SOFTWARE ENGINEERING PROJECT

COMPUTER SCIENCE & ENGINEERING

(Artificial Intelligence & Machine Learning)

Submitted By

23WH1A6613 M.AKSHARA

23WH1A6614 M.VAISHNAVI PRASAD

23WH1A6634 M. VEDA SATHVIKA

23WH1A6641 SANIA IQBAL

23WH1A6644 V.SRESHTITHA

23WH1A6645 M. PAVANI

23WH1A6655 M. SUVARNA

24WH5A6607 B.NANDINI

under the esteemed guidance of

Ms. V. ASHA

Assistant Professor CSE (AI & ML)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(Artificial Intelligence & Machine Learning)

BVRIT HYDERABAD COLLEGE OF ENGINEERING FOR WOMEN

(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)

Accredited by NAAC with A Grade

Bachupally, Hyderabad – 500090

BVRIT HYDERABAD
COLLEGE OF ENGINEERING FOR WOMEN

(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)
Accredited by NAAC with A Grade
Bachupally, Hyderabad – 500090

Department of Computer Science & Engineering
(Artificial Intelligence & Machine Learning)

CERTIFICATE

This is to certify that the Real time research project entitled "STOCK MAINTENANCE SYSTEM"is a bonafide work carried out **by Ms. M. AKSHARA(23WH1A6613)**, **Ms. M. VAISHNAVI PRASAD (23WH1A6614), Ms. M. VEDA SATHVIKA (23WH1A6634), Ms. SANIA IQBAL(23WH1A6641), Ms. V. SRESHTITHA (23WH1A6644)**, **Ms. M. PAVANI (23WH1A6645), Ms. M. SUVARNA (23WH1A6655), Ms. B. NANDINI (24WH5A6607)** in partial fulfilment for the award of B.Tech degree **in Computer Science & Engineering (AI & ML), BVRIT HYDERABAD College of Engineering for Women, Bachupally, Hyderabad,** affiliated to Jawaharlal Nehru Technological University Hyderabad, under my guidance and supervision. The results embodied in the project work have not been submitted to any other.

Internal Guide
Ms. V. ASHA
Assistant Professor
Dept of CSE(AI&ML)

Hea of the Department
Dr. B Lakshmi Praveena
HOD & Professor
Dept of CSE(AI&ML)

INDEX

S.no	Title of the experiment	Page no	Remarks
1.	Stock Maintenance	4-12	
2.	Hotel Management	12-26	
3.	Hospital Management	26-40	

STOCK MAINTENANCE

Problem Statement

The current stock maintenance system of the bookstore is inefficient, making it difficult to manage inventory, track sales, and generate accurate reports. Employees struggle to update information on book details, such as cost, edition, and author, leading to errors in stock records. The system lacks secure access control, which raises concerns about data security. Additionally, the inability to efficiently handle purchase orders and sales reporting hampers decision -making. Customers can view book availability and prices, but the system does not allow for seamless updates. A new, secure, and user-friendly stock maintenance system is required to address these issues.

Software Requirement Specification (SRS)

1. Introduction:

The purpose of this SRS is to outline the requirements for a new Stock Maintenance System for a bookstore, which aims to replace the existing inefficient system. The system will ma nage book inventory, track sales, handle purchase orders, and generate sales reports. It will also provide secure access for employees and allow customers to view book availability and prices. The system will be a Windows-based desktop application that enhances usability, security, and data management.

2. User Characteristics:

Employees:

- Can access and manage book inventory, record sales, and handle purchase orders.

- Limited to authorized actions based on security permissions.

Customers:

- Can view book availability and pricing information but cannot modify data.

3.System Modules:

Inventory Management:

Allows employees to add, update, and delete book details (title, author, cost, edition, etc.).

Sales Management:

Records sales transactions and generates sales reports.

Purchase Order Management:

Manages the creation, update, and tracking of purchase orders.

Employee Preferences and Security:

Provides secure login for employees and enables them to modify personal preferences.

Customer View Module:

Displays book availability and pricing without allowing modifications.

3. Functional Requirements:

- Employees can manage book inventory and update book details.
- Sales transactions and purchase orders are recorded and managed.
- The system generates sales reports based on total sales.
- Employees can log in securely, with restricted access to sensitive functions.
-Customers can view book availability and prices without making changes.

4. Non-Functional Requirements:

Usability: A user-friendly, Windows-based interface for employees and customers.
Security: Secure login and access control for sensitive data (sales, purchase orders).
Performance: Timely processing of inventory updates and report generation.

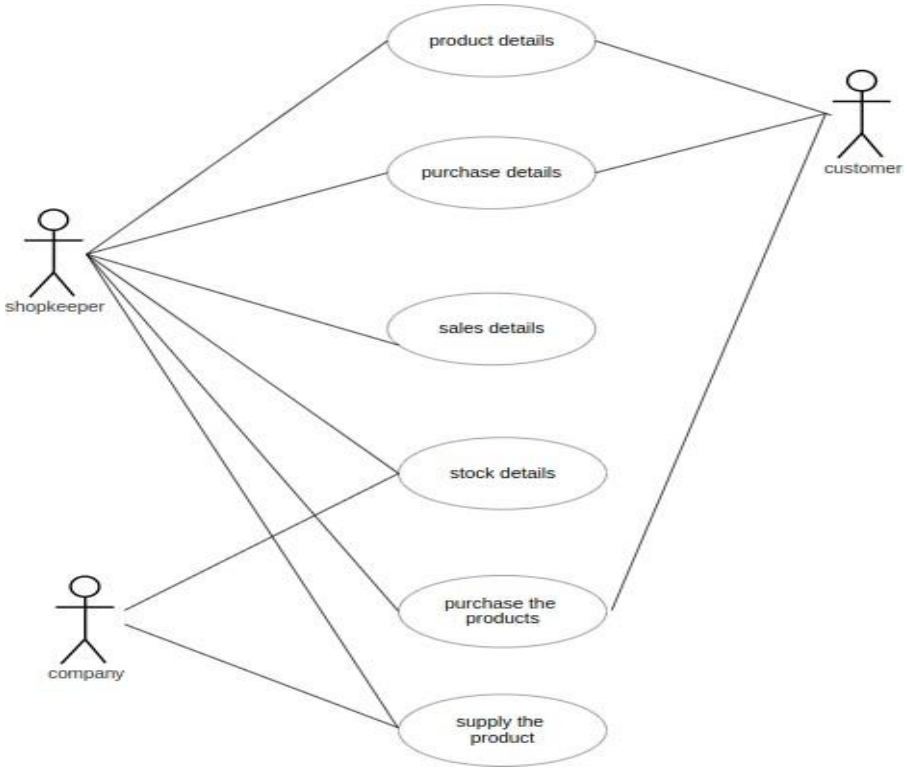
Scalability: Able to handle growing inventory and transaction volumes.

Reliability: High availability with minimal data loss or corruption.

Maintainability: Easy to update and maintain with minimal downtime

UML DIAGRAMS:

USE-CASE DIAGRAM: The diagram illustrates a retail use case where a shopkeeper manages interactions with both customers and a company. The system tracks product details, allowing customers to view and purchase items, while the shopkeeper records sales and purchase details. It also manages stock details, ensuring the shopkeeper can monitor inventory and reorder products as needed. The shopkeeper places orders with the company, which supplies the products to maintain stock. Overall, the system seamlessly handles product flow from the supplier to the shopkeeper and ultimately to

