



A **sequence diagram** is a UML behavioral diagram that shows **how objects and actors interact in a specific order over time** to accomplish a task.

In a **Hospital Management System**, it is used to model real hospital workflows such as:

- Patient registration
- Appointment booking
- Consultation
- Billing and payment

2. Selected Scenario (Start → Finish)

Scenario Chosen: Patient Registration, Appointment Booking, Consultation, and Billing

This scenario covers the **entire hospital visit lifecycle**, making it ideal for demonstrating a complete sequence.

3. Actors and System Objects / Components

Actors

1. **Patient** – Initiates hospital services
2. **Receptionist** – Handles registration and appointment booking
3. **Doctor** – Conducts consultation and updates medical records

System Objects / Components

4. **HMS UI** – Interface used by staff
5. **HMS Controller** – Business logic handler
6. **Patient Database** – Stores patient details
7. **Appointment Database** – Stores appointment schedules
8. **Billing System** – Processes payments
9. **Notification Service** – Sends SMS/email confirmations

Each of these will appear as **vertical lifelines** in the sequence diagram.

4. Explanation of Lifelines and Activation Bars

- **Lifeline:** Vertical dashed line under each actor/object showing its existence during the interaction.
- **Activation bar:** Thin vertical rectangle on the lifeline showing **when the object is active and executing a method**.

5. Detailed Step-by-Step Interaction Flow

Step 1: Patient Registration

1. **Patient → Receptionist**
requestRegistration()
(Synchronous message – patient waits)
2. **Receptionist → HMS UI**
enterPatientDetails()
(Synchronous)
→ Activation bar starts on HMS UI
3. **HMS UI → HMS Controller**
createPatientRecord()
(Synchronous)
→ Activation bar starts on HMS Controller

4. HMS Controller → Patient Database

savePatient(data)

(Synchronous method call)

→ Activation bar on Database

5. Patient Database → HMS Controller

confirmation()

(Return message – dashed arrow)

→ Database activation ends

6. HMS Controller → HMS UI

recordCreated()

(Return message)

7. HMS UI → Receptionist

displayConfirmation()

8. Receptionist → Patient

registrationConfirmed()

Step 2: Appointment Booking

9. Patient → Receptionist

requestAppointment()

10. Receptionist → HMS UI

checkAvailability()

11. HMS UI → HMS Controller

getDoctorSchedule()

12. HMS Controller → Appointment Database

fetchSchedule()

13. Appointment Database → HMS Controller

scheduleDetails() *(return)*

14. HMS Controller → HMS UI

availableSlots()

15. HMS UI → Receptionist

showSlots()

16. Receptionist → HMS UI

bookAppointment(slot)

17. HMS UI → HMS Controller

saveAppointment()

18. HMS Controller → Appointment Database

storeAppointment()

19. Appointment Database → HMS Controller

confirmation() (*return*)

Step 3: Asynchronous Notification

20. HMS Controller → Notification Service

sendAppointmentSMS()

(*Asynchronous message – no waiting*)

→ Notification service works independently

Step 4: Doctor Consultation

21. Doctor → HMS UI

viewPatientRecord()

22. HMS UI → HMS Controller

fetchMedicalHistory()

23. HMS Controller → Patient Database

getMedicalData()

24. Patient Database → HMS Controller

medicalHistory() (*return*)

25. Doctor → HMS UI

updateDiagnosis()

26. HMS UI → HMS Controller

saveDiagnosis()

27. HMS Controller → Patient Database

updateRecord()

Step 5: Billing and Payment (Finish)

28. Receptionist → HMS UI

generateBill()

29. HMS UI → Billing System

calculateCharges() (*Synchronous*)

30. Billing System → HMS UI

billAmount() *(return)*

31. Patient → Billing System

makePayment()

32. Billing System → HMS Controller

paymentConfirmation() *(return)*

33. HMS Controller → HMS UI

paymentSuccessful()

34. HMS UI → Patient

receiptIssued()

6. Synchronous vs Asynchronous Messages

Type	Example
Synchronous	savePatient(), fetchSchedule()
Asynchronous	sendAppointmentSMS()

Return messages confirmation(), billAmount()

7. What This Sequence Diagram Clearly Shows

- ✓ Actors and system components
- ✓ Full message flow from start to finish
- ✓ Method calls and return responses
- ✓ Activation lifelines for each object
- ✓ Realistic hospital workflow
- ✓ UML-compliant structure