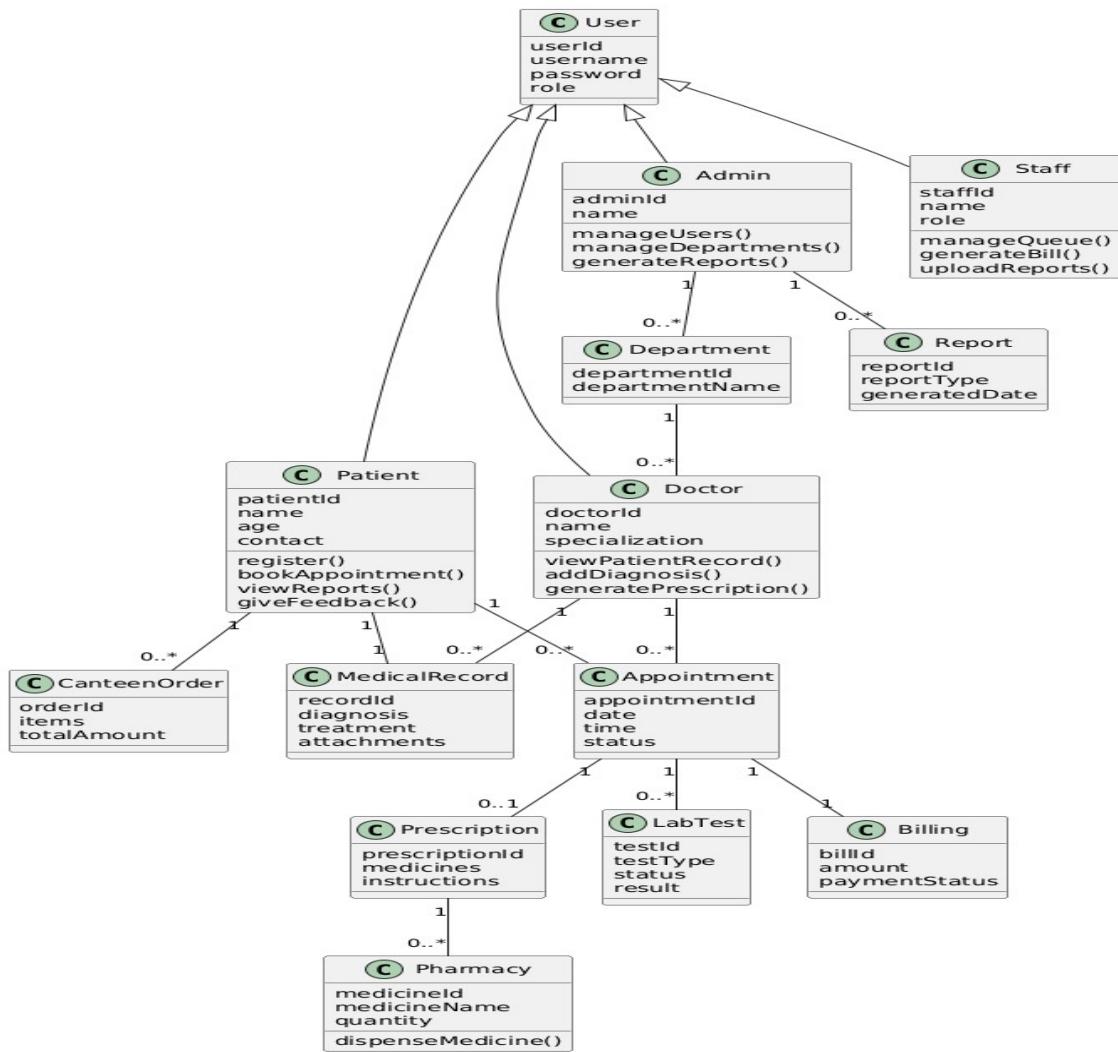


UML DIAGRAMS

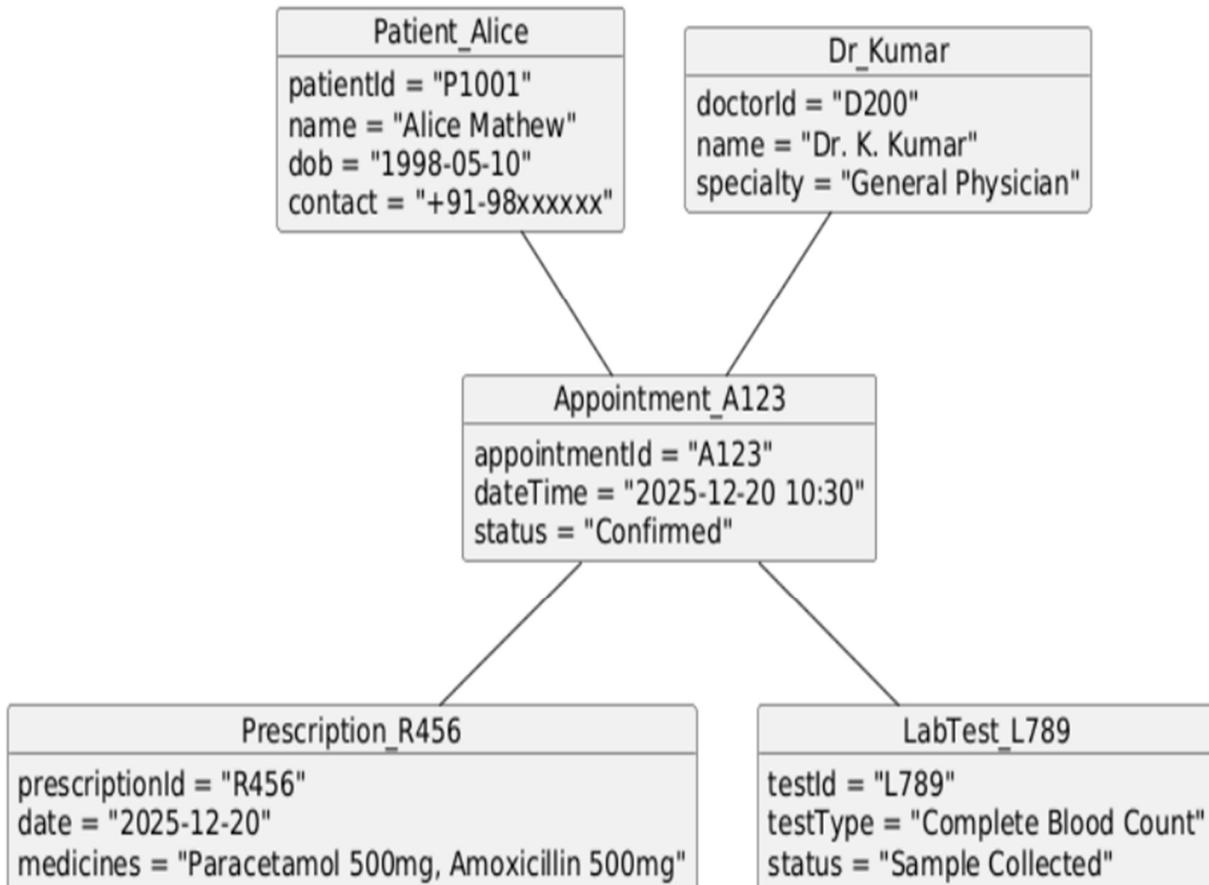
CLASS DIAGRAM

Shows the system's main classes such as User, Admin, Patient, Doctor, Staff, Appointment, Prescription, LabTest, Billing, Report, MedicalRecord, and CanteenOrder with their attributes and relationships. This diagram helps in understanding the overall structure