B.M.S. COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU

OOMD Mini Project Report

HOSPITAL MANAGEMENT SYSTEM

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

HARIKA N (1BM21CS071)
HARSHALA RANI (1BM21CS074)
IBRAHIM IFTEKHAR KHAN (1BM21CS076)
JEEVANTHI KASHYAP (1BM21CS080)

Department of Computer Science and Engineering
B.M.S. College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
2023-2024

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, **HARIKA N (1BM21CS071), HARSHALA RANI (1BM21CS074), IBRAHIM IFTEKHAR KHAN (1BM21CS076), JEEVANTHI KASHYAP (1BM21CS080)** students of 6th Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this OOMD Mini Project entitled "**HOSPITAL MANAGEMENT SYSTEM** " has been carried out in Department of CSE, B.M.S. College of Engineering, Bangalore during the academic semester March - July 2024. I also declare that to the best of our knowledge and belief, the OOMD mini Project report is not from part of any other report by any other students.

Signature of the Candidate

HARIKA N (1BM21CS071)
HARSHALA RANI (1BM21CS074)
IBRAHIM IFTEKHAR KHAN (1BM21CS076)
JEEVANTHI KASHYAP (1BM21CS080)

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the OOMD Mini Project titled "**HOSPITAL MANAGEMENT SYSTEM"** has been carried out by Harika N (1BM21CS071), Harshala Rani (1BM21CS074), Ibrahim Iftekhar Khan (1BM21CS076), Jeevanthi Kashyap (1BM21CS080) during the academic year 2023-2024.

Signature of the Faculty in Charge

Table of Contents

Chapter 1: Problem Statement

Many hospitals rely on manual or outdated systems that lead to inefficiencies, errors, and delays in managing patient care and hospital operations. This project seeks to develop an integrated Hospital Management System to automate and streamline processes such as patient registration, appointment scheduling, medical records management, billing, inventory control, and staff management. By addressing these issues, the HMS will enhance operational efficiency, reduce errors, and improve patient care, ensuring a seamless and efficient hospital management experience.

Chapter 2: Software Requirement Specification

1. Introduction

1.1 Purpose of this Document:

This document serves to define the requirements for the development of a comprehensive Hospital Management System (HMS). It outlines the system's functionalities, performance requirements, and design constraints, providing a detailed guide for developers, stakeholders, and project managers. The purpose is to ensure that the system meets the needs of hospital staff, patients, and administrators while adhering to regulatory and operational standards.

1.2 Scope of this Document:

The scope of this document includes the detailed description of the HMS, focusing on automating and optimizing hospital operations such as patient registration, appointment scheduling, medical records management, billing, inventory control, and staff management. The document will cover the system's functionality, performance expectations, and design constraints. Additionally, it will provide an overview of the development costs and time required for the project's completion, ensuring that stakeholders understand the value and investment required.

1.3 Overview:

The Hospital Management System is designed to streamline hospital operations and enhance patient care. It will provide an integrated platform for managing patient information, scheduling appointments, handling billing and payments, managing inventory, and overseeing staff schedules. The system aims to improve operational efficiency, reduce errors, and enhance patient satisfaction. This document summarizes the project's objectives, features, and expected benefits.

2. General Description

The HMS will facilitate efficient hospital management by automating critical processes and providing real-time access to information. It will support various user roles, including administrators, doctors, nurses, and patients, each with specific features tailored to their needs.

Objectives:

- Improve operational efficiency and reduce administrative workload.
- Enhance patient care through timely and accurate information.
- Ensure data security and compliance with healthcare regulations.

User Characteristics:

- Administrators: Manage overall hospital operations, user access, and system settings.
- Doctors: Access and update patient records, schedule appointments, and manage treatment plans.
- Nurses: Track patient vitals, manage inventory, and assist in patient care.

- Patients: Schedule appointments, access medical records, and manage billing information.

Features and Benefits:

- Patient Registration: Quick and efficient patient onboarding process.
- Appointment Scheduling: Centralized scheduling to avoid conflicts and delays.
- Medical Records Management: Secure and accessible storage of patient medical histories.
- Billing and Payments: Accurate and timely billing processes.
- Inventory Management: Effective control of medical supplies and equipment.
- Staff Management: Efficient scheduling and communication among staff.

3. Functional Requirements

The HMS will include the following functional requirements, listed in order of priority:

- Patient Registration and Management: Capture and store patient details, generate unique patient IDs, and manage patient information.
- Appointment Scheduling: Allow patients and staff to book, reschedule, and cancel appointments.
- Medical Records Management: Create, update, and retrieve patient medical records, ensuring data integrity and security.
- Billing and Payment Processing: Generate invoices, process payments, and manage financial records.
- Inventory Management: Track inventory levels, manage orders, and alert staff of low stock.
- Staff Management: Schedule staff shifts, manage availability, and facilitate internal communication.
- Reporting and Analytics: Generate reports on various aspects of hospital operations for analysis and decision-making.

4. Interface Requirements

The HMS will interface with several other systems and provide the following interfaces:

- User Interface: Intuitive, user-friendly web and mobile interfaces for different user roles.
- Database Interface: Efficient data storage and retrieval mechanisms to manage hospital data.
- External Systems Interface: APIs for integration with external systems such as insurance providers, laboratory systems, and government health databases.
- Communication Interface: Secure messaging and notification system for communication between users.

5. Performance Requirements

The HMS must meet the following performance criteria:

- Response Time: The system should respond to user actions within 2 seconds.
- Availability: The system should have an uptime of 99.9%, ensuring high availability.
- Scalability: The system should handle up to 10,000 concurrent users without performance degradation.
- Error Rate: The maximum allowable error rate should be less than 0.01%.