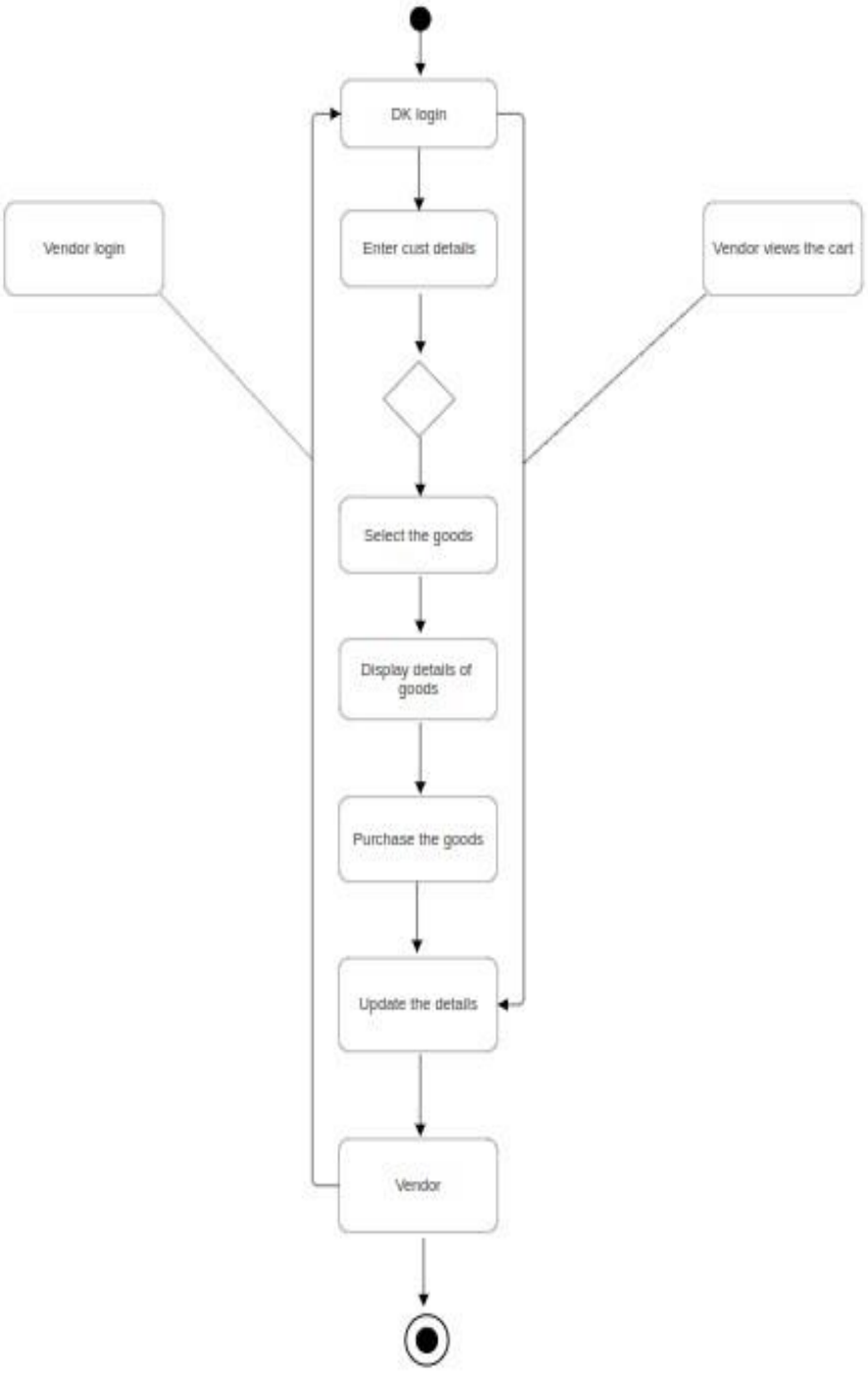


the customer.

ACTIVITY DIAGRAM :

The activity diagram depicts the process of stock maintenance, starting with the vendor logging in via the DK login interface. After entering customer details, a decision is made to select goods. Once selected, the details of the goods, such as availability and pricing, are displayed.

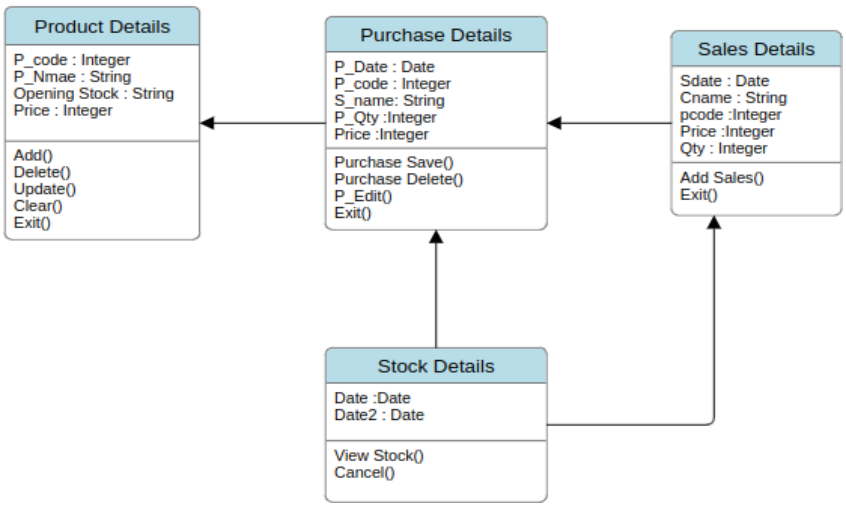
The vendor then proceeds with the purchase, and the system updates the relevant stock information. The process concludes with the vendor completing the transaction. Additionally, the diagram includes alternate flows for direct vendor login or viewing the cart before selecting goods, reflecting different paths for stock management.


DK login
Vendor login
Enter cust details
Vendor views the cart
Select the goods
Display details of goods
Purchase the goods
Update the details
Vendor

CLASS DIAGRAM:

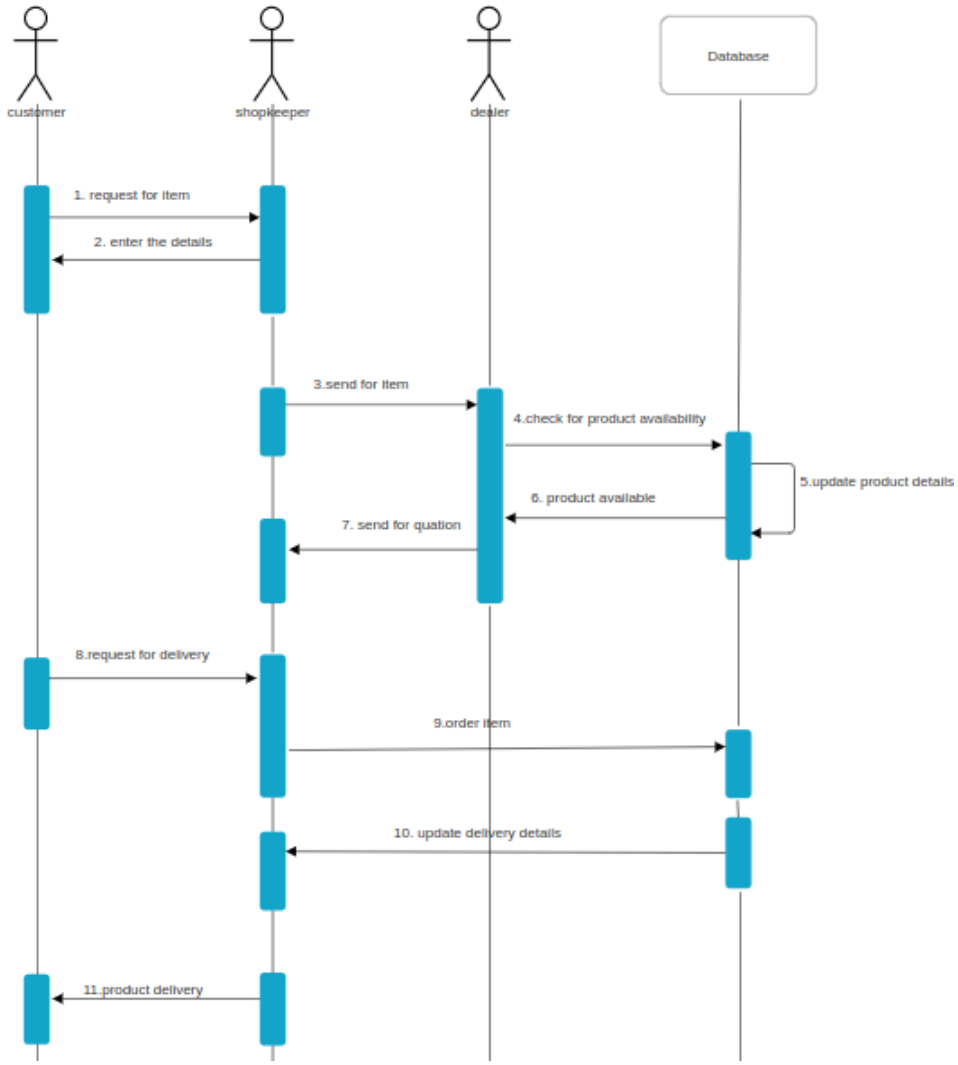
The class diagram represents a stock management system with four key components: Product Details, Purchase Details, Sales Details, and Stock Details. The Product Details class handles product information like code, name, stock, and price, with methods to add, update, or delete products. Purchase Details manages purchase transactions, tracking dates, supplier names, and quantities, with options to save or edit purchases. Sales Details monitors sales, recording customer names, sale dates, and quantities, with a method for adding sales records. Lastly, Stock Details allows viewing and canceling stock records. Together, these classes ensure efficient product, purchase, sales, and stock management.