of the HealCare system and forms the basis for database table design.



OBJECT DIAGRAM

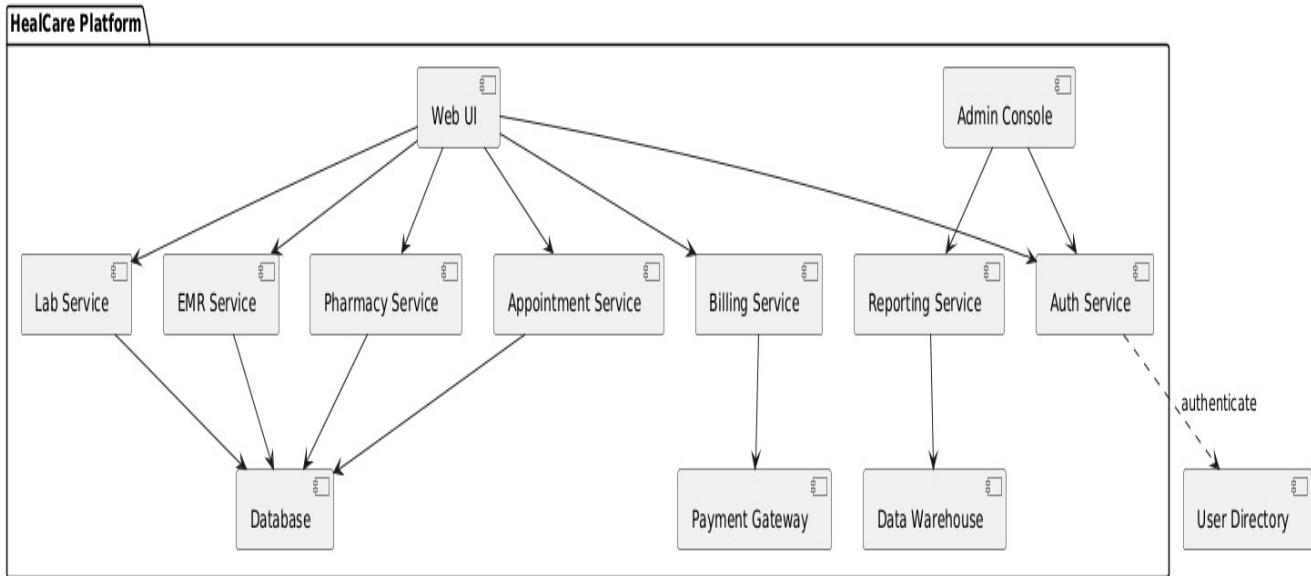
Represents a snapshot of the system at a particular time. Example objects include a Patient object booking an Appointment with a Doctor, a generated Prescription, and a LabTest report, showing how real-time instances interact in HealCare.



