

Comprehensive Technical Architecture Document

Introduction

Project Overview

This document presents the technical architecture for a web-based platform that enables users to search for places based on specific criteria, create accounts, submit reviews, and manage listings. The system is designed with a multi-layered architecture that separates concerns and ensures scalability, maintainability, and reliability.

Purpose of the Document

This comprehensive technical document serves as a reference guide for the implementation phases of the project. It provides detailed insights into the system's architecture, business logic components, and API interaction patterns. The document is our guideline during the lifecycle of our application HBnB.

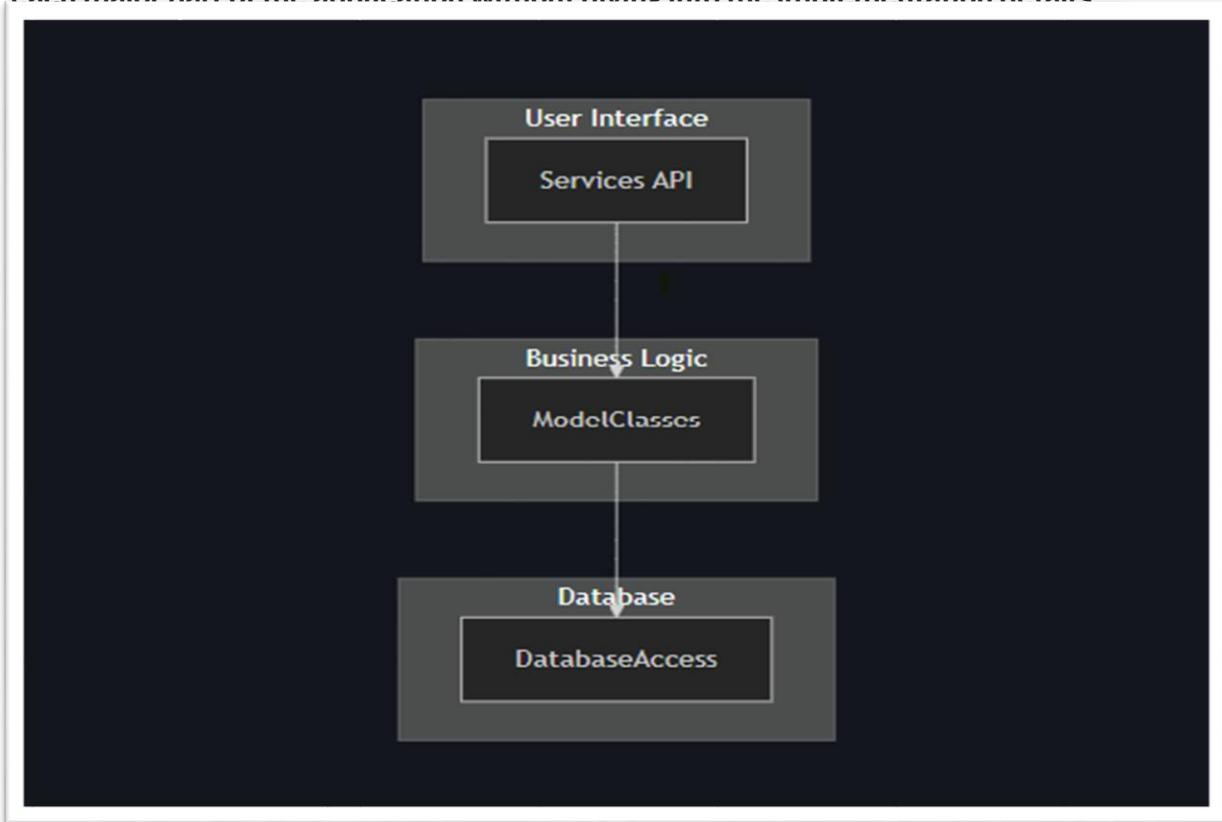
Table of Contents

1. [Introduction](#)
2. [High-Level Architecture](#)
3. [Business Logic Layer](#)
4. [API Interaction Flow](#)
5. [Conclusion](#)

High-Level Architecture

The system follows a three-tier architecture pattern, ensuring clear separation of concerns and optimal resource utilization.

In this diagram, the purpose is to show how the global architecture of the application work. It illustrates how the application is modularly divided into three keys logical layers. This diagram helps developers quickly understand the structure and responsibility of each major part of the application without diving into the implementation details.



◆ Key components

1 UserInterface

Contains Services and API components.

Handles external user interactions, such as HTTP requests.

2 BusinessLogic

Contains core ModelClasses such as User, Place, Review, and Amenity.

Implements the application's rules and workflows.

3 DataBase

Contains DataBaseAccess components, typically involving repositories or data mappers.

Responsible for storing and retrieving persistent data.

