

HOTEL MANAGEMENT SYSTEM

Problem statement: The Hotel Management System is a web-based application designed to streamline hotel operations for managers through an interactive GUI, enabling efficient handling of room bookings, staff management, and various hotel activities. Tailored for busy managers, this system centralizes management tasks, eliminating the need for manual, paper-based processes. Managers can post available rooms, and customers can easily view and book them online. An admin feature allows for booking approvals, ensuring controlled and organized reservations. Additionally, customers can access and book other hotel services, making the system convenient for both managers and customers by providing an all-in-one platform to efficiently manage hotel activities.

Features:

- **Admin login & admin dashboard:** It has admin login who has the authority of the system & he is responsible for approving & disapproving the users request for room booking. Admin can add & delete notifications & updates in the system.
- **User registration:** There is user registration form available where new users can create their account by providing required information to the system.
- **Booking system:** User can request for the table booking for a particular date & time.
- **Approving/Disapproving request:** The booking requests are directly sent to admin account by the system. Admin can view all the requests along with respective user details & therefore make decisions for cancelling the requests.

Software Requirements Specification:

- **Functional requirements:**

These describe what the system should do, i.e., the core functionalities needed to manage the operations of a hotel.

1. User Management:

- The system should allow administrators to create, update, and delete user profiles for different roles (e.g., admin, staff, guest).
- The system should allow guests to create their own accounts and manage personal information.

2. Room Management:

- The system should allow the hotel to manage room availability, types, and prices.
- The system should display available rooms based on guest search criteria (e.g., room type, number of guests, check-in/check-out dates).
- The system should allow staff to update room status (e.g., clean, dirty, maintenance).

3. Reservation Management:

- Guests should be able to make room reservations online or at the hotel front desk.
- The system should validate room availability during booking and prevent overbooking.
- Guests should receive confirmation of their reservation via email or SMS.
- The system should allow cancellations and modifications to bookings within specified policies.

4. Check-in and Check-out:

- The system should support check-in and check-out processes, including issuing room keys (physical or digital).
- The system should track check-in and check-out times.
- The system should automatically calculate the total bill based on stay duration, room type, and any additional services.

5. Billing and Payments:

- The system should support generating invoices based on room charges, services used (e.g., minibar, room service), and taxes.

- The system should allow guests to pay using various methods (e.g., credit/debit cards, cash, online payments).
 - The system should allow guests to view a detailed breakdown of charges at check-out.
6. Inventory and Housekeeping Management:
- The system should track inventory levels for consumables (e.g., linens, toiletries).
 - The system should allow housekeeping staff to track room cleaning schedules and requirements.
 - The system should notify staff if supplies are running low.
7. Customer Feedback and Complaints:
- The system should allow guests to provide feedback or lodge complaints about their stay.
 - The system should track customer feedback and allow staff to respond or resolve complaints.
8. Reporting:
- The system should provide reports on occupancy rates, revenue, guest demographics, and room usage.
 - It should allow administrators to generate financial reports, including income, expenses, and outstanding payments.
9. Multi-language Support:
- The system should support multiple languages for international guests.

● **Non-functional requirements:**

These describe the qualities the system must exhibit, ensuring it performs well under various conditions and is maintainable.

1. Performance:

- The system should handle up to [X] simultaneous users without significant degradation in performance.
- The system should process room reservations and check-ins within [X] seconds to ensure quick service.
- The system should be able to generate reports within [X] seconds/minutes.

2. Scalability:

- The system should be able to scale horizontally to accommodate increasing user load as the hotel chain grows (e.g., adding more locations or users).

3. Availability:

- The system should be available 24/7 with an uptime of [99.9%] or better.
- Maintenance windows should be scheduled and communicated in advance.

4. Security:

- The system should comply with data protection regulations (e.g., GDPR, CCPA) to safeguard guest information.
- The system should implement encryption for sensitive data (e.g., payment information, personal guest details).
- Access control should be implemented to ensure that only authorized personnel can access specific data or functions (e.g., admin access, financial reports).
- The system should support two-factor authentication (2FA) for sensitive user accounts.

5. Usability:

- The system should have an intuitive and user-friendly interface, making it easy for staff and guests to use.
- The system should support mobile devices and provide a responsive design for both hotel staff and guests.

6. Compatibility:

- The system should be compatible with different browsers (e.g., Chrome, Firefox, Safari) and mobile operating systems (iOS, Android).
- It should integrate with third-party tools (e.g., payment gateways, CRM systems, channel managers).