COMPONENT DIAGRAM

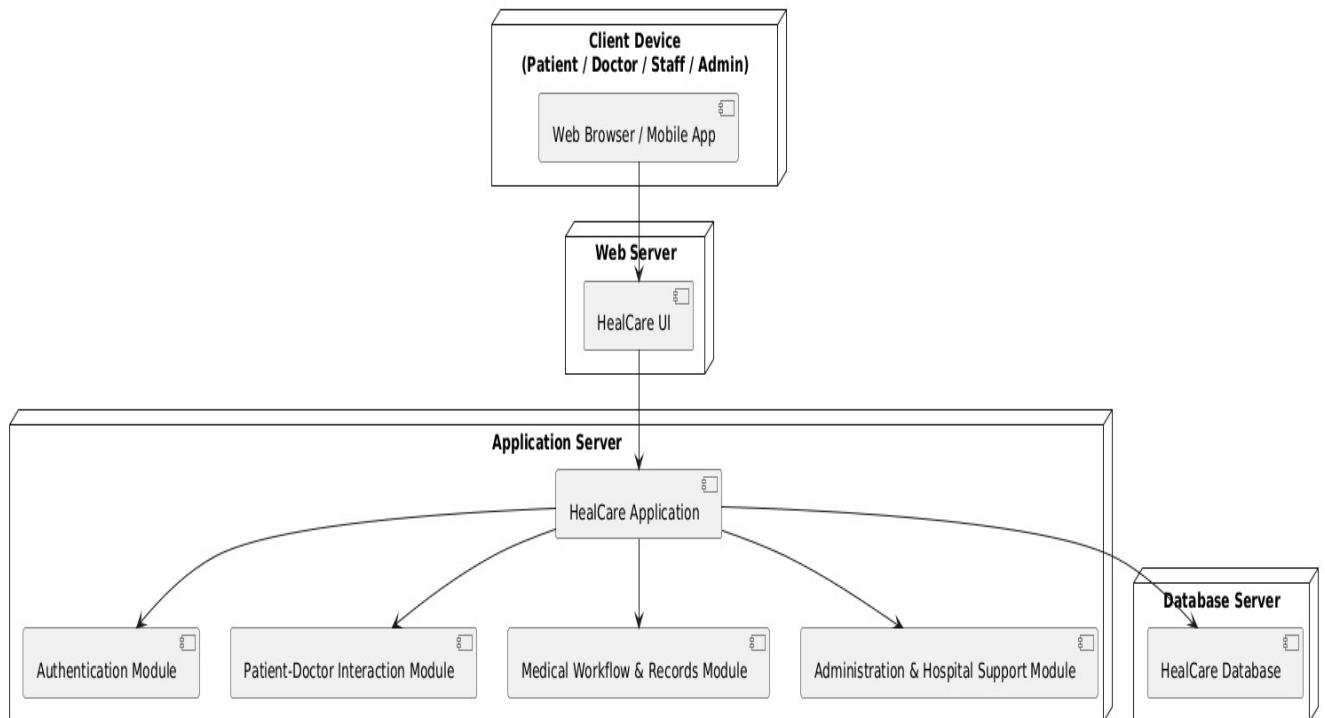
The component diagram displays the **high-level modules** of the HealCare system.

Displays high-level components such as Web Application, Authentication Service, Appointment Module, Medical Records Module, and Database. It provides a clear view of the system's modular architecture.