Class Diagram:



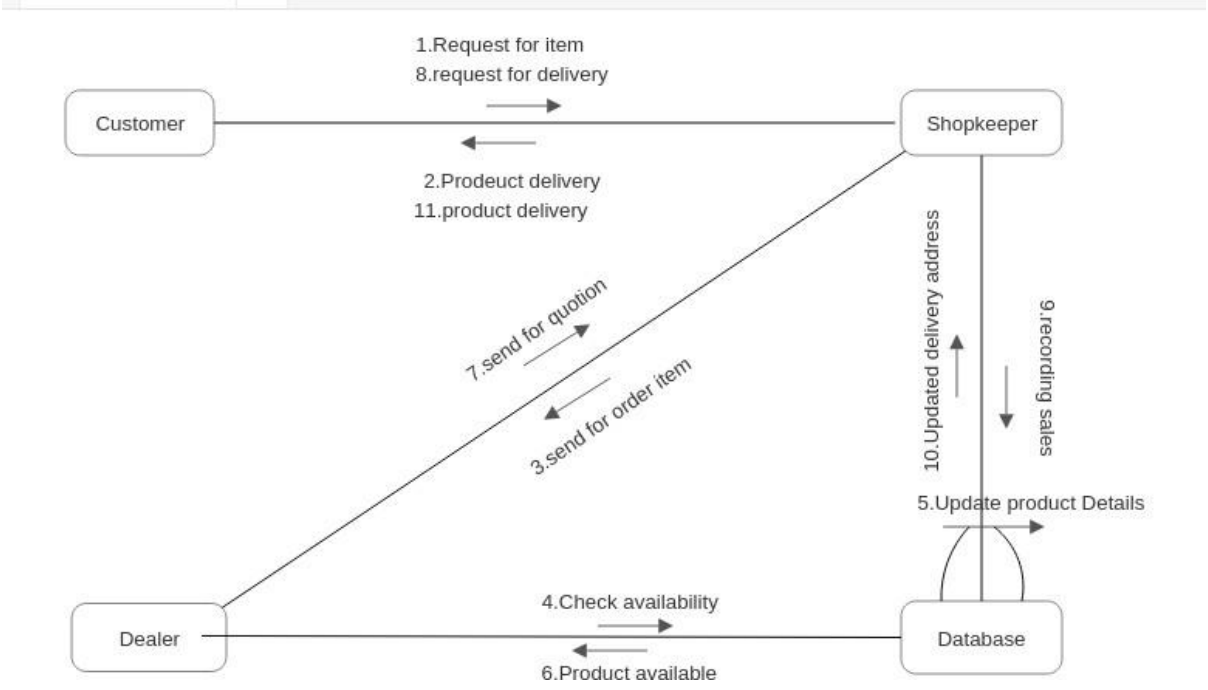
SEQUENCE DIAGRAM:

The sequence diagram for stock maintenance illustrates the interactions between a stock manager, the inventory system, and the database. Initially, the stock manager requests current stock levels, prompting the inventory system to retrieve this data from the database and display it. When updating stock levels after receiving new inventory or sales, the stock manager inputs the changes, which the inventory system processes and confirms with the database. Finally, the stock manager can generate stock reports, leading the inventory system to fetch relevant data from the database and present it back. Error handling and low stock notifications are also considered**.**



COLLABORATION DIAGRAM:

The collaboration diagram for stock maintenance visually represents the interactions and relationships among key participants: the stock manager, the inventory system, and the database. Initially, the stock manager communicates with the inventory system to request the current stock levels, initiating a sequence of interactions. The inventory system then accesses the database to retrieve this information and returns it to the stock manager. When the stock manager receives new inventory or processes sales, they update stock levels by sending the new data to the inventory system. The inventory system validates this information and updates the database, confirming the changes. Furthermore, the stock manager can request comprehensive stock reports, leading the inventory system to query the database for relevant data and present the report. The diagram emphasizes how these components work together to ensure accurate stock management, highlighting their interdependencies and the flow of information critical for maintaining optimal inventory levels.

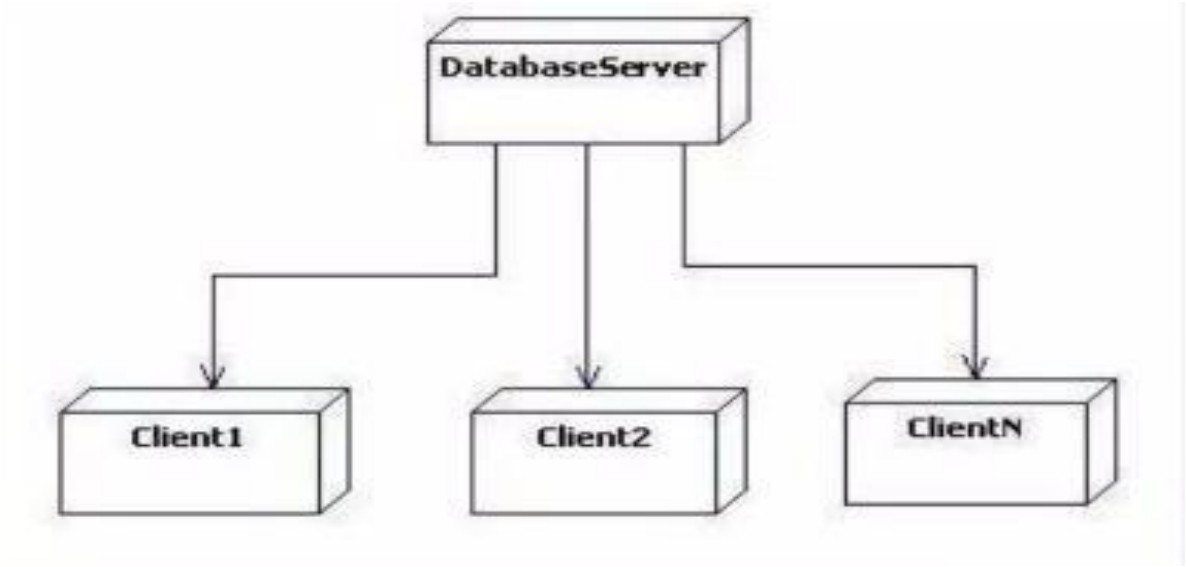


STATE CHART DIAGRAM :

The state chart diagram for stock maintenance outlines the various states and transitions involved in managing inventory levels effectively. It begins with the initial state, where stock is considered "Available" or "Low." When a stock level falls below a predetermined threshold, the system transitions to the "Low Stock" state, prompting the stock manager to take action. In this state, the stock manager can initiate a restocking process, leading to a transition to the "Restocking" state. Once the inventory is replenished, the system transitions back to the "Available" state. Additionally, if stock levels are updated due to sales, the system may move to the "Stock Updated" state, reflecting the changes in inventory. Throughout these transitions, the state chart diagram captures the conditions and events that trigger each state change, providing a clear visualization of how the system reacts to different scenarios in stock management and ensuring efficient inventory control

DEPLOYMENT DIAGRAM & COMPONENT DIAGRAM:

The deployment diagram for stock maintenance illustrates the physical architecture of the system, showcasing the nodes and their relationships. It typically includes components like the inventory management server, the database server, and client devices used by stock managers. The inventory management server hosts the inventory system application, responsible for processing stock requests, updates, and reports. The database server stores all stock-related data, ensuring efficient retrieval and updating of inventory information. Client devices, such as desktops or tablets, interact with the inventory system to manage stock levels. In contrast, the component diagram provides a detailed view of the system's software architecture, depicting the various components, their interfaces, and relationships. Key components include the user interface, inventory processing module, database access layer, and reporting module. This diagram highlights how these components collaborate to ensure seamless functionality, facilitating effective stock management by clearly defining the interactions between software elements and their dependencies in the system.



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 functionalitiesneeded 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.

1. 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.
2. 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.
3. 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.
4. 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

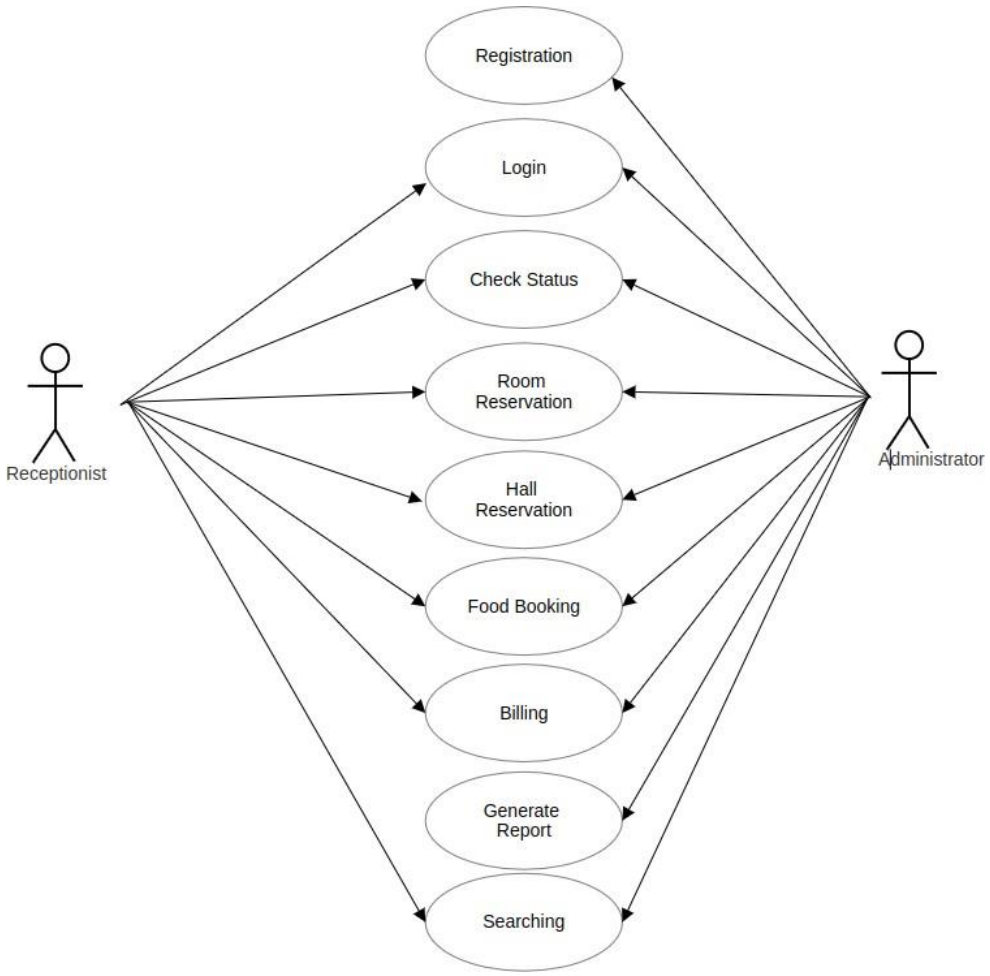
Hardware components:

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

UML Diagrams:

Use-case diagram:

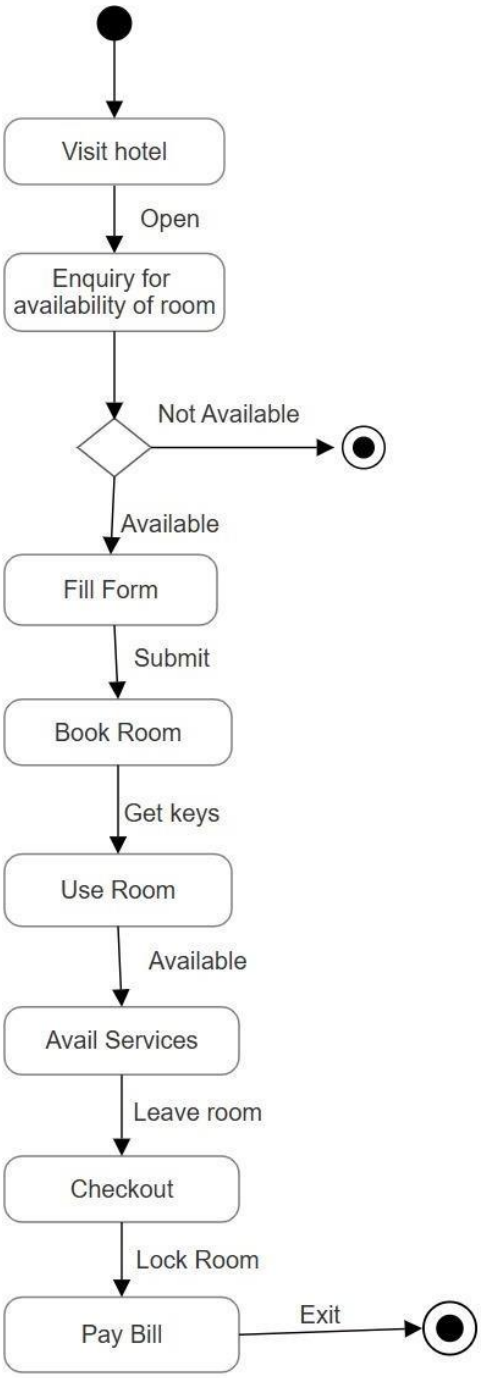
A use case diagram is a tool to visualize system requirements by showing how actors interact with the system. In the Hotel Management System, there are two main actors: the Receptionist and the Administrator. The Receptionist handles tasks like guest registration, booking rooms and halls, managing food 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:

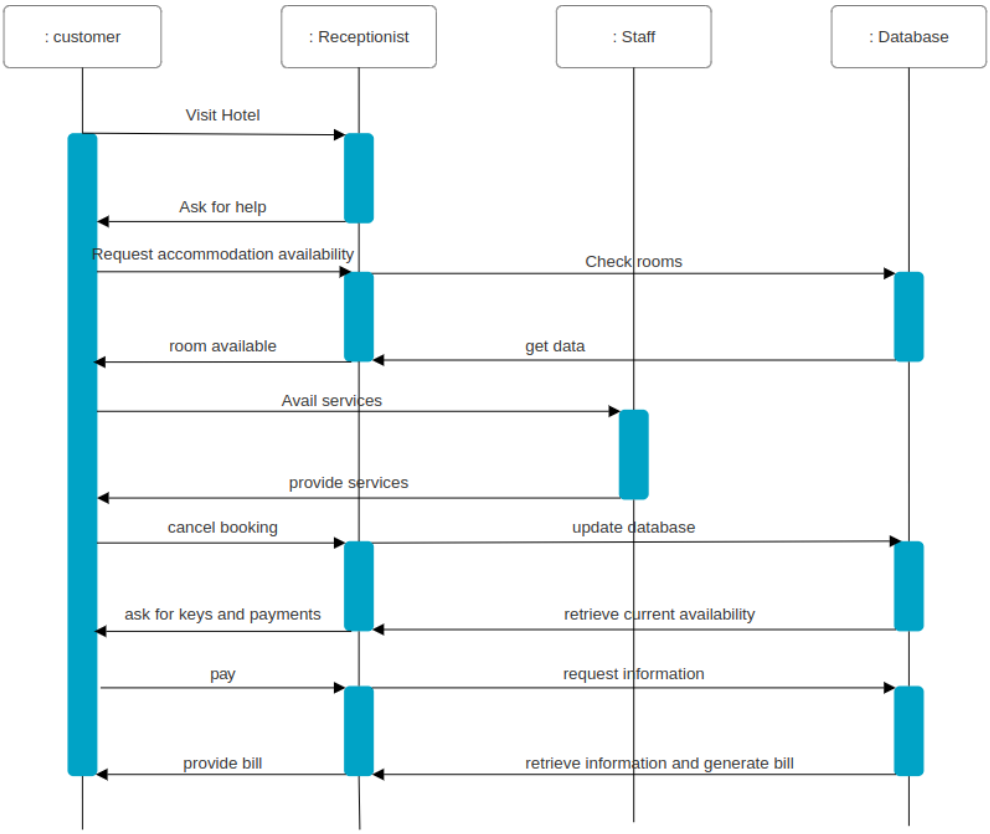
An activity diagram shows the flow of actions in a system. In the Hotel Management System, the Receptionist registers guests, checks availability, books room, manages orders, and processes billing. Decisions like availability orpayment affect the next steps. The Administrator oversees and manages these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

points to represent the workflow



Sequence diagram:

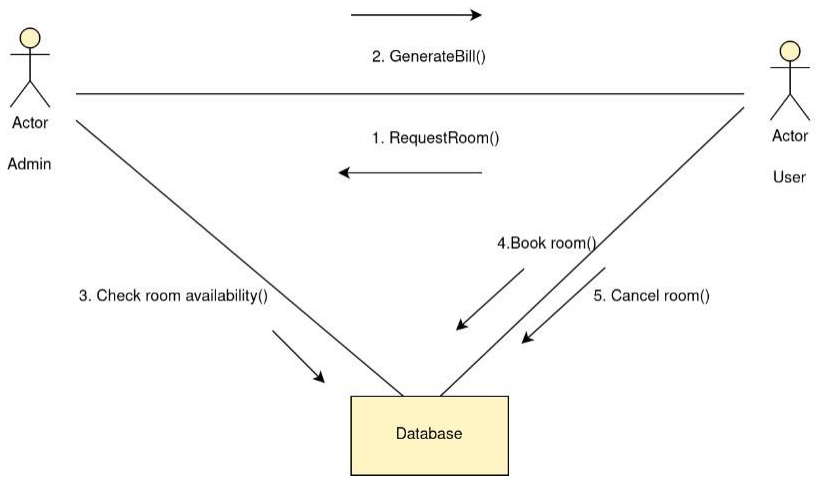
A sequence diagram shows how actors interact with the system over time. It displays the order of messages exchanged between actors and the system to perform tasks such as booking rooms or processing payments. The diagram focuses on the sequence of actions, helping visualize the flow of operations in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



Collaboration diagram:

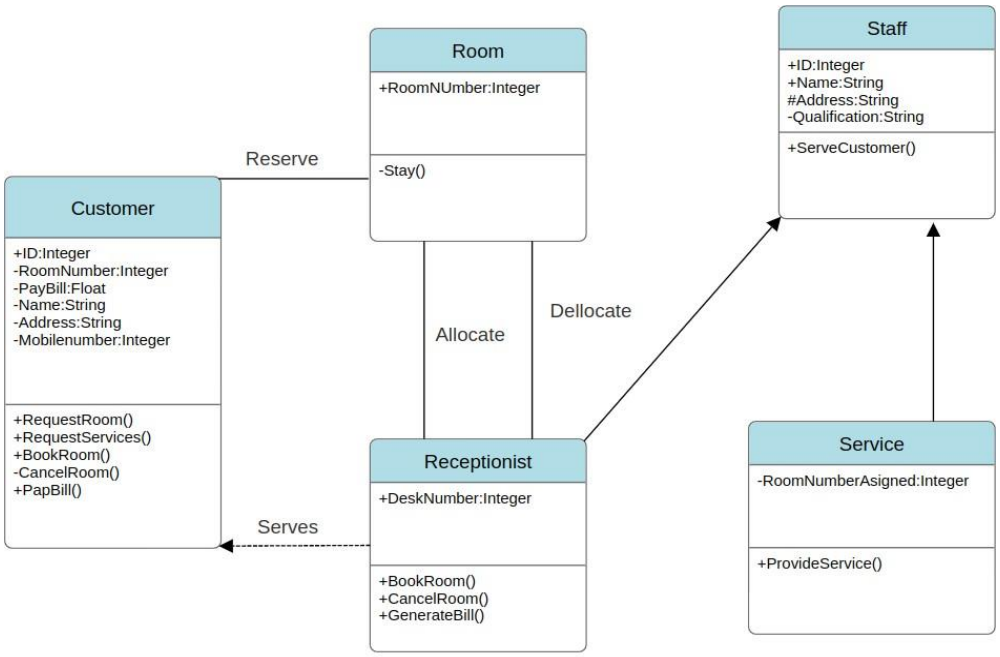
A collaboration diagram shows how actors and objects interact within a system. It focuses on the relationships between components, highlighting the messages exchanged to complete tasks. In the Hotel Management System, the diagram

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'sarchitecture 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.