7. Maintainability:

- The system should be easy to update and maintain, with clear documentation for both users and developers.
- It should support modular development to add new features with minimal disruption.

8. Backup and Recovery:

- The system should have an automated backup mechanism to ensure data is regularly backed up.
- In the event of a failure, the system should have a disaster recovery plan to restore data within [X] hours.

9. Compliance:

- The system should meet all local and international regulations for handling guest data, payments, and business operations.

Software requirements:

- Windows XP, Windows 7(ultimate, enterprise)
- Visual Studio 2010
- SQL 08

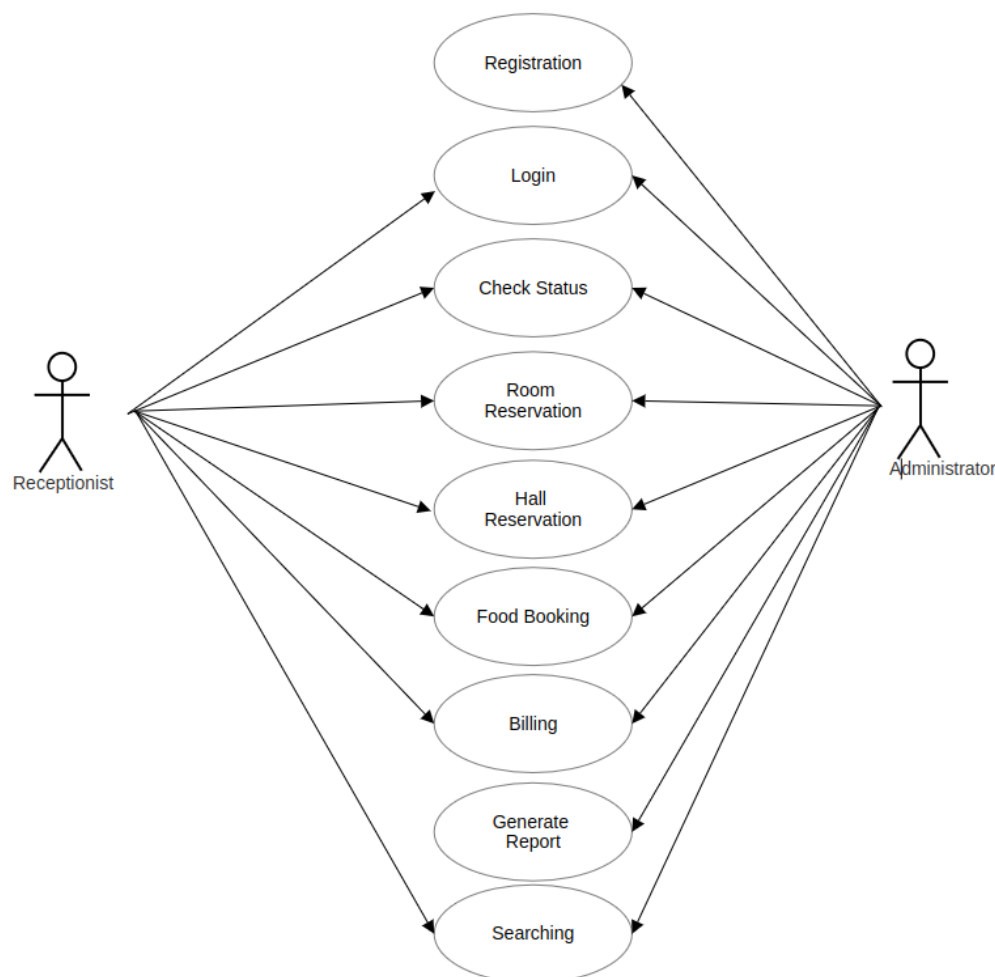
Hardware components:

- Process - p4
- Hard disk - 5GB
- Memory - 1GB RAM

UML Diagrams:

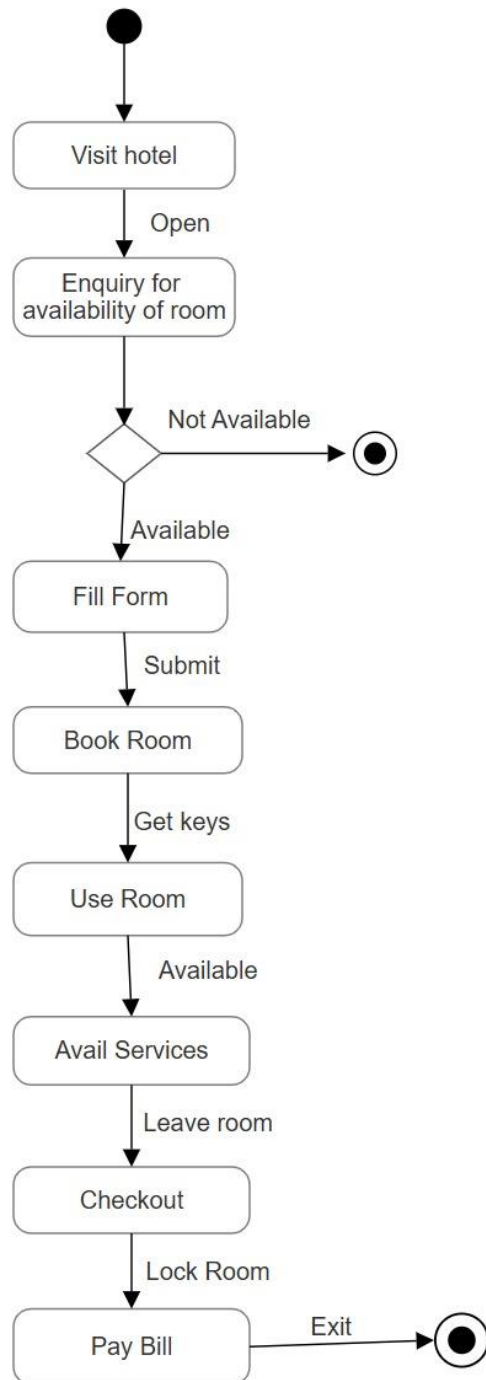
Use-case diagram:

A use case diagram is a tool to visualize system requirements by showing how actors interact with the system. In the Hotel Management System, there are two main actors: the Receptionist and the Administrator. The Receptionist handles tasks like guest registration, booking rooms and halls, managing food orders, processing billing, and generating reports. The Administrator has control over all functions, including monitoring and adjusting operations. The diagram uses stick figures to represent actors, connecting them to system functions to clarify user interactions and system requirements.



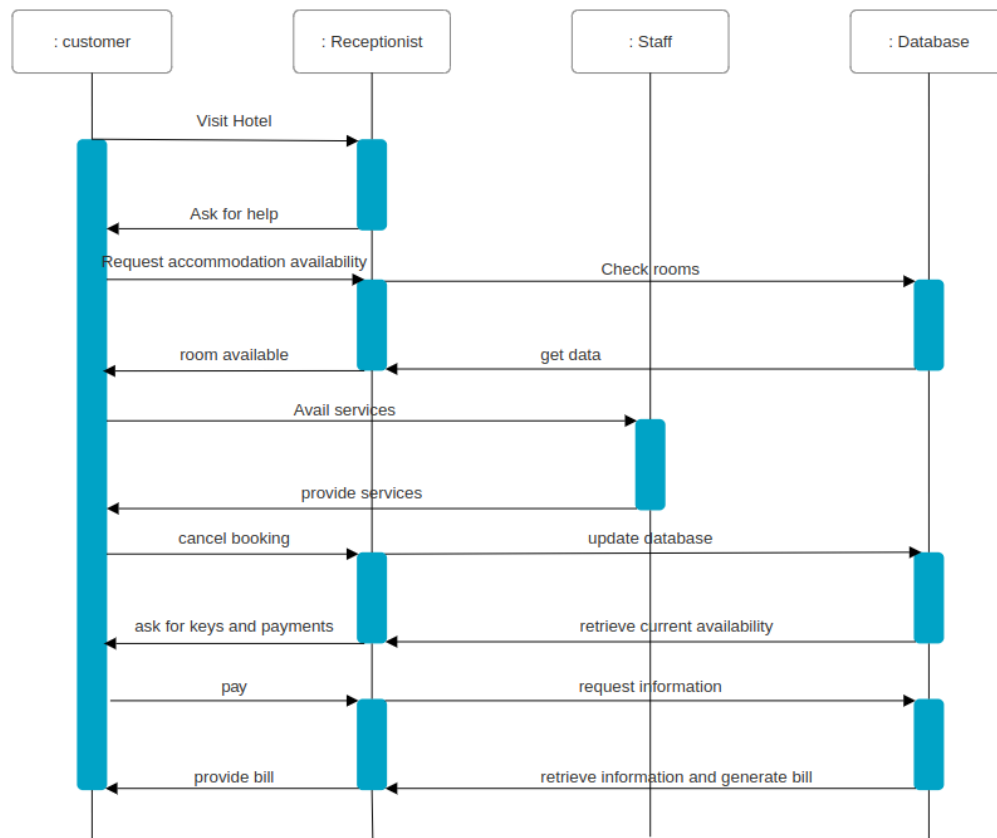
Activity diagram:

An activity diagram shows the flow of actions in a system. In the Hotel Management System, the Receptionist registers guests, checks availability, books room, manages orders, and processes billing. Decisions like availability or payment affect the next steps. The Administrator oversees and manages these tasks, ensuring smooth operations. The diagram uses actions and decision points to represent the workflow.



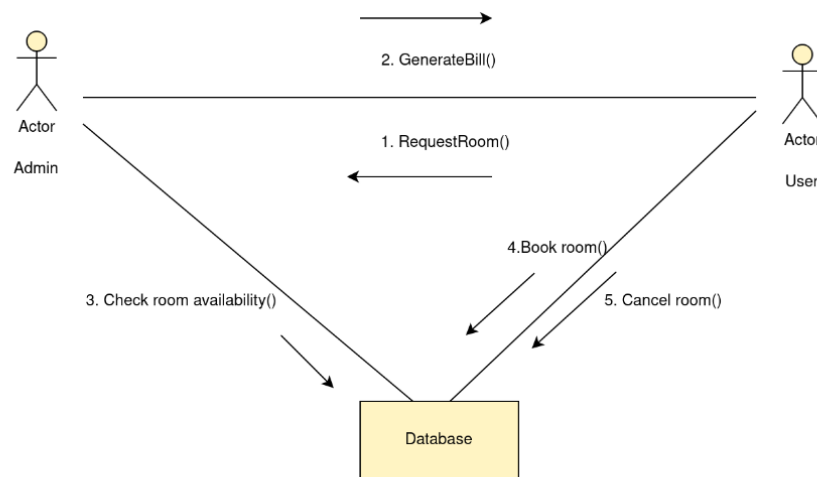
Sequence diagram:

A sequence diagram shows how actors interact with the system over time. It displays the order of messages exchanged between actors and the system to perform tasks such as booking rooms or processing payments. The diagram focuses on the sequence of actions, helping visualize the flow of operations in the system.



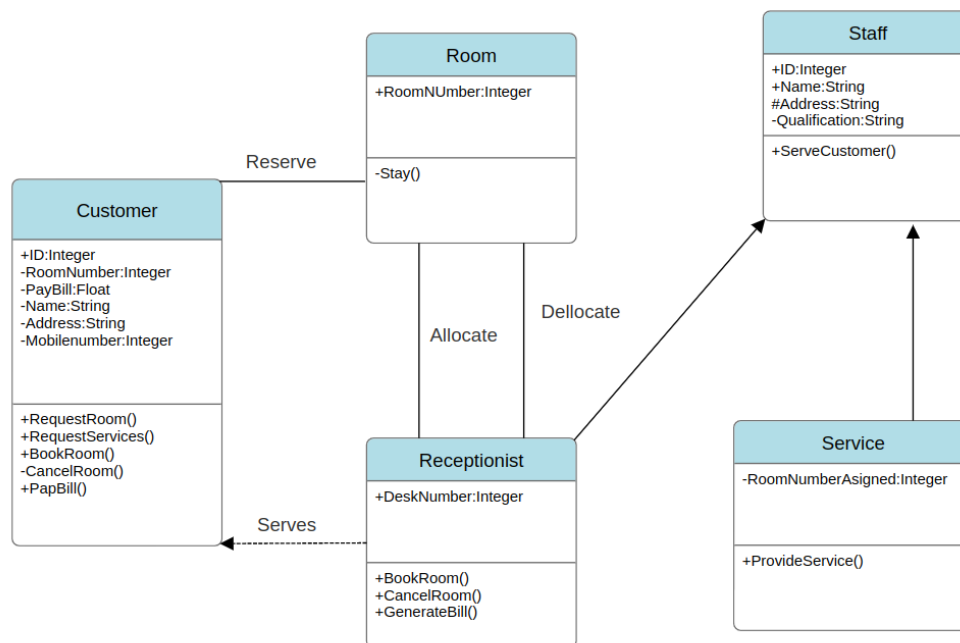
Collaboration diagram:

A collaboration diagram shows how actors and objects interact within a system. It focuses on the relationships between components, highlighting the messages exchanged to complete tasks. In the Hotel Management System, the diagram illustrates how the Receptionist and Administrator interact with different system objects to perform tasks, emphasizing the structure and flow of communication between them.