6. Design Constraints

The design of the HMS will adhere to the following constraints:

- Regulatory Compliance: The system must comply with healthcare regulations such as HIPAA for data privacy and security.
- Technology Stack: Use of specific technologies (e.g., MySQL for database, React for frontend,and Node.js for backend).
- Hardware Limitations: The system must be deployable on standard server configurations without requiring specialized hardware.
- Data Security: Implementation of robust security measures to protect patient data.

7. Non-Functional Attributes

The HMS must exhibit the following non-functional attributes:

- Security: Implement strong authentication, authorization, and encryption mechanisms.
- Portability: The system should be compatible with various operating systems and devices.
- Reliability: Ensure consistent performance and error recovery mechanisms.
- Reusability: Components of the system should be reusable across different modules.
- Compatibility: Ensure compatibility with existing hospital systems and external applications.
- Data Integrity: Maintain data accuracy and consistency across the system.
- Scalability: The system should scale horizontally to accommodate growing user demands.
-

8. Preliminary Schedule and Budget

Preliminary Schedule:

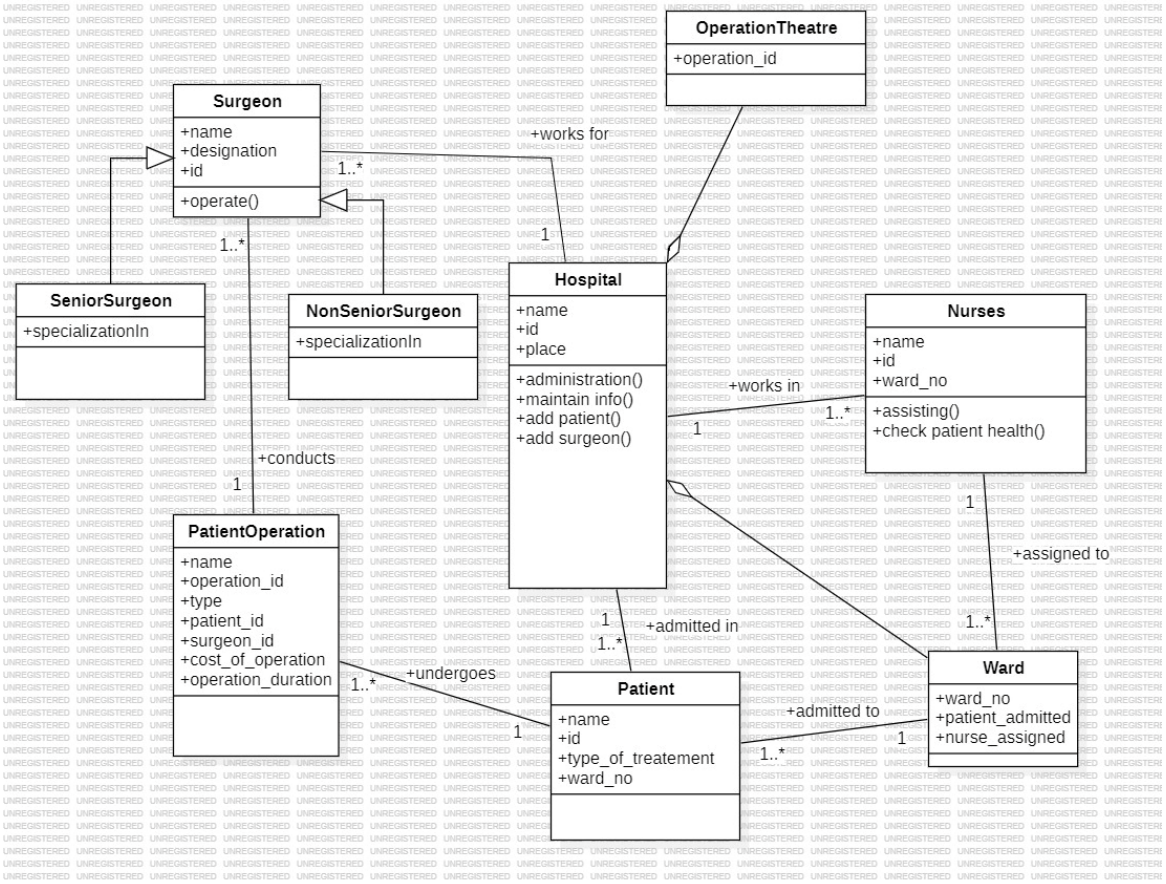
- Requirement Analysis: 4 weeks
- System Design: 6 weeks
- Development: 12 weeks
- Testing: 6 weeks
- Deployment: 2 weeks
- Total Duration: 30 weeks

Budget Estimate:

- Personnel Costs: $150,000
- Software and Licensing: $20,000
- Hardware Costs: $30,000
- Miscellaneous Expenses: $10,000
- Total Budget: $210,000

Chapter 3: Class Modeling

CLASS DIAGRAM:

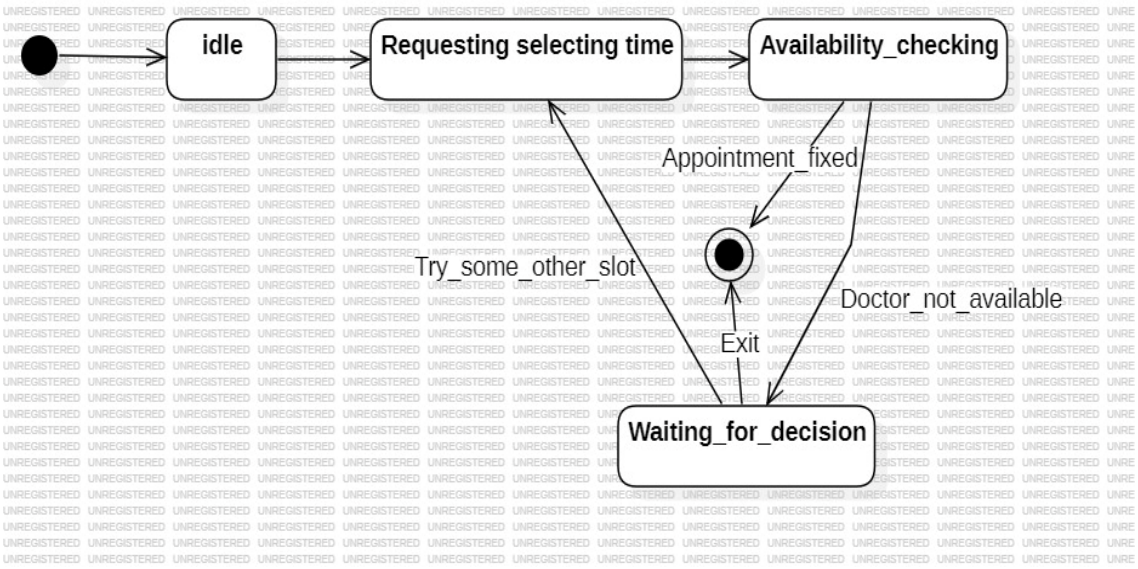


Explanation: The class diagram provides a comprehensive overview of the hospital management system's structure by detailing various classes, their attributes, methods, and the relationships among them. The primary classes include Surgeon, Senior Surgeon, Non-Senior Surgeon, Patient Operation, Hospital, Operation Theatre, Nurses, Ward, and Patient. Each class encapsulates specific attributes and methods relevant to its role. For instance, the Surgeon class contains attributes such as name, designation, and id, and a method named operate(). It is further divided into SeniorSurgeon and NonSeniorSurgeon, each having a specialization attribute. The Hospital class acts as a central entity, linking to multiple other classes and managing administrative functions, patient admissions, and surgeon assignments. Relationships between these classes are crucial for representing the real-world interactions within a hospital environment. For example, a surgeon can conduct multiple patient operations, and a patient can be admitted to a ward and assigned nurses who assist in their care. The 1-to-many and 1-to-1 relationships between these classes are essential for modeling the complexities of hospital operations. Overall, the class diagram effectively captures the hierarchical and functional relationships between different hospital entities, providing a clear blueprint of the hospital management system's architecture.

Chapter 4: State Modeling

STATE DIAGRAM:

1. State-chart diagram for Patient-Appointment:

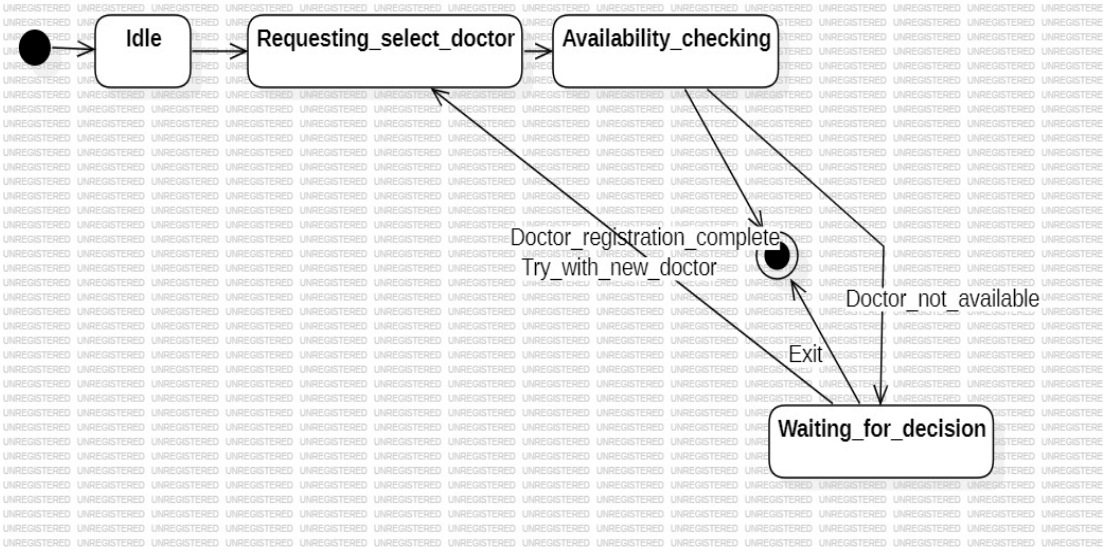


Explanation:

The diagram represents a state machine for a patient appointment scheduling system, beginning with the "Idle" state, where the system awaits the patient's action to start the scheduling process. Once the patient decides to schedule an appointment, the system transitions to the "Requesting Selecting Time" state, prompting the patient to choose a preferred time slot. After the selection, the system moves to the "Availability Checking" state to verify the doctor's availability. If the doctor is available, the system finalizes the process in the "Appointment Fixed" state. If not, it transitions to the "Waiting for Decision" state, where the patient must decide whether to try another time slot or exit the process. The process continues based on the patient's choice, either looping back to time selection or exiting the scheduling system.

The events in the diagram are crucial for the transitions between states. The event from "Idle" to "Requesting Selecting Time" occurs when the patient initiates the scheduling process. Transitioning from "Requesting Selecting Time" to "Availability Checking" happens when the patient selects a time slot, prompting the system to check its availability. If the selected slot is unavailable, the event leads to "Waiting for Decision," where the patient must decide the next step. If the patient chooses to try another time, the system loops back to "Requesting Selecting Time." If the patient exits, the event leads the system to the "Exit" state, ending the process. An "Appointment Fixed" event occurs if the selected time is available, confirming the booking and completing the scheduling process. These events ensure a systematic flow, handling both successful appointments and scenarios requiring further action from the patient.