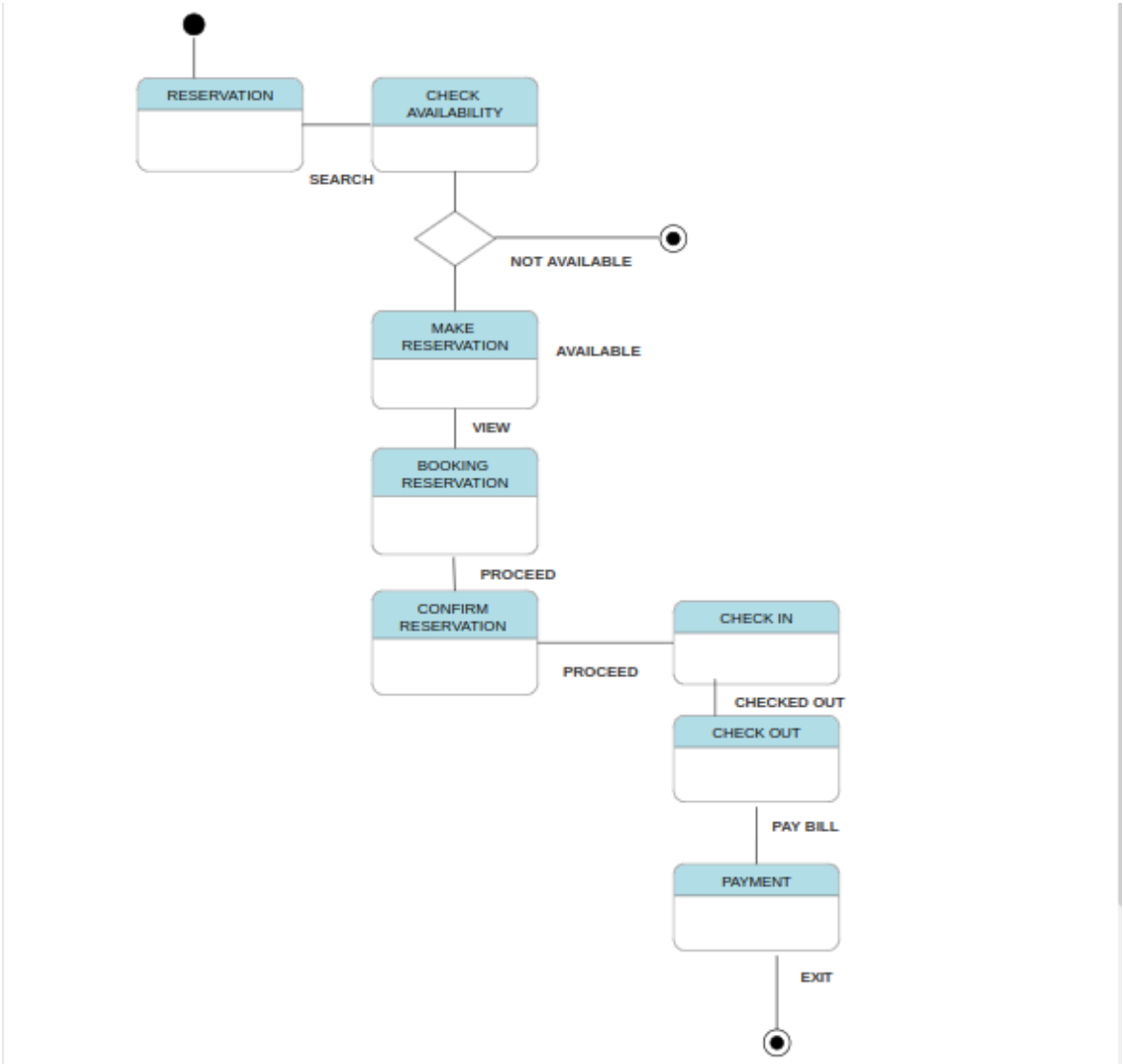


Class Diagram for Hotel Mangement System

State chart diagram:

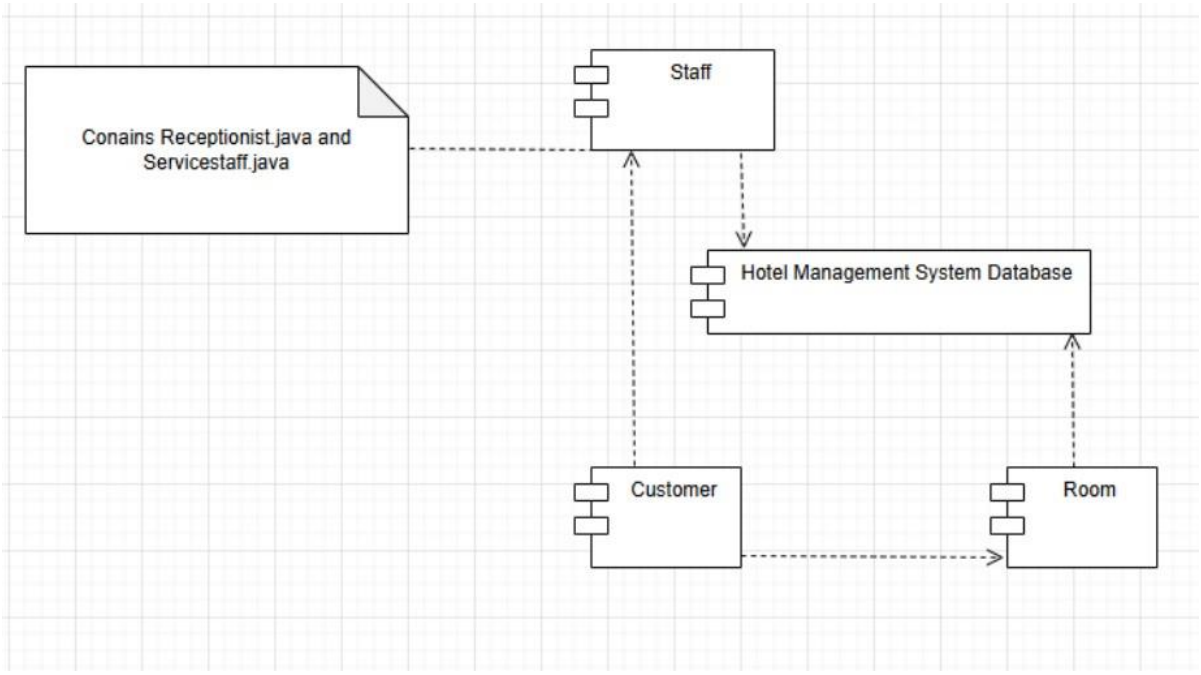
A state chart diagram shows the different states an object can be in and how it moves between those states based on events. In a Hotel Management System, a guest might move from "Reserved" to "Checked-in" or "Checked-out"

depending on actions taken. It helps track the changes an object goes through over time.



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 howthey work together to handle tasks like reservations and payments.



HOSPITAL MANAGEMENT SYSTEM

Problem statement: The Hospital Management System (HMS) addresses the challenges faced by healthcare facilities in managing patient records, appointments, billing, and inventory. Manual processes often lead to

inefficiencies, data errors, and delays in service delivery, impacting patient care. The system aims to automate and centralize these functions, improving operational efficiency and communication. It

provides seamless management of patient data, scheduling, and billing, ensuring accurate records and compliance with regulations. Ultimately, the HMS enhances the overall hospital experience for both patients and staff, optimizing healthcare delivery

Features:

- **Admin login & admin dashboard:** The admin can securely log in to the system, accessing a centralized dashboard to manage hospital operations, view statistics, and oversee users and appointments.
- **User registration:** Patients and hospital staff can register on the system, providing necessary personal details and medical information for efficient service management.
- **Booking system:** Patients can book appointments with doctors through an easy-to-use interface, choosing their preferred doctor and available time slots.
- **Approving/Disapproving request:** The admin or relevant authority can approve or disapprove patient requests, such as appointment bookings or service access, based on availability and requirements.

Software Requirements Specification:

- **Functional requirements:**

These describe what the system should do, i.e., the core functionalities neededto manage the operations of a Hospital Management System.

1 Patient Management

- **Registration:** Create and update patient profiles.
- **EHR Management:** Maintain and update patient's medical history, diagnoses, and treatment records.
- **Patient Search:** Easily retrieve patient records using patient ID, name, or contact details.