C Communication Arrows

UserInterface --> BusinessLogic is done via a Facade to decouple the frontend logic from core rules.

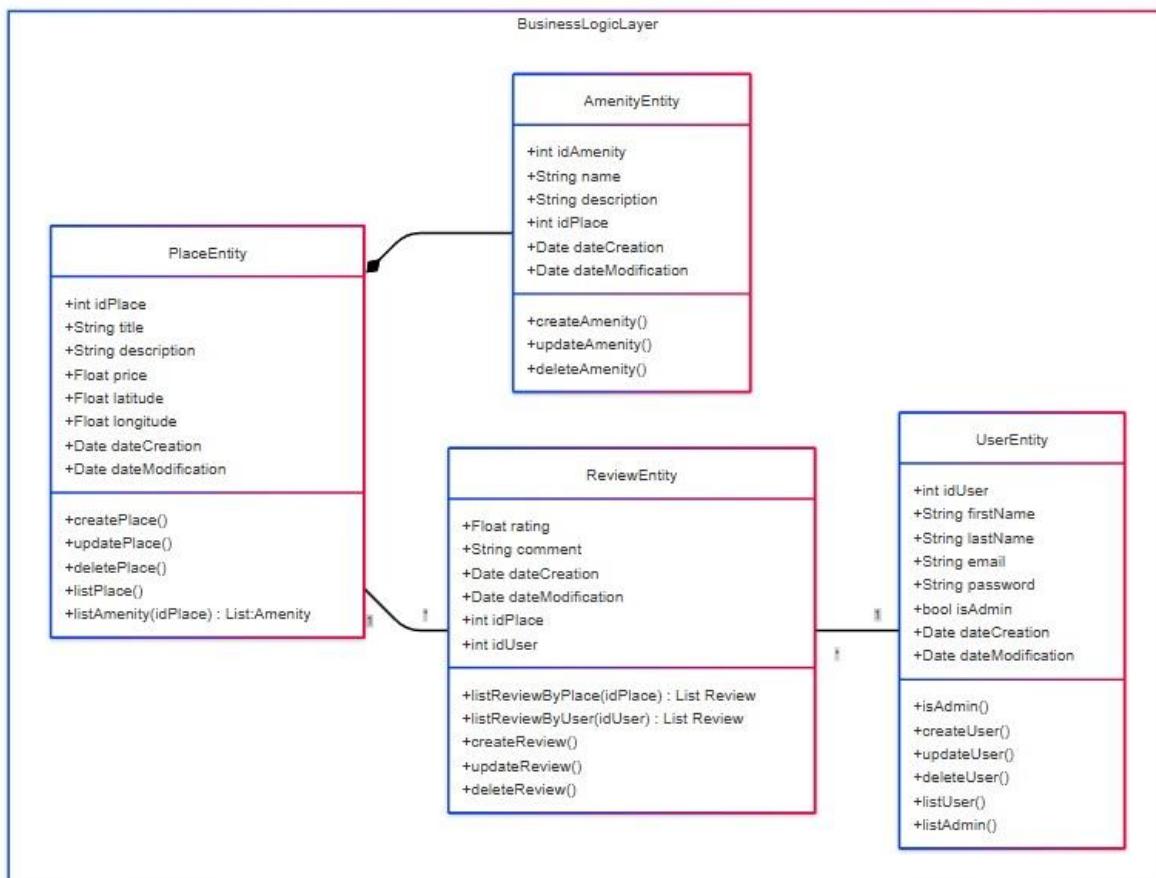
BusinessLogic --> DataBase represents Database Operations.

Business Logic Layer

Class Diagram

This diagram describe the interaction with the different Logic component. First the place have some amenity, a amenity cannot exist without place. A place is located by a user, this user can make a review about the quality of service. AMENITY, USER, PLACE AND REVIEW are the entity use by the website via the BusinessLogicLayer Relation between place and review.

- One place can have many review
- One review is written by one user and one place
- One user can create many review



Purpose: Handles all user-related operations including registration, authentication, and profile management