2. State-chart diagram for Doctor-Selection:

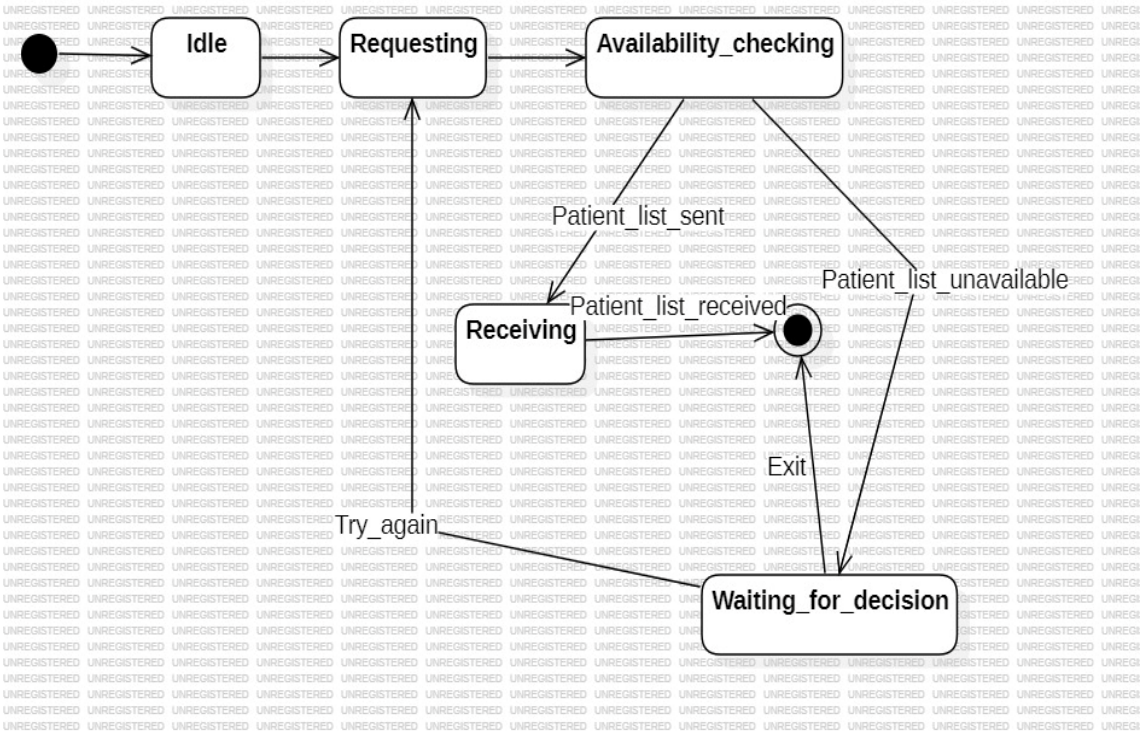


Explanation:

The diagram represents a state machine for a system where a patient is selecting and scheduling an appointment with a doctor. The system starts in the "Idle" state, waiting for the patient to begin the process. Once the patient initiates the process, the system transitions to the "Requesting Select Doctor" state, where the patient is prompted to choose a doctor. After the doctor selection, the system moves to the "Availability Checking" state to verify if the selected doctor is available. If the doctor is available, the process concludes with a "Doctor Registration Complete" state, confirming the registration. If the doctor is not available, the system transitions to the "Waiting for Decision" state, where the patient decides whether to try selecting a new doctor or exit the process. Based on the patient's decision, the system either loops back to "Requesting Select Doctor" to choose another doctor or exits the process.

The events drive the transitions between these states. The transition from "Idle" to "Requesting Select Doctor" occurs when the patient starts the doctor selection process. Once a doctor is selected, the system transitions from "Requesting Select Doctor" to "Availability Checking" to verify the doctor's availability. If the doctor is not available, the system transitions to "Waiting for Decision," where the patient must decide the next action. The patient can choose to "Try with New Doctor," looping back to the "Requesting Select Doctor" state to make another selection. If the patient decides to "Exit," the system transitions to the "Exit" state, ending the process. If the selected doctor is available, the event "Doctor Registration Complete" confirms the booking and completes the process. These events ensure a structured flow, handling both successful doctor registrations and scenarios where further action is required by the patient.

3. State-chart diagram for Patients list:



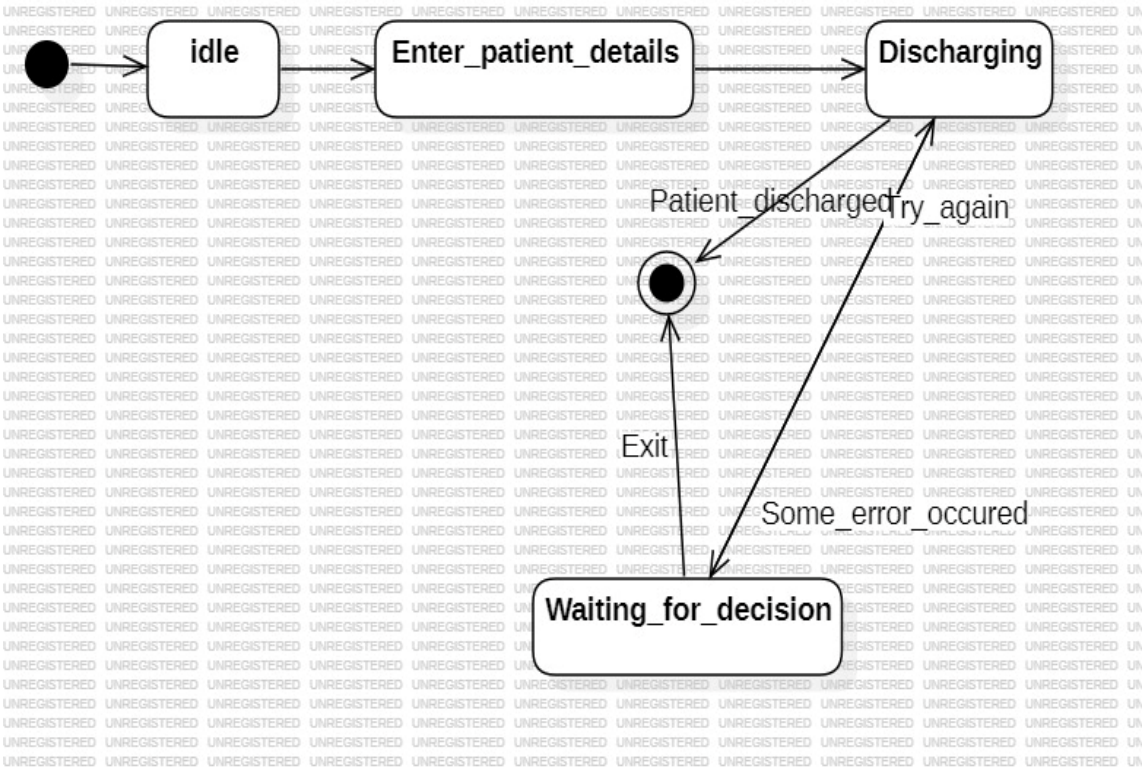
Explanation:

The diagram represents a state machine for a system that handles patient list requests and availability checks. The system starts in the "Idle" state, where it waits for the patient to initiate the process. When the patient starts a request, the system transitions to the "Requesting" state. Here, the system waits for the patient's request for a list. The system then moves to the "Availability Checking" state, where it checks the availability of the patient list. If the patient list is available, the system sends the list, transitioning to the "Receiving" state. In this state, the system confirms the patient list has been received.

If the patient list is unavailable, the system transitions to the "Waiting for Decision" state, where the patient must decide whether to try again or exit the process. Based on the patient's decision, the system either loops back to "Requesting" to try again or exits the process. If the list is successfully received, the process ends with the confirmation.

The events that drive the transitions between these states are crucial. The transition from "Idle" to "Requesting" occurs when the patient initiates the request process. From "Requesting" to "Availability Checking," the system moves to verify the availability of the requested list. If the list is available, the event "Patient List Sent" transitions the system to "Receiving," where the list is confirmed to be received. If the list is unavailable, the event "Patient List Unavailable" moves the system to "Waiting for Decision." Here, the patient decides the next action. If the patient opts to "Try Again," the system loops back to "Requesting" to make another attempt. If the patient decides to "Exit," the system transitions to the "Exit" state, ending the process. These events ensure a structured flow, managing both successful list retrievals and scenarios where further action is needed from the patient.

4. State-chart diagram for Discharge-patient:

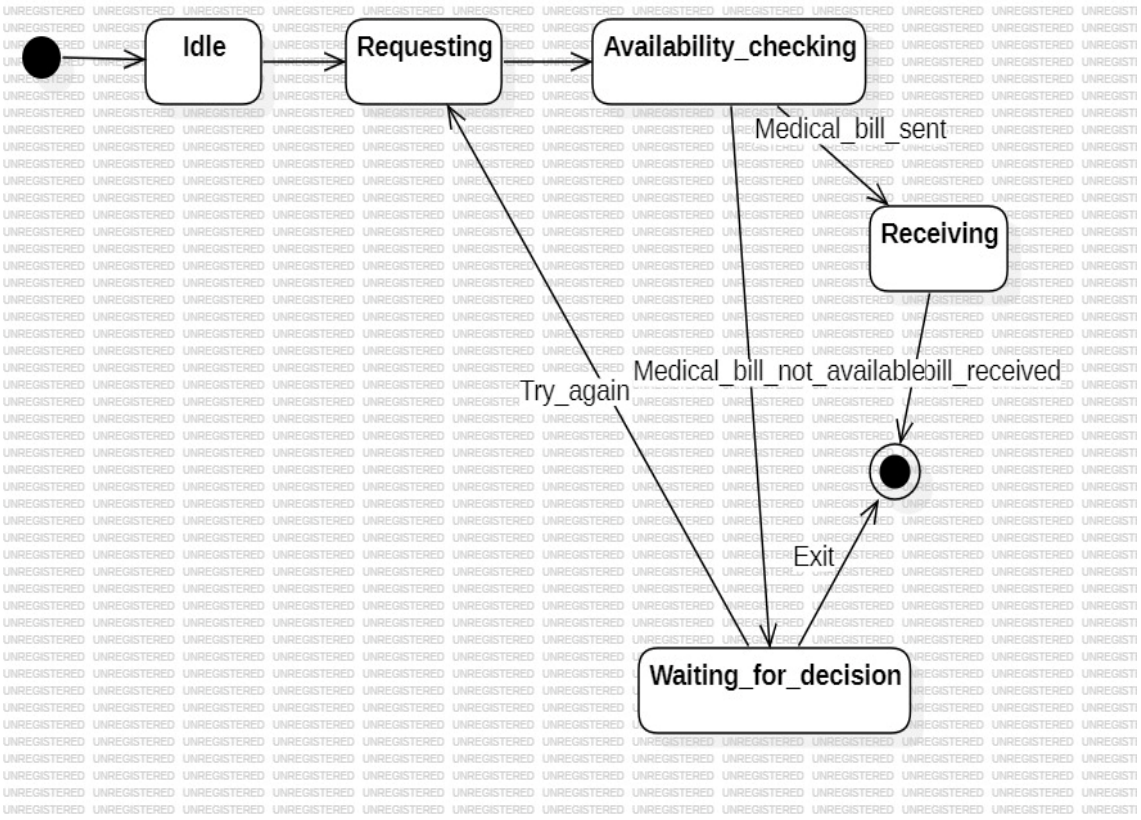


Explanation:

The "idle" state represents the initial or default state of the system, likely when it is not actively engaged in any specific task or process. The "Enter_patient_details" state suggests that the system is designed to handle patient information, possibly in a healthcare or medical context. The "Discharging" state indicates that the system is involved in the process of discharging patients, perhaps after completing their treatment or stay. The "Waiting_for_decision" state implies that the system is awaiting some decision or input, possibly related to the patient's discharge or treatment plan.