Class diagram:

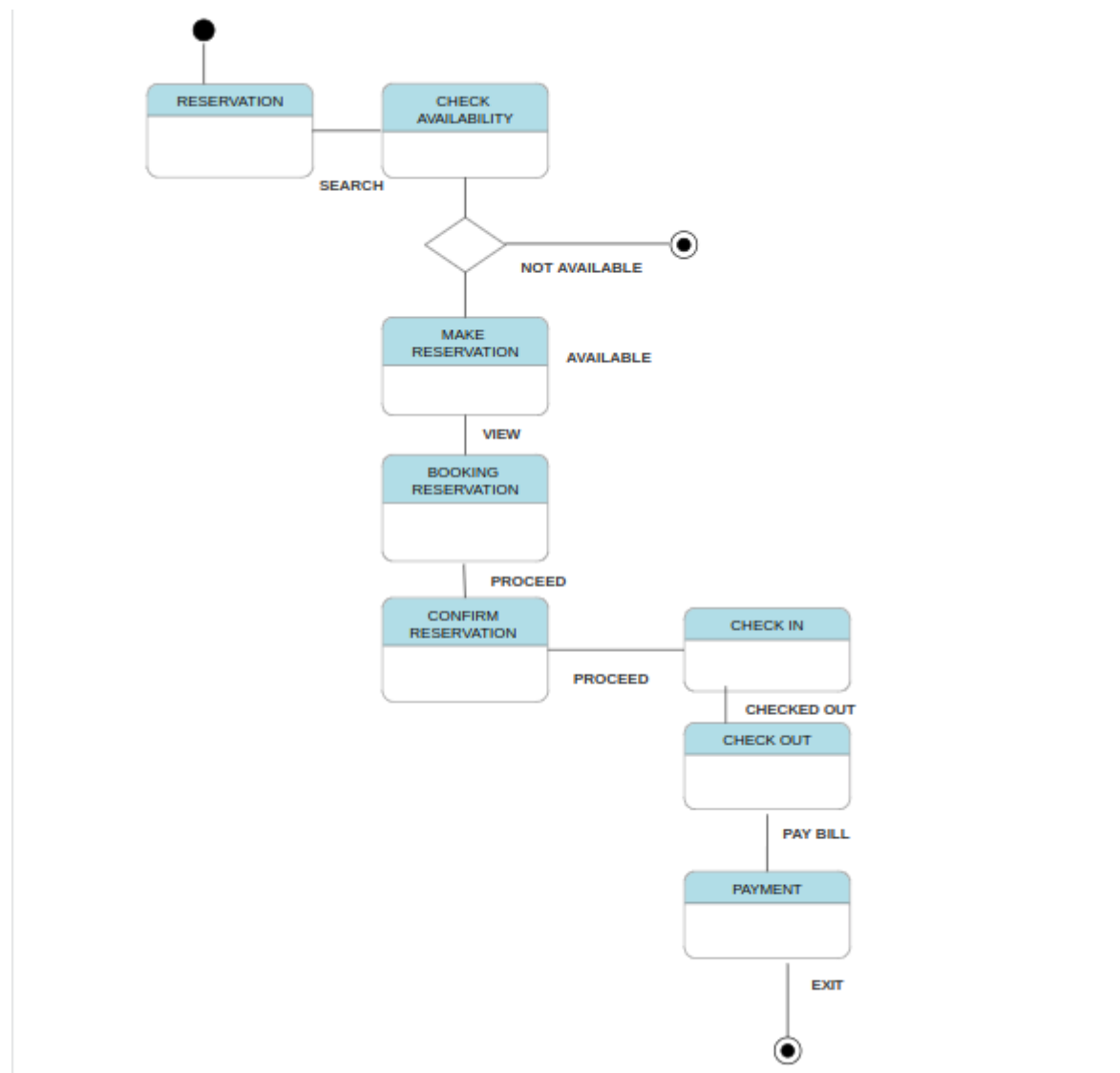
A class diagram shows the structure of a system by defining its classes, attributes, methods, and relationships. It represents real-world entities and how they interact within the system. The diagram helps visualize the system's architecture and the connections between its components. For a Hotel Management System, you'd have classes like Guest, Room, Reservation, and Billing, each with their own details, like the guest's name or the room number, and functions, such as checking in or processing payments.