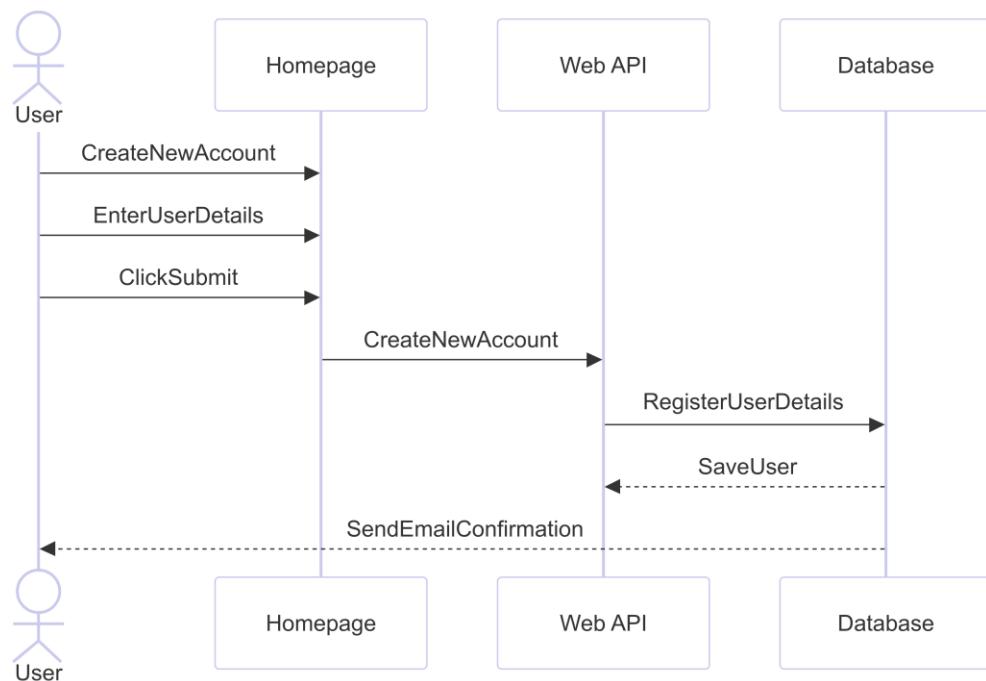
API Interaction Flow

Sequence Diagram

In this Sequence we created sequence diagrams that represent the flow of interactions across the different layers of the application for specific API calls. These diagrams show how the presentation layer, business logic and persistence layer communicate with each other to handle user requests.

User Registration

User account creation process. The user fills in their information on the homepage, which is transmitted via the web API to be stored in the database with confirmation email sending.



Key Interactions:

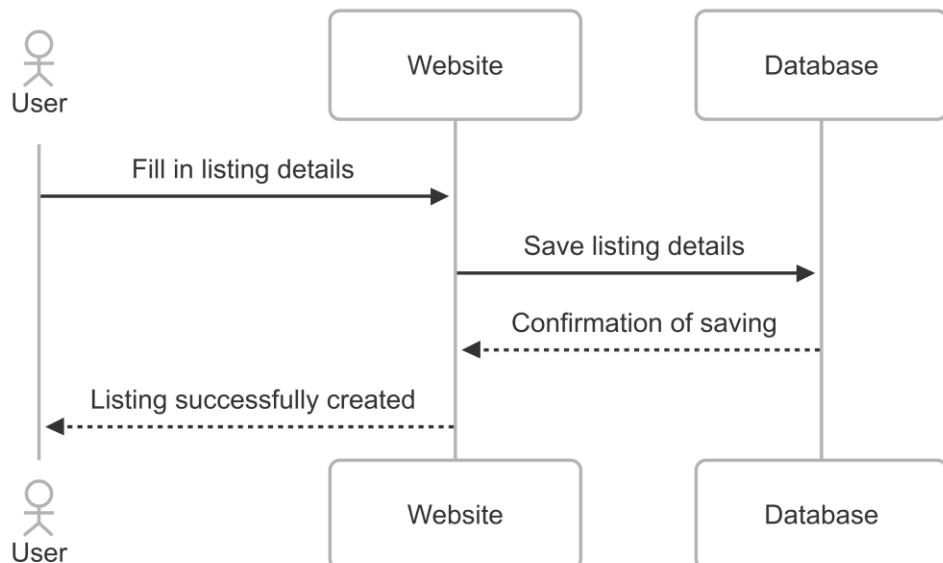
- Form validation occurs both client-side and server-side
- Password encryption is handled by the API layer

- Email confirmation ensures user authenticity
- Database transactions ensure data consistency

Purpose: This sequence demonstrates the place search functionality where users can discover locations based on their specific requirements. The process begins when a user connects to the website and enters their search criteria. The website forwards these criteria to the business logic layer, which processes the request and interacts with the database to retrieve matching places. The results are then formatted and displayed to the user.

Place creation

Simplified listing creation process. The user enters their listing details on the website, which saves them directly to the database and confirms successful creation.