The "Patient_discharged_try_again" event suggests that the system encountered an issue or error during the discharge process, prompting a retry or revisiting of the discharge steps. The "Exit" event likely represents a successful completion of the discharge process or a decision to exit the current workflow. The "Some_error_occurred" event indicates that an error or exception occurred during the system's operation, potentially requiring intervention or a transition to an error-handling state.

5. State-chart diagram for Medical bills:

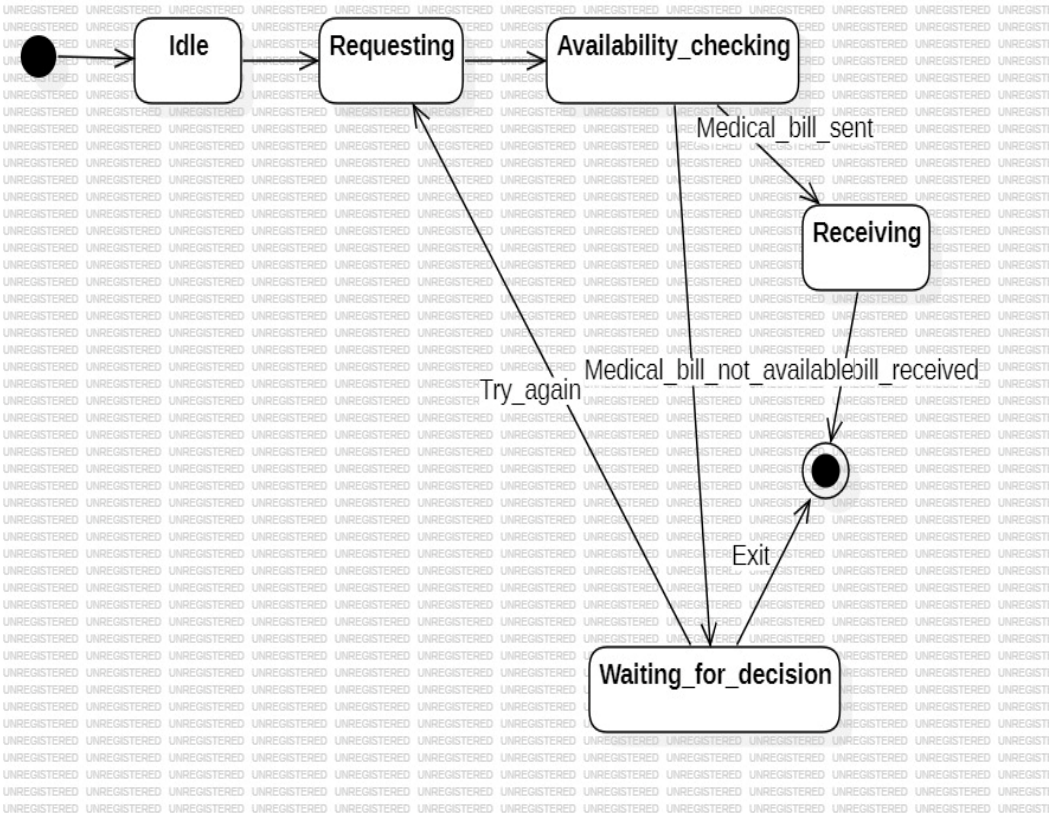


Explanation:

The "Idle" state likely represents the initial state when the system is not actively processing any medical bill or claim. The "Requesting" state suggests that a request or submission for a medical bill has been initiated. The "Availability_checking" state indicates that the system is verifying the availability or eligibility of the submitted medical bill or claim. The "Receiving" state implies that the system is receiving or processing the medical bill or payment. The "Waiting_for_decision" state suggests that the system is awaiting a decision or action, possibly related to the processing or approval of the medical bill.

The "Medical_bill_sent" event represents the submission or sending of a medical bill or claim for processing. The "Medical_bill_not_available" event indicates that the submitted medical bill or claim is not available or eligible for processing, possibly due to issues with the patient's coverage or plan. The "Bill_received" event signifies that the medical bill or payment has been successfully received and processed. The "Try_again" event suggests that the system encountered an error or issue during the processing of the medical bill, prompting a retry or resubmission. The "Exit" event likely represents the completion or termination of the medical bill processing workflow.

