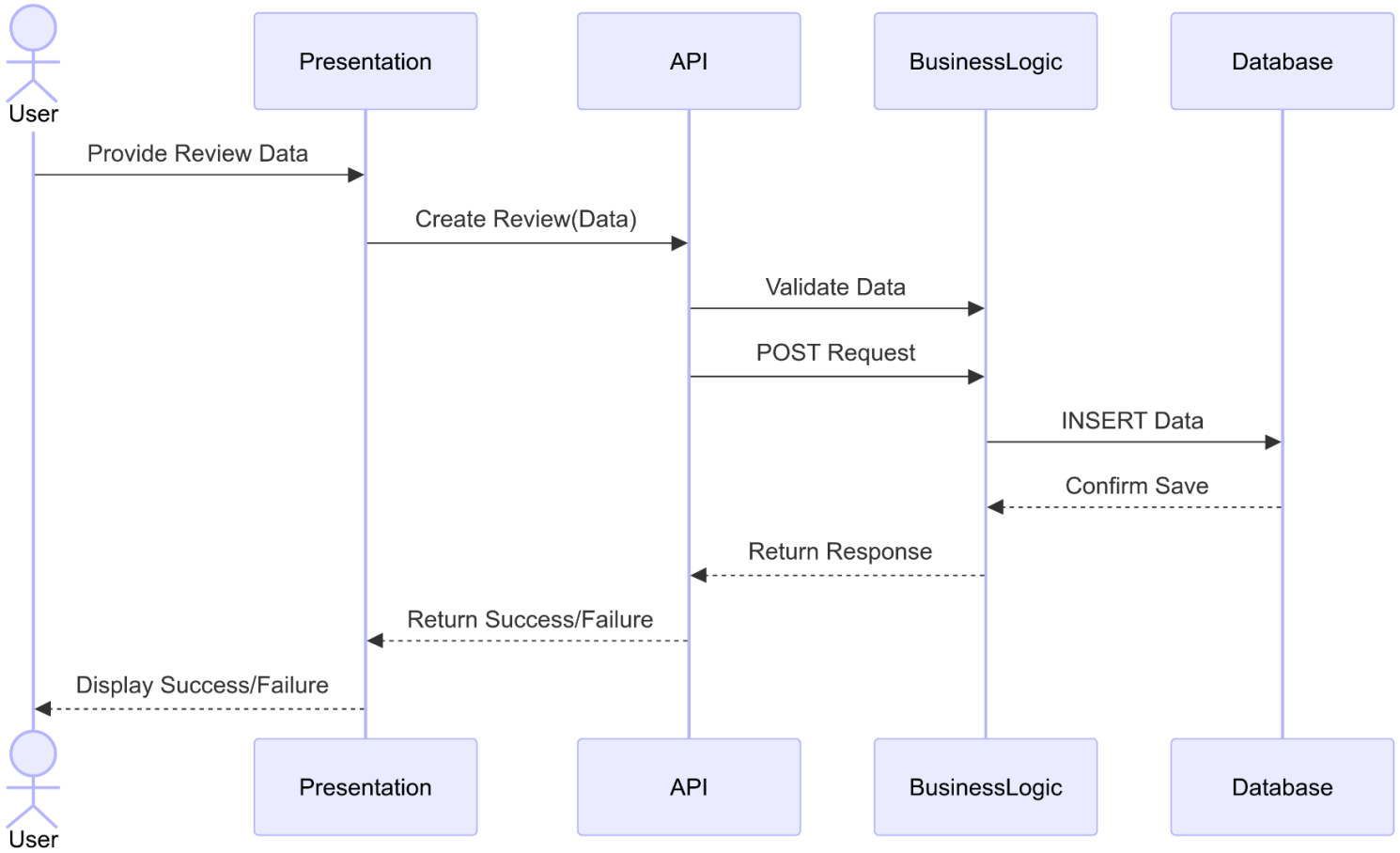
Sequence Diagram: User Registration



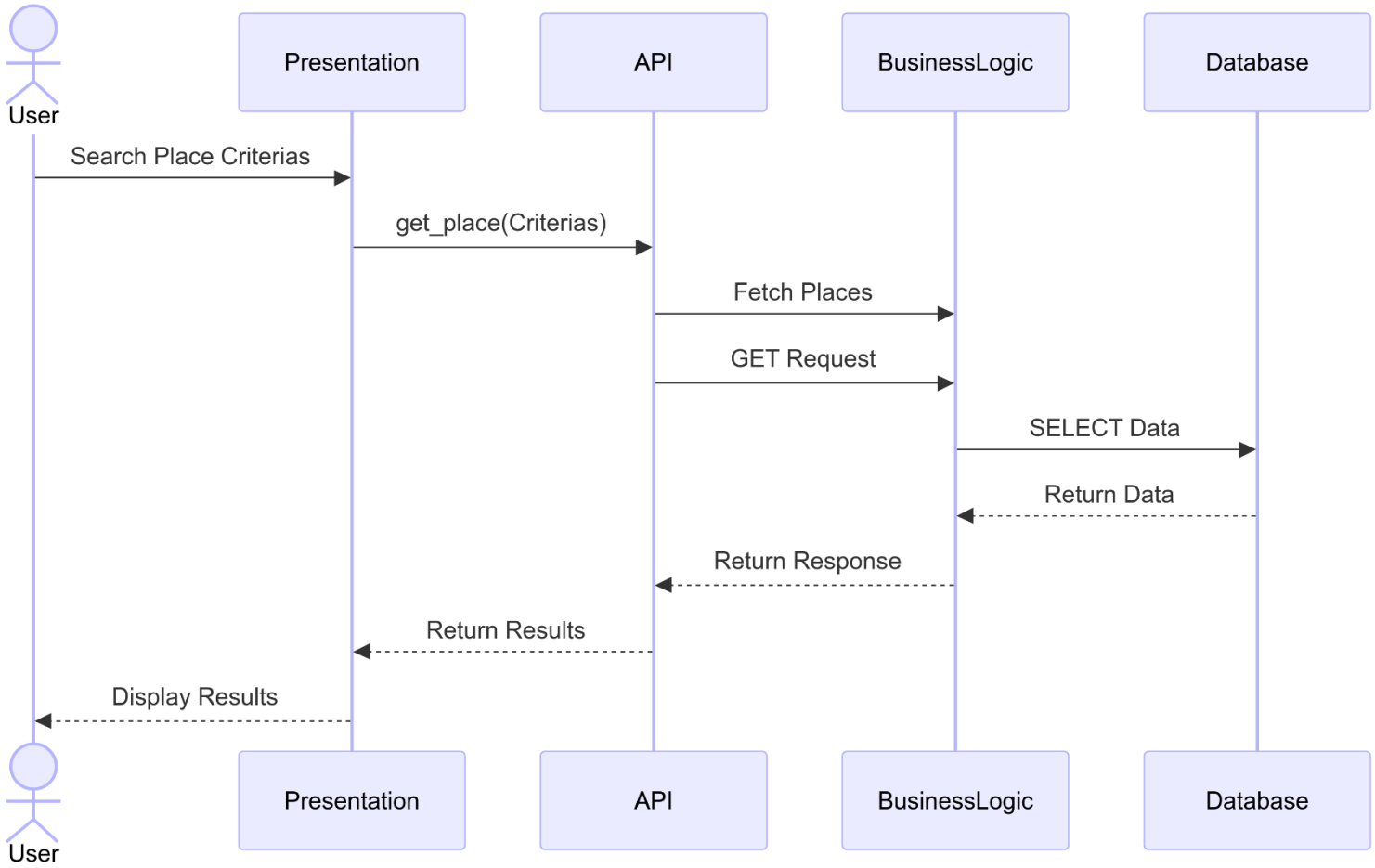
Sequence Diagram: Place Creation



Sequence Diagram: Review Submission



Sequence Diagram: Places List Fetching



Conclusion

At the end of this initial part, we have created a complete set of technical documentation.
This documentation ensures that we have a solid understanding of the application design and architecture. This whole document will be updated as needed during the development of the HBnB Evolution, in order to match any future change made in its architecture.