2 Appointment Management

- **Appointment Scheduling:** Patients can book, reschedule, or cancel appointments.
- **Doctor's Schedule:** Doctors can update their availability, view, and manage appointments.
- **Appointment Notifications:** Send reminders to patients and doctors via email or SMS.

3 Billing and Payments

- **Invoice Generation:** Generate invoices for consultations, procedures, andtreatments. **Payment Processing:** Integrate with payment gateways for onlinepayments.
- **Insurance Integration:** Handle insurance claims and generate insurance reports.

4 Inventory and Pharmacy Management

- **Stock Management:** Track and manage stock levels for medicines, medical supplies, and equipment.
- **Inventory Alerts:** Notify administrators of low stock or expired items.
- **Pharmacy POS System:** Simplify medicine dispensing and billing at the pharmacy.

1 Reporting and Analytics

- **Operational Reports:** Generate reports for patient inflow, financials, inventory levels, and staff performance.
- **Regulatory Compliance Reports:** Produce reports for healthcare regulations (e.g., HIPAA compliance).

- **Dashboard:** Visual display of hospital performance metrics, such as patient wait times and revenue.

2 Security and Access Control

- **Role-based Access Control:** Different user roles (e.g., Administrator, Doctor, Patient) with specific access permissions.
- **Audit Logs:** Track system usage and changes made to sensitive data.
- **Data Encryption:** Ensure patient data is encrypted during storage and transmission to maintain privacy.

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 must provide high responsiveness and efficiency under normal and peak usage conditions. All user actions, such a booking appointments, retrieving medical records, and generatinginvoices, should have a response time of no more than 2 seconds.

   Additionally, the system should be capable of handling a

   significant number of concurrent users (e.g., 500+ simultaneous users) without experiencing performance degradation..

2. Scalability:

   The HMS should be designed to scale as the hospital grows. It must accommodate an increasing number of patients, doctors, and other users. The system architecture should allow for the addition of new modules (e.g., additional branches, new departments, or specialized services) and should support horizontal scaling, allowing it to handlemore users or transactions as needed, without requiring a completeredesign of the system.

3. Availability:

   - The system should be available 99.9% of the time, ensuring that users can access the hospital services without interruptions. This

4. includes considerations for system uptime, backup mechanisms, and disaster recovery procedures. Planned maintenance should be scheduled during off-peak hours, with advance notice given to users. Security:

   - The system must meet high-security standards to protect sensitive

patient data and comply with regulations such as HIPAA or GDPR. This includes secure user authentication (e.g., multi-factor

authentication), encryption of patient records during transmission and storage, access control based on user roles, and regular security audits to detect vulnerabilities. The system must also prevent unauthorized access to medical and financial records.

5. Usability:
   - The system must be user-friendly, ensuring that all types of users, from patients to hospital staff, can navigate and operate the system easily. The interface should be intuitive, with clear labeling and
6. minimal learning curves. It should also be designed to accommodateusers with disabilities, meeting accessibility guidelines. Compatibility:
   - The HMS should be compatible with a wide range of operating systems (e.g., Windows, macOS, Linux) and devices (desktop computers, laptops, tablets, and smartphones). It should also support different web browsers (e.g., Chrome, Firefox, Safari) and allow integration with external systems, such as third-party payment gateways, laboratory systems, and medical equipment..
7. Maintainability:
   - The system should be designed for easy updates and maintenance with minimal downtime and clear documentation..
8. Backup and Recovery:
   - The system must implement automatic daily backups of all critical data (e.g., patient records, appointments, billing) to ensure data integrity. In case of system failure or data loss,
9. recovery processes should allow for quick restoration of data with minimal downtime, ensuring that operations can resumesmoothly without significant impact on hospital services Compliance:
   - The system must adhere to healthcare regulations such as HIPAA and GDPR to ensure the privacy and security of patient data. It should also meet industry standards for data protection and accessibility.

Software requirements:

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

Hardware components:

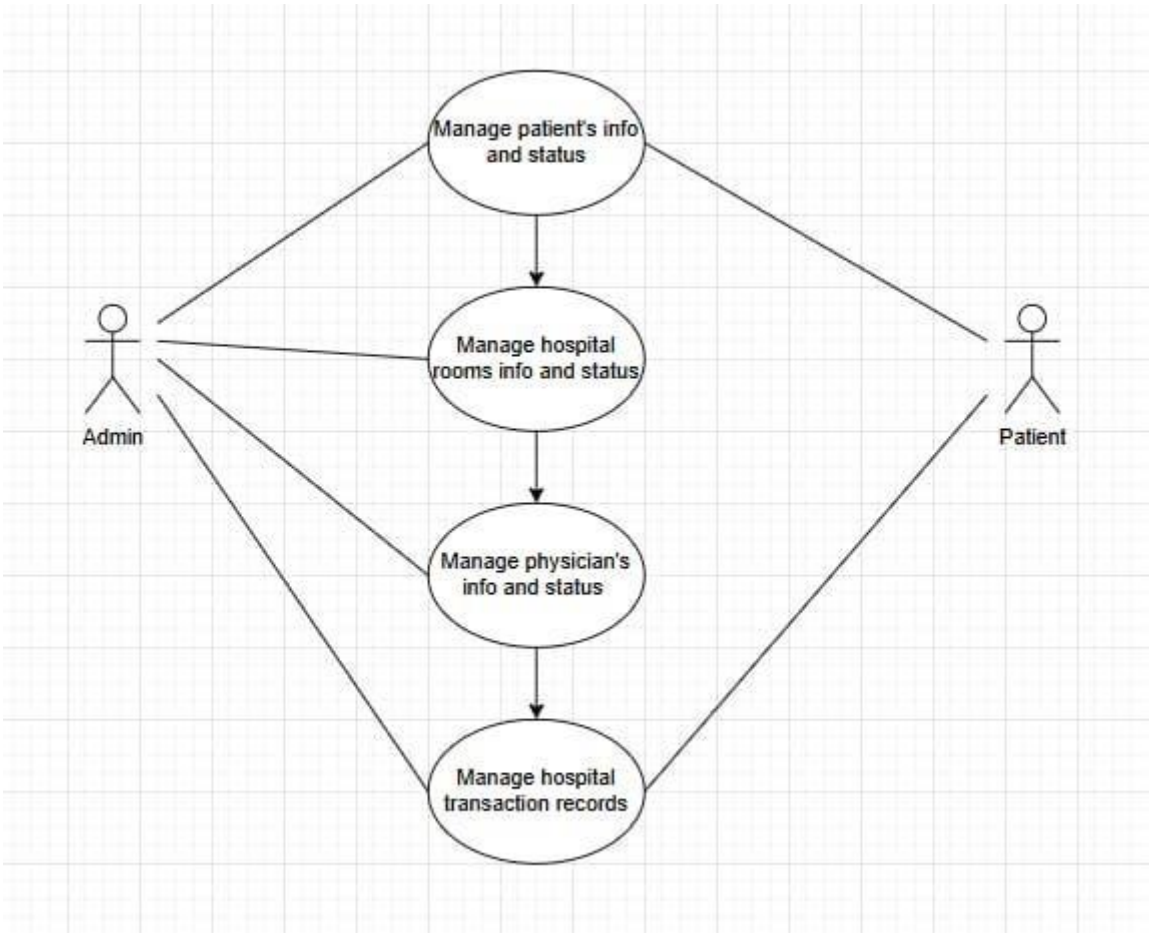
- Process - p4
- Hard disk - 5GB

Memory - 1GB RAM

UML Diagrams:

Use-case diagram:

The use case in a Hospital Management System enables patients to book consultations with doctors seamlessly. The patient logs into the system, selects the desired specialization, and views available doctors and time slots. After choosing a doctor and an appropriate time, the patient confirms the appointment, which is then updated in both the patient's and the doctor's schedules. Notifications are sent to both parties via SMS or email to ensure reminders. If no slots are available, the system suggests alternate dates or doctors with similar expertise. The receptionist can assist patients who are unable to book appointments themselves by performing the same process on their behalf. In case of emergencies or changes in the doctor's availability, the system notifies the patient to reschedule. This feature streamlines appointment management, reduces wait times, and enhances patient experience.

