Hotel Room Reservation System HubTech Solutions

Adan Delgado Ghaith Ishaq Agustin Sandoval Rymma Won



Project Overview

a complete online hotel reservation system that makes it simple for users to explore, book, and manage reservations. Features that are geared for mobile and the web include room administration, booking management, transaction reports, and more.

Key Architectural Drivers

The architecture is driven by requirements related to scalability, security, usability on mobile and web platforms, and integration capabilities for administrative and payment processing.











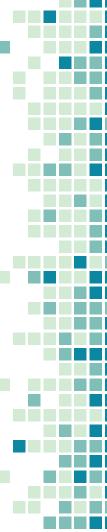


Architectural Style Choices

Evaluated styles:

- Microservices for flexibility and scalability.
- Monolithic for simplicity and development speed.

Chosen style: Microservices because of the ease of integrating different services and the requirement for scalability.



Architecture Diagram Overview:





ARCHITECTURE DIAGRAM

User Interface (UI) Layer:

Web Application: For accessing the system through browsers on desktops and mobile devices.

Mobile App: Native or hybrid applications for iOS and Android platforms for on-the-go access.

API Gateway:

Serves as the entry point for all client requests, directing them to the appropriate microservice.

Microservices:

User Management Service: Handles user registration, login, and profile management.

Room Management Service: Manages room details, statuses, and availability.

Reservation Service: Processes reservations, including bookings, modifications, and cancellations.

Payment Service: Manages payment processing and integrates with external payment gateways.

Email Service: Sends out transaction confirmations and other notifications via email.

Database/ Admin Layer:

Separate databases for each microservice, ensuring service independence and data isolation.

A web-based dashboard for hotel staff to manage reservations, view reports, and update room details.

External Integrations:

Payment Gateway: For processing credit card transactions securely.

Email Provider: For sending emails to users.



Conclusion

A Microservices architecture is used in our system to guarantee scalability and flexibility.

Key Issues include managing service communication, integrating payment services, and guaranteeing data security.