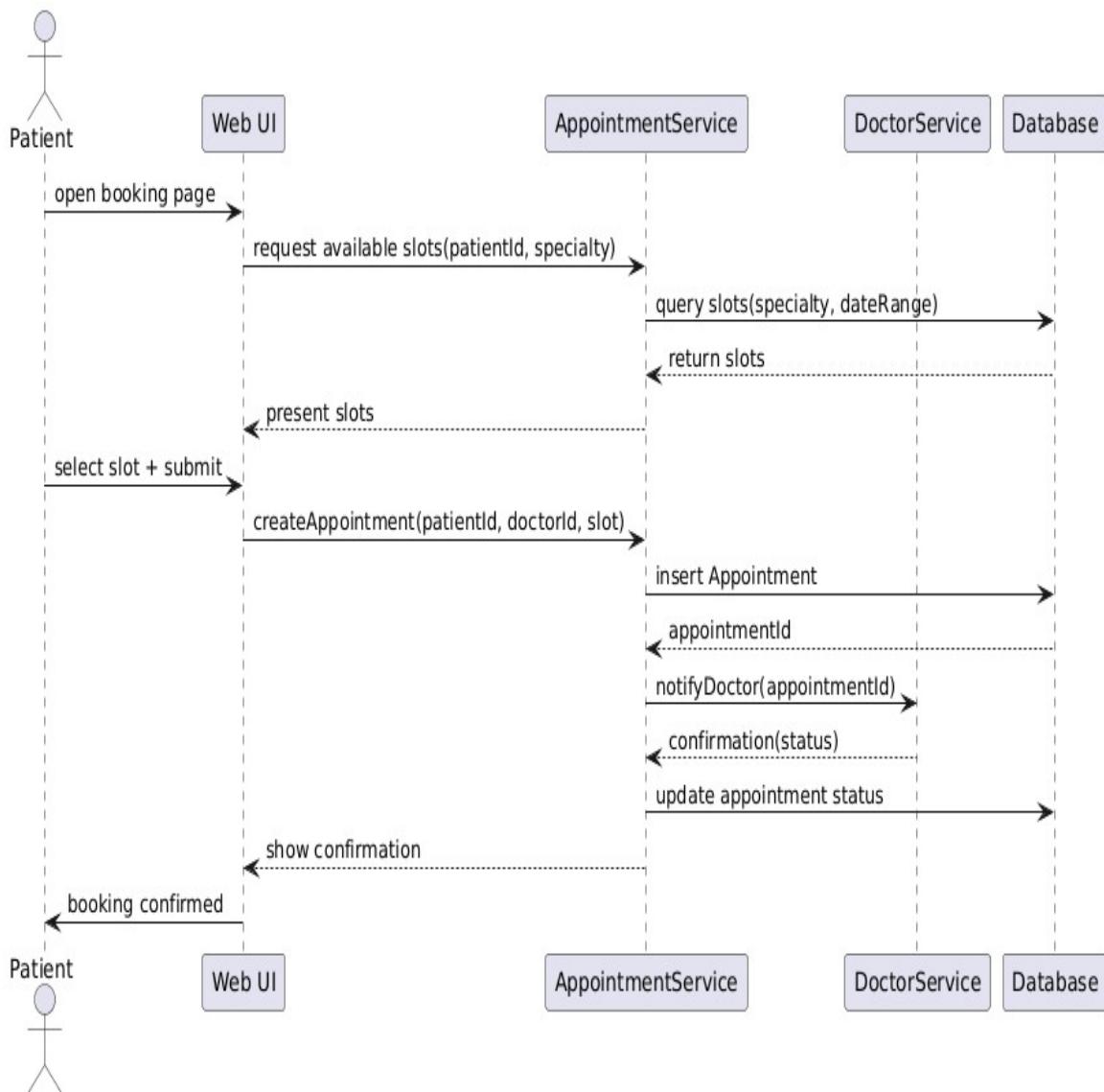
DEPLOYMENT DIAGRAM

Deployment diagram shows the **physical architecture** of the system, how the HealCare system is physically installed and accessed. Users (patients, doctors, staff, and admins) use a web or mobile interface connected to a web server, which forwards requests to the application server containing all healthcare modules.