6. State-chart diagram for Payment:



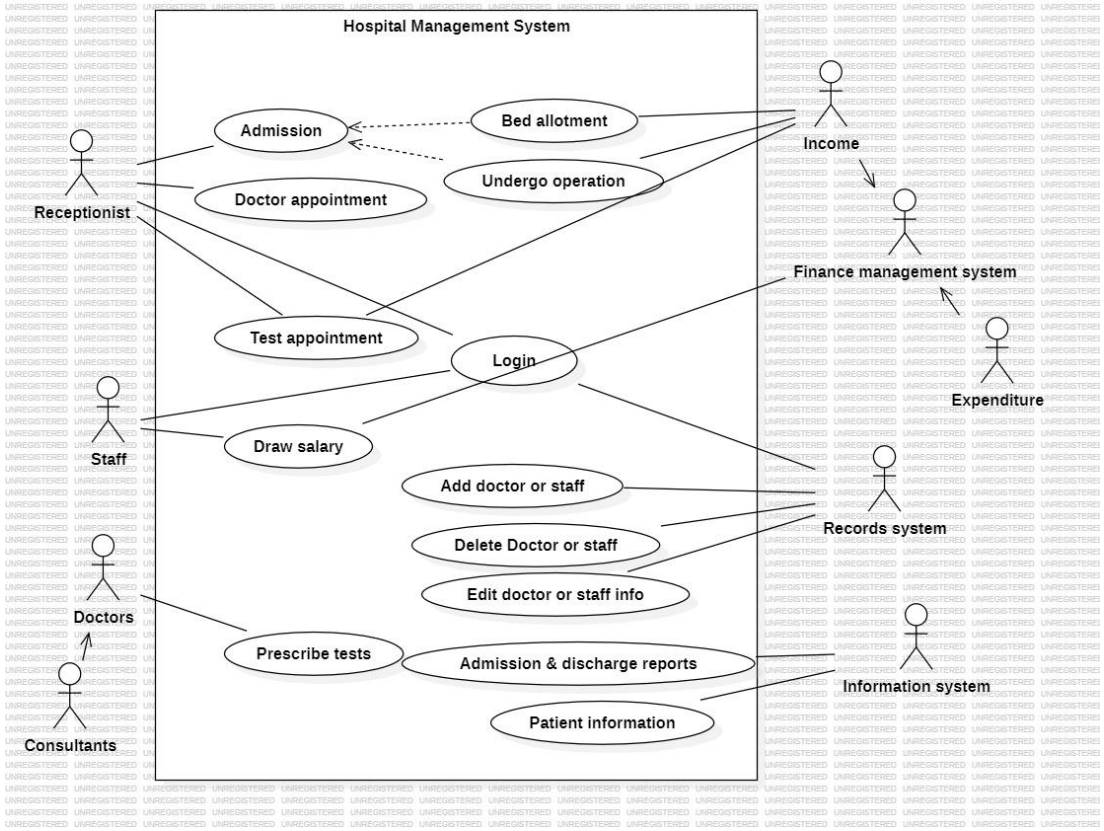
Explanation:

The "Idle" state likely represents the initial state when the system is not actively processing any payment or transaction. The "Enter_bankdetails_or_pay_cash" state suggests that the system is prompting the user or customer to provide their bank details or choose the cash payment option. The "Verifying" state indicates that the system is verifying the provided bank details or payment method. The "Receiving" state implies that the system is receiving or processing the payment. The "Waiting_for_decision" state suggests that the system is awaiting a decision or action, possibly related to the payment or transaction.

The "Payment_transaction_started" event represents the initiation of the payment or transaction process. The "Insufficient_funds" event indicates that there are not enough funds or balance available for the payment or transaction to be completed successfully. The "Payment_done" event signifies that the payment or transaction has been successfully completed and processed. The "Try_again" event suggests that the system encountered an error or issue during the payment process, prompting the user to try again or resubmit their payment details. The "Exit" event likely represents the completion or termination of the payment or transaction workflow.

Chapter 5: Interaction Modeling

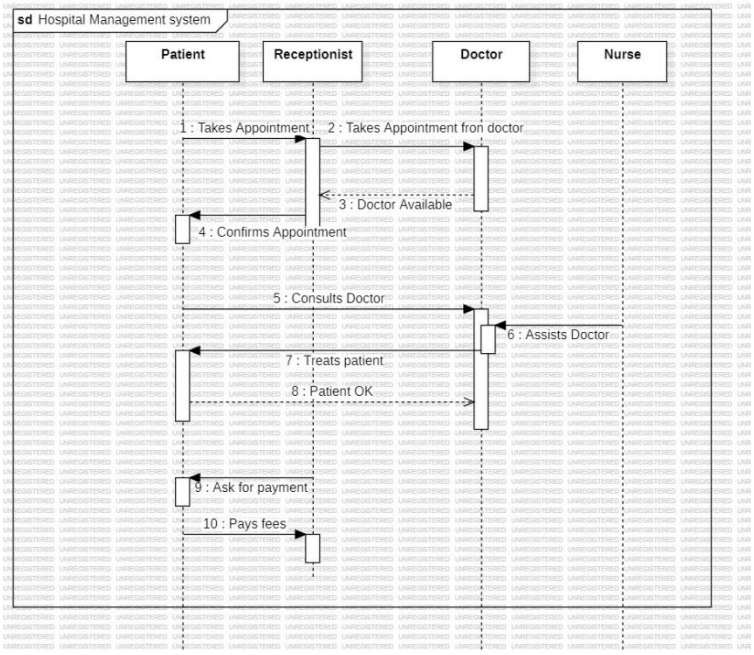
USE CASE DIAGRAM:



Explanation:The actors in the hospital management system diagram include the Receptionist, who is responsible for handling patient admissions and appointments; the Staff, who perform administrative tasks like managing personnel information and payroll; the Doctors, who examine patients, order tests, and perform medical procedures; and the Consultants, who provide specialized medical expertise. Additionally, there are integrated systems for Finance management, handling income and expenditure; Records, maintaining patient records and reports; and Information, storing and retrieving patient details.

The use cases or functionalities depicted in the diagram are as follows: Admission handles the process of admitting patients, while Bed allotment assigns available beds or rooms. Undergo operation represents patients undergoing surgical procedures, and Doctor appointment schedules consultations with doctors. Test appointment handles scheduling diagnostic tests, and Login is the authentication process for authorized access. Draw salary allows staff to access payroll information, while Add doctor or staff, Delete Doctor or staff, and Edit doctor or staff info manage personnel records. Prescribe tests enables doctors to order diagnostic tests, and Admission & discharge reports generate relevant reports. Patient information covers the storage and retrieval of patient-related details. Overall, the diagram comprehensively covers the various aspects of managing a hospital, including patient care, staff management, financial operations, and data management.