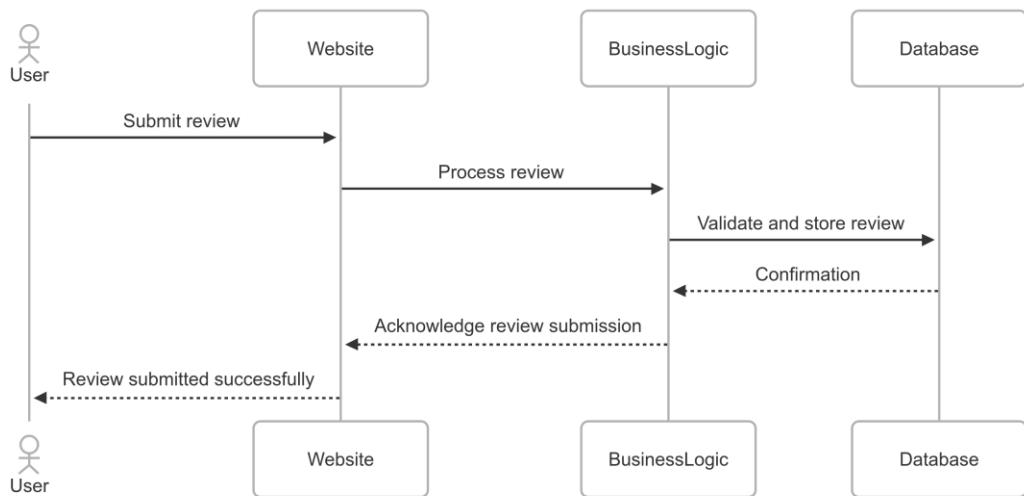
Key Interactions:

- User input validation occurs at the website level
- Business logic handles search algorithm implementation
- Database queries are optimized for performance
- Results are cached for subsequent similar requests

Purpose: This sequence shows the listing creation process, where users can add new places to the platform.

Review Submission

User review submission flow. The system processes and validates the review through business logic before storing it in the database and confirming submission to the user.



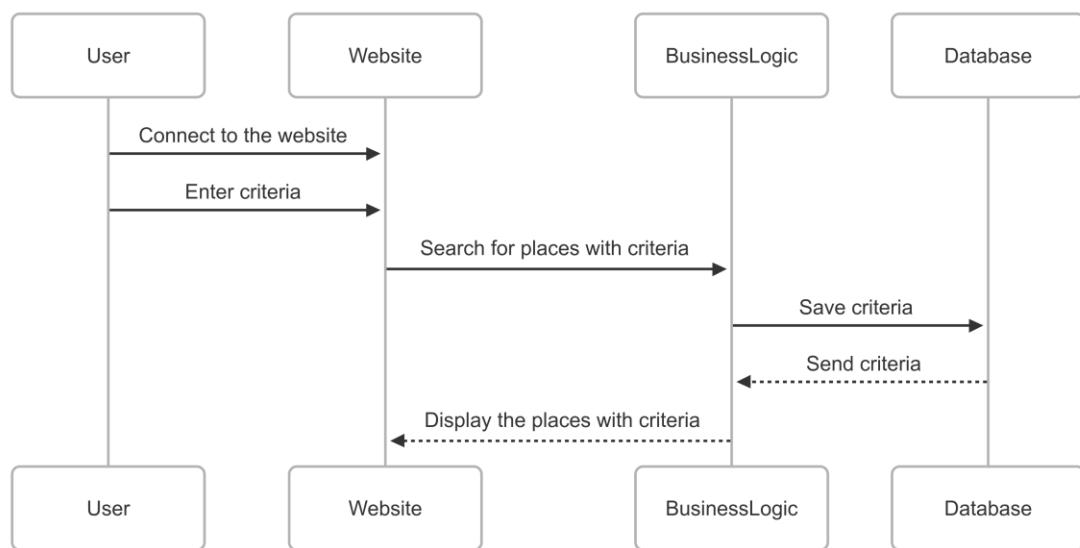
Key Interactions:

- Content moderation prevents inappropriate submissions
- Review validation ensures quality standards
- User authentication is verified before submission
- Confirmation feedback improves user experience

Purpose: This sequence demonstrates the review submission process, which includes content validation, moderation, and storage. Users submit reviews through the website interface, which are then processed by the business logic layer for content validation and moderation before being stored in the database.

Fetching a List of Places

Place search flow based on user-specified criteria. The user enters their criteria via the website, which interacts with business logic to retrieve and display results from the database.



Key Interactions:

- **Form validation ensures complete listing information**
- **Image upload and processing (if applicable)**
- **Automatic categorization based on listing attributes**
- **Immediate confirmation provides user feedback**

Purpose: This simplified sequence shows the listing creation process, where users can add new places to the platform.

Conclusion

System Benefits

This architecture provides several key advantages:

- **Scalability:** Modular design allows independent scaling of components
- **Maintainability:** Clear separation of concerns facilitates easier updates
- **Security:** Multiple validation layers ensure data protection
- **Performance:** Optimized data flow and caching strategies
- **Reliability:** Robust error handling and transaction management

This technical document serves as the foundation for successful implementation of our HBnB project, providing clear guidelines and ensuring proper architectural decisions during the lifecycle of the project.

Authors

- [@Sebastien Salgues](#)
- [@Elhadj Reziga](#)