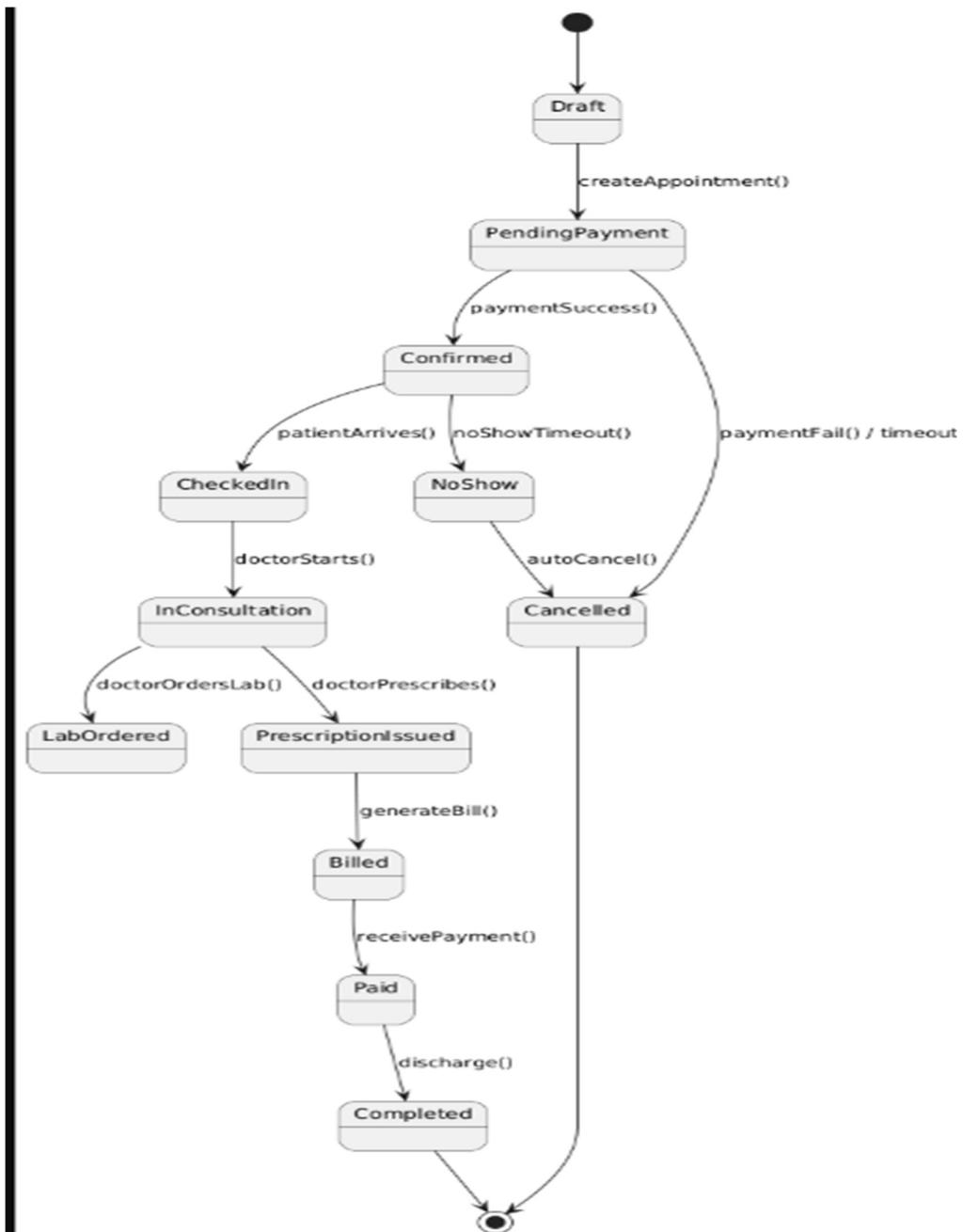
SEQUENCE DIAGRAM

The sequence diagram shows the **order of interactions over time**. Shows the interaction flow for processes like appointment booking .It explains how requests move between the patient, system, doctor, and database in a time sequence.