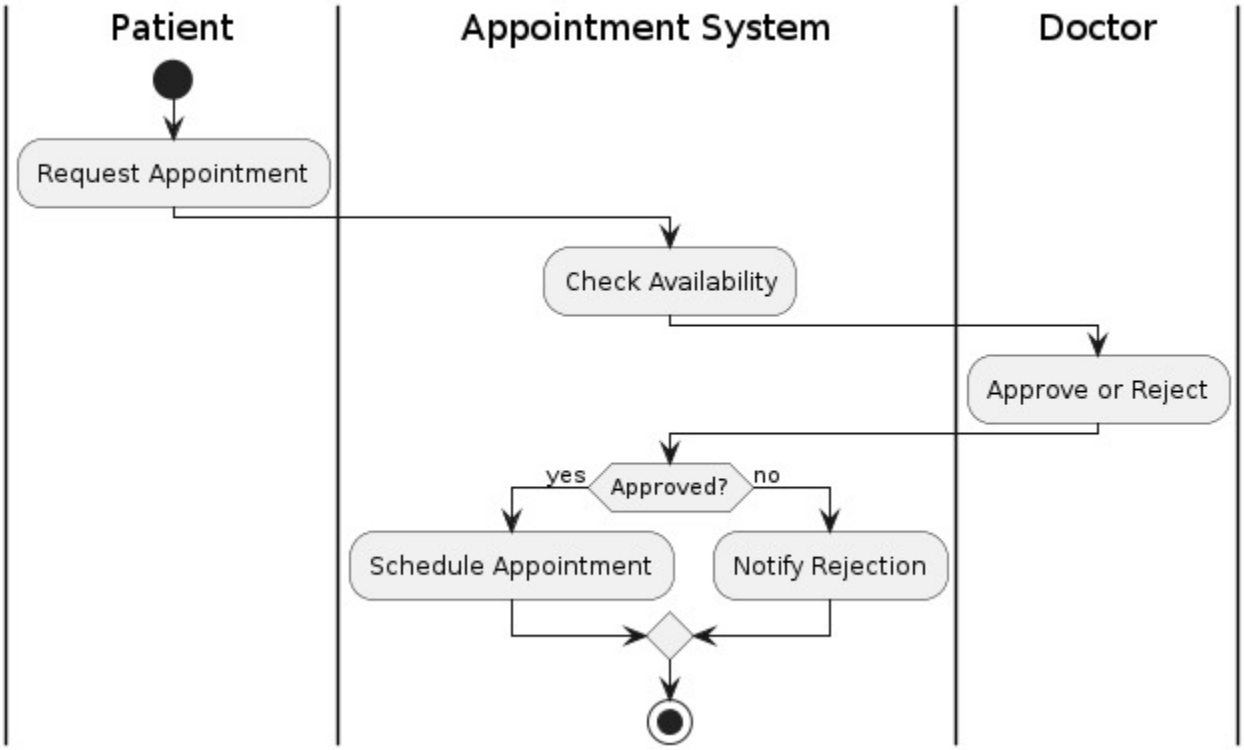
SEQUENCE DIAGRAM:



Explanation: The sequence diagram captures the dynamic interactions between various roles involved in a patient's journey through the hospital system, focusing on the appointment and treatment process. The key participants in this sequence are the Patient, Receptionist, Doctor, and Nurse. The sequence begins with the patient taking an appointment, followed by the receptionist arranging this appointment with the doctor. The doctor checks their availability, and once confirmed, the receptionist informs the patient about the appointment.

This process ensures a structured flow of communication and scheduling within the hospital system. The diagram effectively highlights the synchronous and asynchronous messages exchanged between these entities to achieve a smooth appointment booking process. By delineating the step-by-step interaction, it provides a clear visualization of the responsibilities and actions taken by each role, ensuring that the system functions efficiently and that patients receive timely care. The sequence diagram meticulously details each interaction, ensuring clarity in the procedural flow and roles involved. It encapsulates the essential operations within the hospital management system, from initial appointment scheduling to final billing, thereby ensuring all actions are coordinated and well-documented for effective patient management.

ACTIVITY DIAGRAM:



Explanation: The activity diagram outlines the workflow of a patient's visit to the hospital, starting from taking an appointment to the eventual discharge, if necessary. Initially, the patient takes an appointment, and the receptionist checks for the availability of a doctor. If no doctor is available, the process terminates with a notification to the patient. If a doctor is available, the patient consults the doctor, who then determines whether surgery is needed. This decision point is crucial as it dictates the subsequent flow of activities. If surgery is not required, the patient continues with standard treatment. If surgery is needed, the patient undergoes the operation, conducted by a surgeon with assistance from a nurse. The diagram provides a clear visual representation of these decision points and activities, ensuring each step in the patient's journey is meticulously planned and documented.

Post-operation, the patient's condition is monitored to determine if they are okay. If the patient is not okay, further medical interventions are carried out. If the patient is stable, they are discharged from the hospital. The diagram effectively captures these critical junctures and the potential paths a patient's treatment may follow, providing a comprehensive view of the patient care process. Each activity is represented as a distinct step, ensuring clarity in the sequence of actions and decision-making process. This detailed flowchart helps in understanding the various pathways within the hospital management system, ensuring that medical staff can follow a standardized procedure for patient care, thereby enhancing efficiency and patient satisfaction. The activity diagram serves as a vital tool for visualizing and optimizing the workflow within the hospital, ensuring that all possible scenarios are accounted for and managed effectively.

References

https://www.geeksforgeeks.org/unified-modeling-language-uml-state-diagrams/

https://www.krishnagudi.com/wp-content/uploads/2016/07/chapter-5.pdf

https://support.ptc.com/help/modeler/r9.0/en/index.html#page/Integrity_Modeler/rtsme/class_modeling_overview.html

https://www.ibm.com/docs/en/dma?topic=diagrams-classes