Class Diagram for Hotel Mangement System

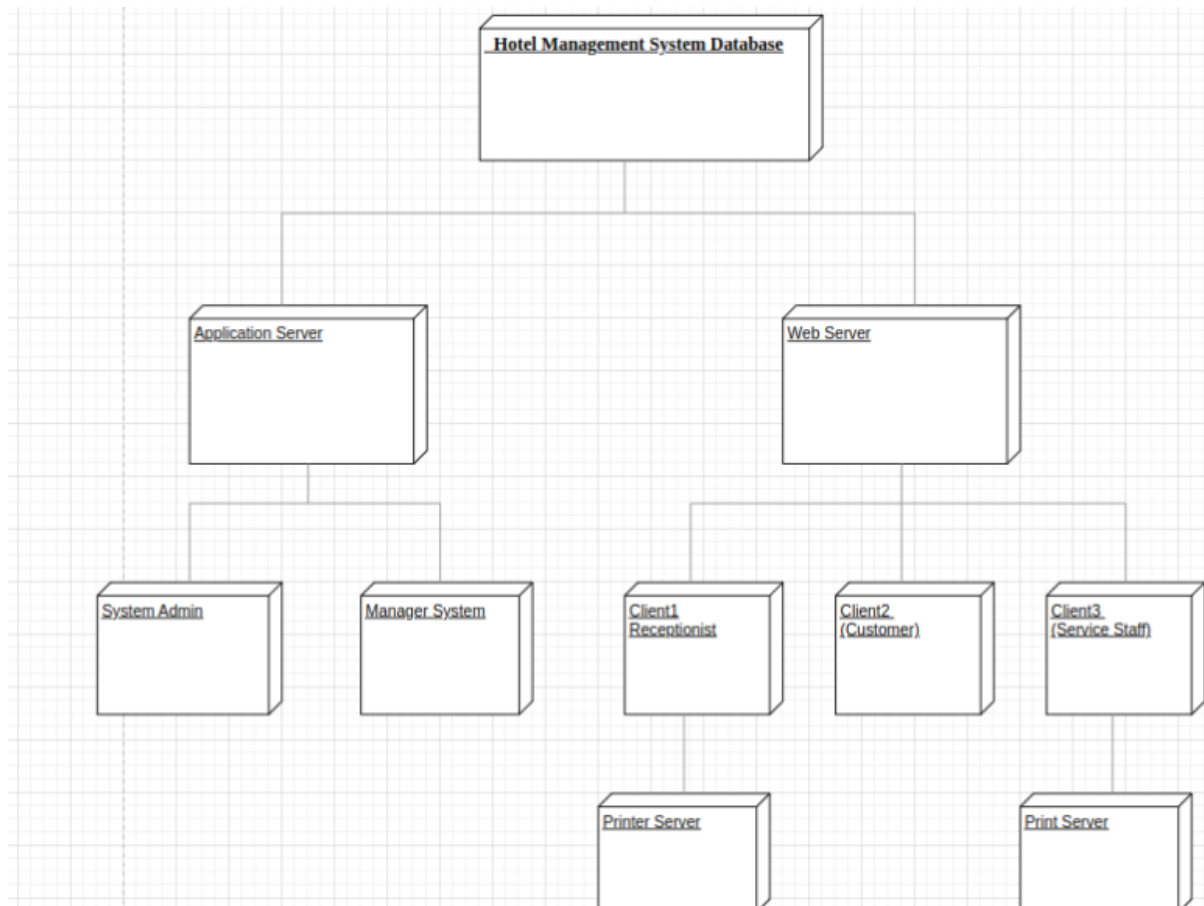
State chart diagram:

A state chart diagram shows the different states an object can be in and how it moves between those states based on events. In a Hotel Management System, a guest might move from "Reserved" to "Checked-in" or "Checked-out" depending on actions taken. It helps track the changes an object goes through over time.



Deployment diagram:

A deployment diagram shows the physical setup of a system, including hardware components and their connections. In a Hotel Management System, it illustrates how devices like servers and reception computers are linked, showing the system's architecture.



Component diagram:

A component diagram shows the major components of a system and how they interact. It focuses on the system's structure by breaking it down into smaller, functional pieces. In a Hotel Management System, components might include the booking system, billing system, and database, each connected to show how they work together to handle tasks like reservations and payments.