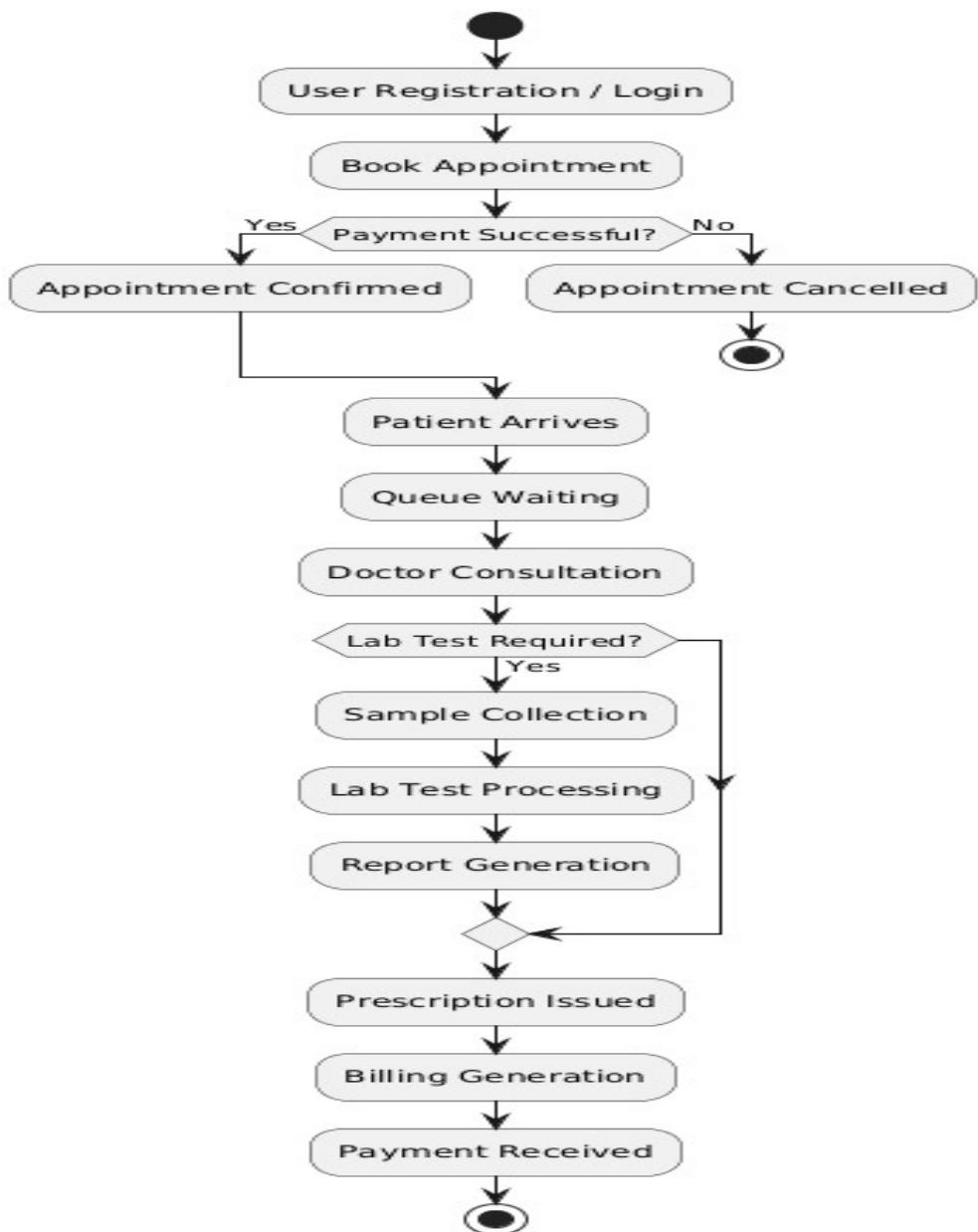
STATE CHART DIAGRAM

Shows the various states of an appointment or patient visit such as Requested, Confirmed, In Consultation, LabOrdered, Prescription Issued, Billed, Paid, Completed, and Closed, along with transitions between these states.