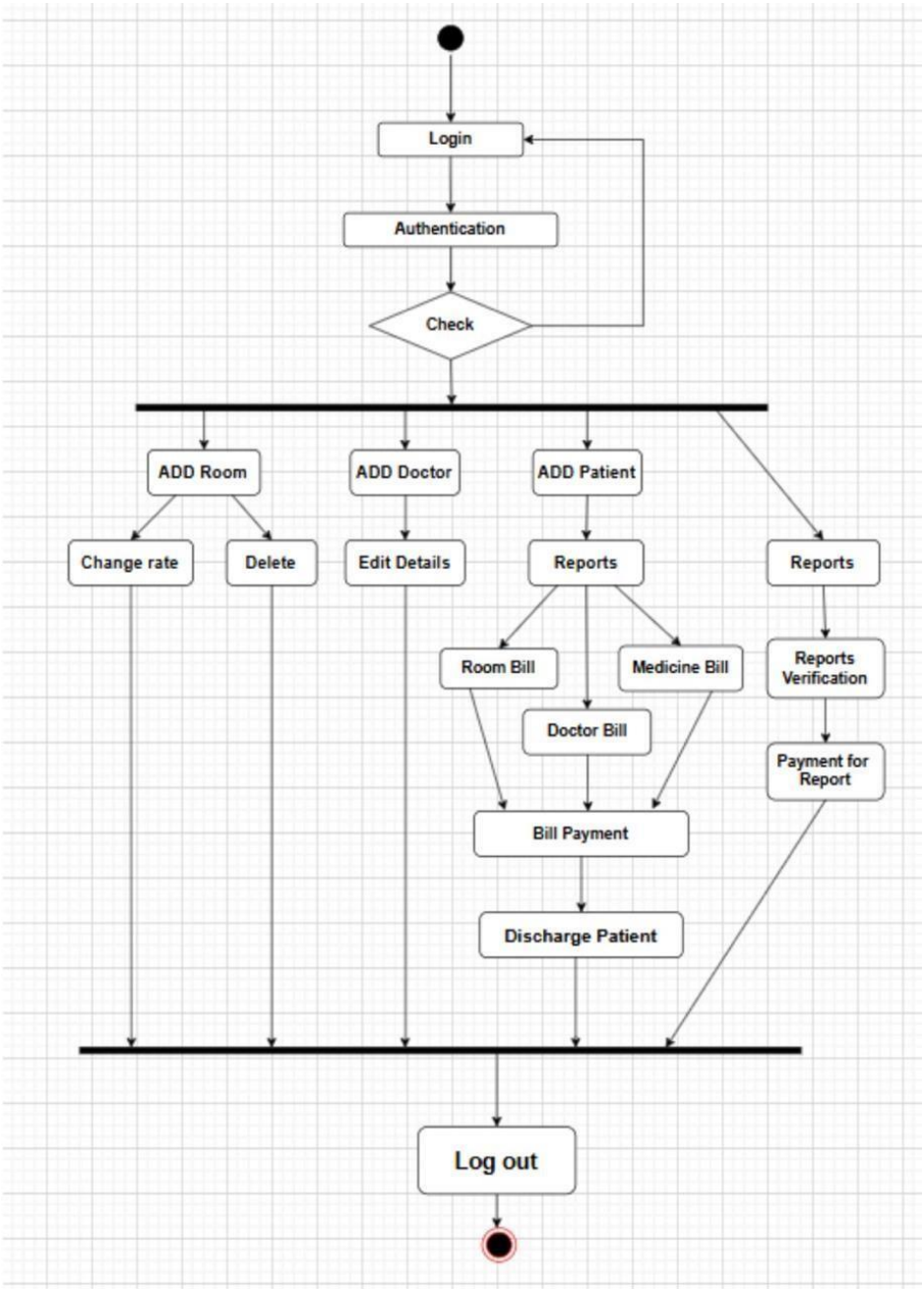


.

Use-case diagram for hospital management system

Activity diagram:

An activity diagram for a Hospital Management System outlines the workflowof processes such as patient registration, appointment scheduling, billing, andinventory management. It visualizes the flow of activities, from a patient booking an appointment to the completion of medical treatment and the finalbilling process. The diagram also includes decision points, like handling payment methods or managing inventory stock levels, ensuring clearunderstanding of system processes and roles.

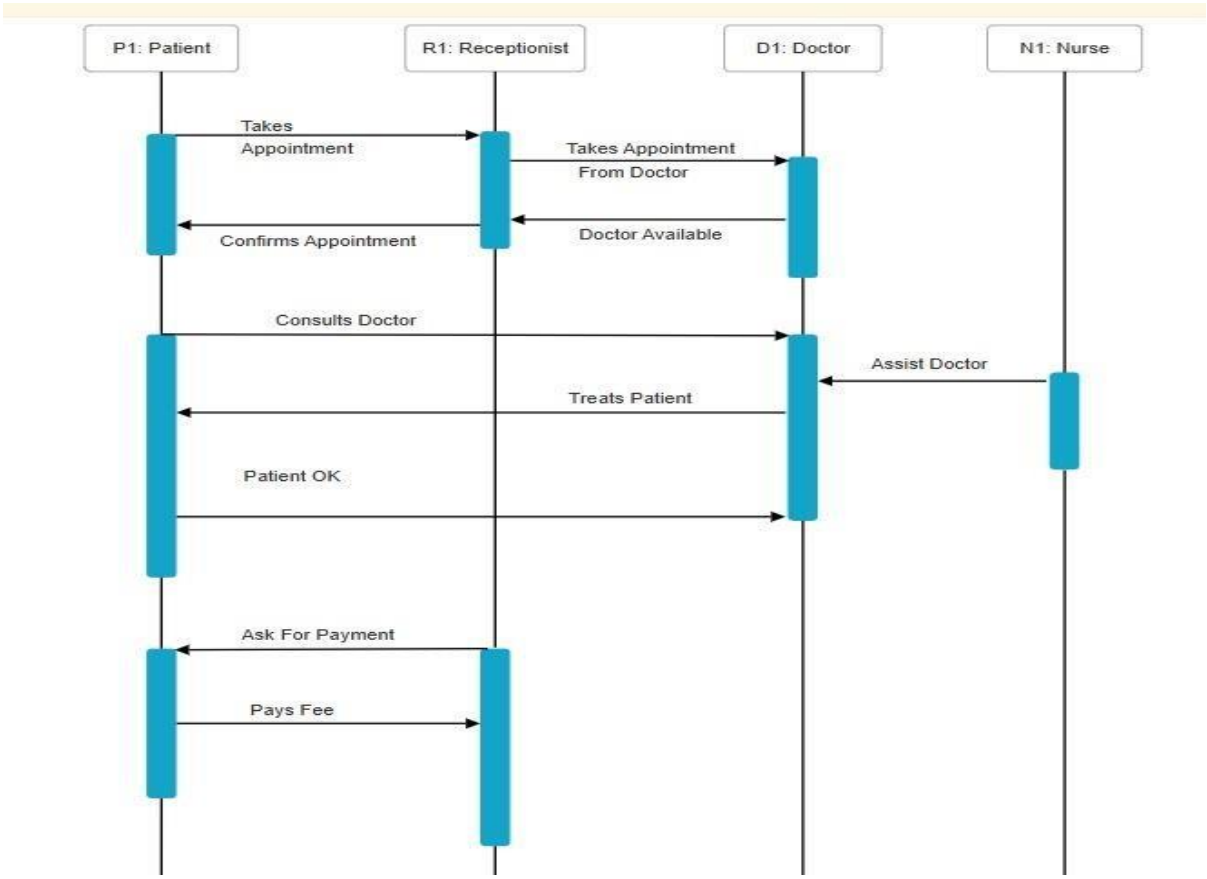

Login
Authentication
Check
ADD Room
ADD Doctor
ADD Patient
Change rate
Delete
Edit Details
Reports
Reports
Room Bill
Medicine Bill
Reports Verification
Doctor Bill
Payment for Report
Bill Payment
Discharge Patient
Log out

Sequence diagram:

A sequence diagram for a Hospital Management System illustrates the

interaction between objects for processes like appointment scheduling. It shows the sequence of messages exchanged between the patient, the system, and the doctor, starting from the patient requesting an appointment to the system confirming the booking. This diagram helps visualize the step-by-step communication flow, ensuring that each actor and system component. performs its role in a timely and synchronized manner.

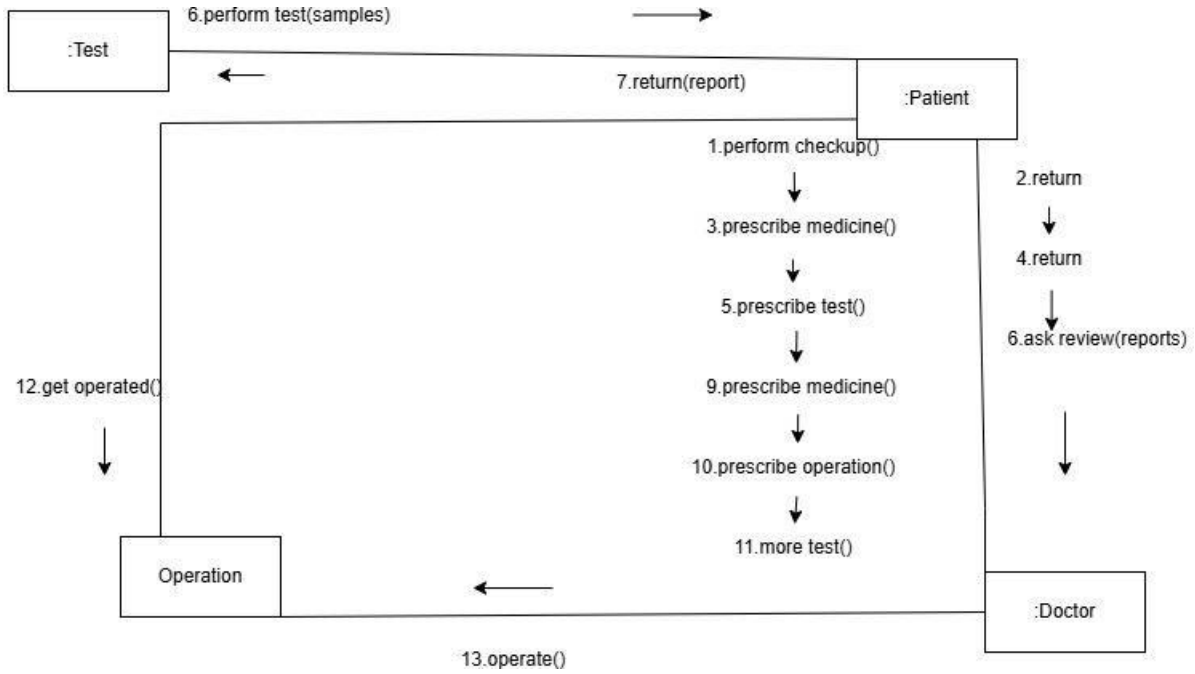
.



