Hotel Room Reservation System HubTech Solutions

Adan Delgado Ghaith Ishaq Agustin Sandoval Rymma Won



Project Overview

A comprehensive system for online hotel room reservation, allowing users to browse, book, and manage reservations with ease. Features include booking management, transaction reports, room management, and more, all optimized for mobile and web.

Key Architectural Drivers

Requirements driving the architecture include scalability, security, usability on mobile and web platforms, and integration capabilities for payment processing and administrative functions.











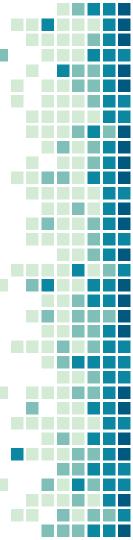


Architectural Style Choices

Evaluated styles:

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

Chosen style: Microservices, due to the need for scalability and the ease of integrating various services.



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

Our system utilizes a Microservices architecture to ensure scalability and flexibility.

Key issues include ensuring data security, integrating payment services, and managing service communication.