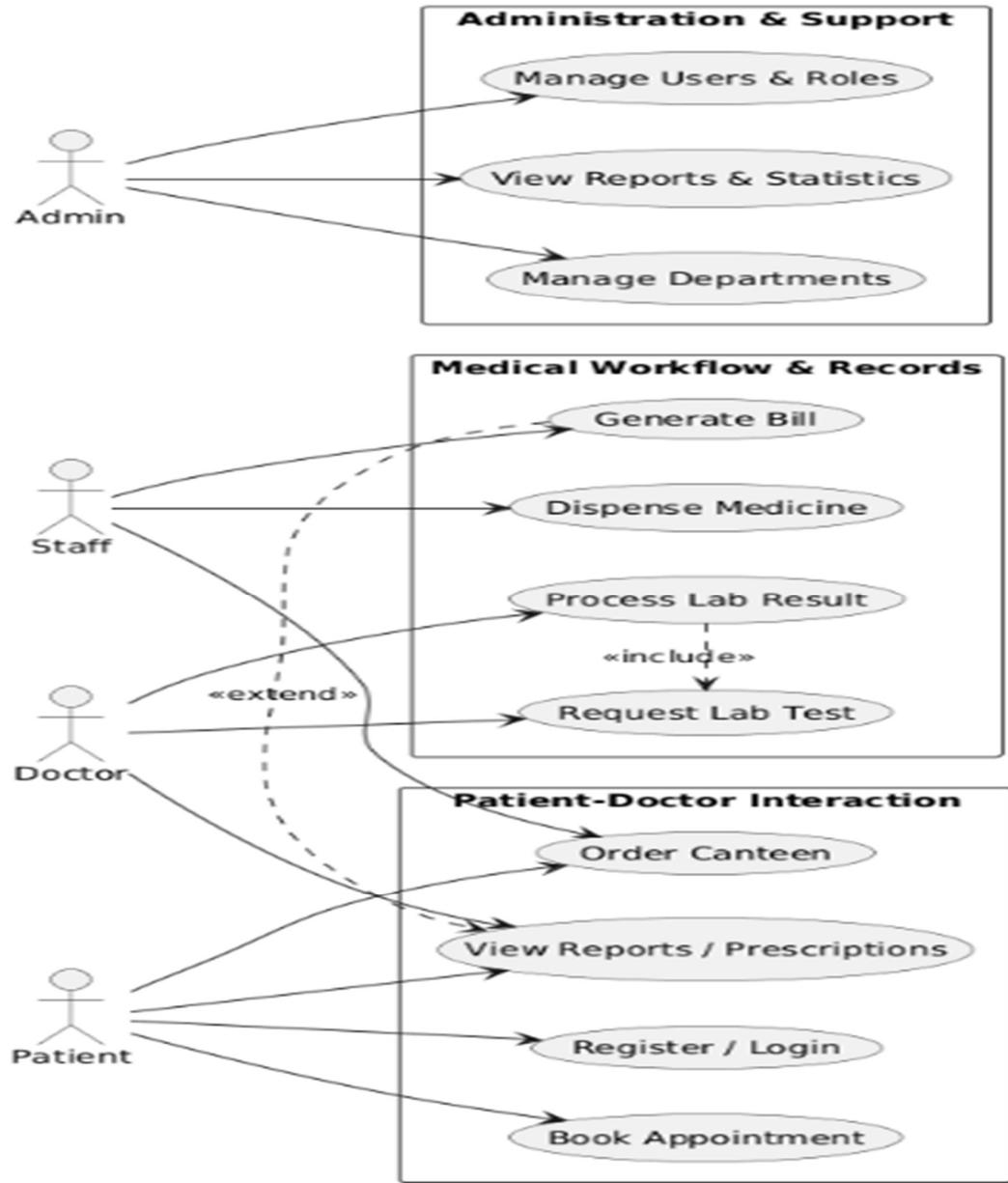
ACTIVITY DIAGRAM

Represents the workflow of patient care in HealCare. Activities include registration, appointment booking, Queue waiting, doctor consultation, lab testing or prescription, billing, and report generation.



USE CASE DIAGRAM

Illustrates the interactions between actors (Patient, Doctor, Staff, Admin) and system functionalities such as registration, appointment booking, diagnosis entry, lab test requests, billing, report generation, and system administration.



COLLABORATION DIAGRAM

Illustrates how system components collaborate to complete healthcare operations such as patient consultation. It focuses on message exchange between modules like Appointment Service, Lab Service, Medical Records, and Billing.