.

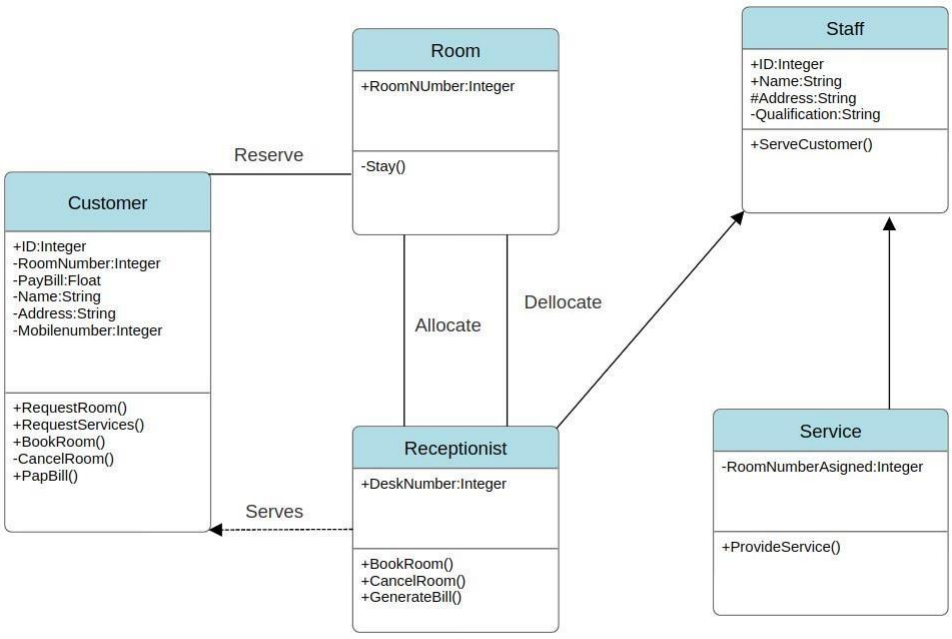
Collaboration diagram:

A collaboration diagram for a Hospital Management System represents the interaction between system components and actors, focusing on how objects collaborate to complete a process. For example, when a patient schedules an appointment, the diagram shows the interactions between the patient, the appointment system, the doctor's schedule, and notification services. This diagram highlights the relationships and communication paths between objects, making it easier to understand the flow of data and responsibilitieswithin the system.



Class diagram:

A class diagram for a Hospital Management System defines the structure of the system by representing key classes, their attributes, and methods. It includes classes such as Patient, Doctor, Appointment, Billing, and Inventory, with relationships like associations, inheritances, and dependencies between them. This diagram provides a blueprint for the system's object-oriented design, helping to visualize how different entities within the hospital interact and collaborate to manage operations efficiently..



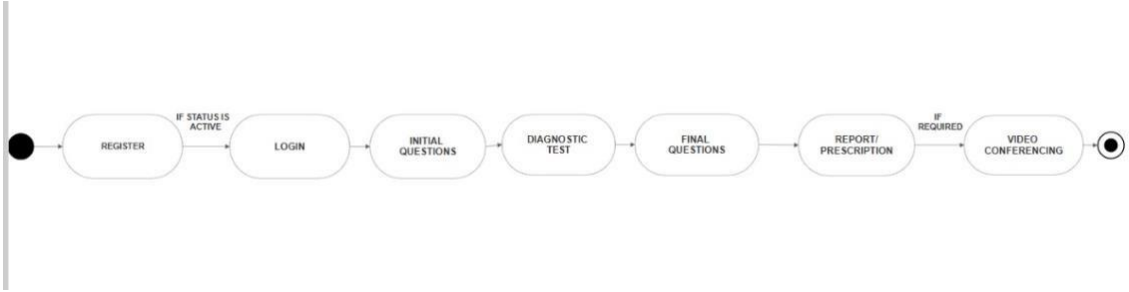
Class Diagram for Hotel Mangement System

State chart diagram:

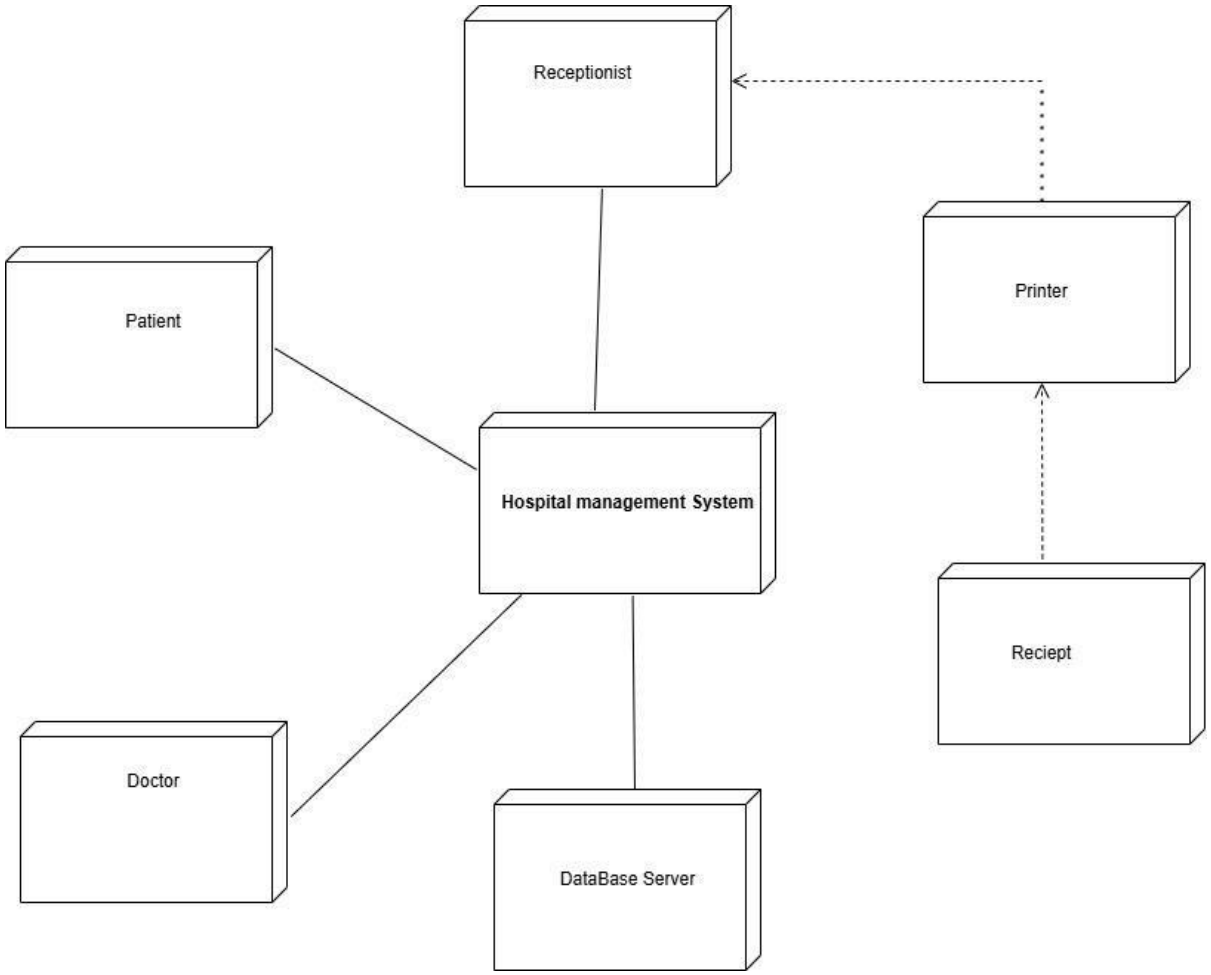
A state chart diagram for a Hospital Management System illustrates the

different states an object, such as an Appointment or Patient, can transition through during its lifecycle. For example, an appointment can move through states like Scheduled, In Progress, Completed, and Canceled based on

interactions with the system. This diagram helps visualize the dynamic behavior of objects, ensuring that the system responds appropriately to different events or changes in state throughout hospital operations



Deployment diagram:



Component diagram:

A component diagram for a Hospital Management System depicts the high-level structure of the system by showing the main components and their relationships. It includes components such as the User Interface, Backend Server, Database, Payment Gateway, and Notification Service. This diagram helps to understand how each part of the system interacts with others, ensuring smooth data ensuring smooth data flow and integration between modules like patientmanagement, billing, and appointment scheduling... ensuring smooth data ensuring smooth data flow and integration between modules like patientmanagement, billing, and appointment scheduling...