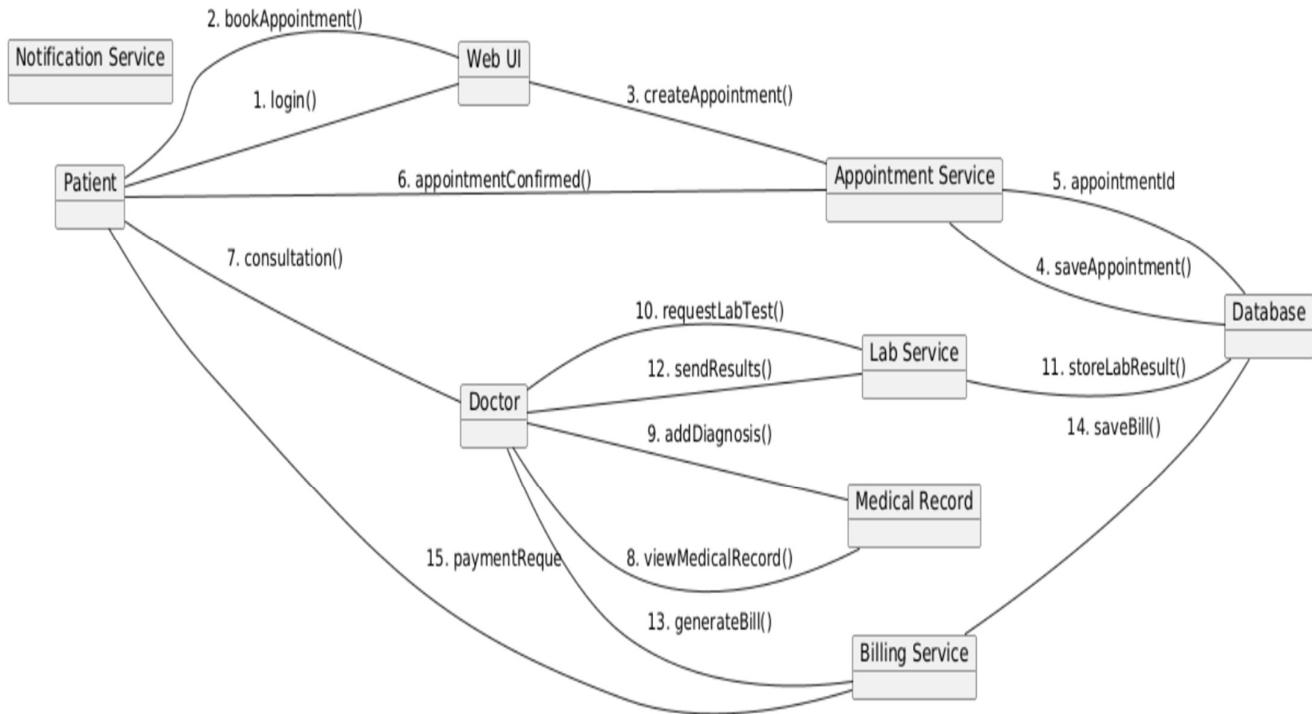


Table Design

USERS TABLE

Field	Type
user_id	INT AUTO_INCREMENT PRIMARY KEY
name	VARCHAR(100)
email	VARCHAR(100) UNIQUE
phone	VARCHAR(20)
password_hash	VARCHAR(255)
role	ENUM('Patient','Doctor','Staff','Admin')
created_at	DATETIME

Patients Table

Field Name	Data Type
patient_id (PK)	INT
user_id (FK)	INT
name	VARCHAR(100)
age	INT
gender	VARCHAR(10)
address	TEXT
blood_group	VARCHAR(5)
medical_history	TEXT

Doctors Table	
Field Name	Data Type
doctor_id (PK)	INT
user_id (FK)	INT
name	VARCHAR(100)
specialization	VARCHAR(100)
experience	INT
availability	VARCHAR(20)

Staff Table	
Field Name	Data Type
staff_id (PK)	INT
user_id (FK)	INT
name	VARCHAR(100)
department	VARCHAR(50)
role	VARCHAR(50)

MedicalRecords Table	
Field Name	Data Type
record_id (PK)	INT
patient_id (FK)	INT
doctor_id (FK)	INT
diagnosis	TEXT
treatment	TEXT
visit_date	DATE

Billing Table

Field Name	Data Type
bill_id (PK)	INT
patient_id (FK)	INT
total_amount	DECIMAL(10,2)
payment_status	VARCHAR(20)
payment_date	DATE

LabTests Table

Field Name	Data Type
abtest_id (PK)	INT
patient_id (FK)	INT
test_name	VARCHAR(100)
result	VARCHAR(100)
test_date	DATE

Appointments Table

Field Name	Data Type
appointment_id (PK)	INT
patient_id (FK)	INT
doctor_id (FK)	INT
appointment_date	DATE
time_slot	VARCHAR(20)
status	VARCHAR(30)

CanteenOrders Table

Field Name	Data Type
order_id (PK)	INT
patient_id (FK)	INT
item_name	VARCHAR(100)
quantity	INT
order_status	VARCHAR(20)

PRESCRIPTIONS TABLE

Field	Type
prescription_id	INT AUTO_INCREMENT PRIMARY KEY
patient_id	INT (FK)
doctor_id	INT (FK)
medicines	TEXT
instructions	TEXT
created_at	DATETIME

FEEDBACK TABLE

Field	Type
feedback_id	INT AUTO_INCREMENT PRIMARY KEY
patient_id	INT (FK)
doctor_id	INT (FK)
rating	INT
comments	TEXT
created_at	DATETIME

PHARMACY TABLE

Field	Type
medicine_id	INT AUTO_INCREMENT PRIMARY KEY
medicine_name	VARCHAR(100)
quantity	INT
price	DECIMAL(8,2)
expiry_date	DATE

Queue Table

Field	Data Type
queue_id	INT
patient_id	INT
department	VARCHAR
queue_number	INT
status	VARCHAR

Department Table

Field	Data Type
department_id	INT
department_name	VARCHAR
description	TEXT
status	VARCHAR

Reports Table

Field Name	Data Type
report_id (PK)	INT
patient_id (FK)	INT
report_type	VARCHAR(50)
generated_date	DATE

TABLE NORMALIZATION (1NF → 2NF → 3NF)

UNNORMALIZED FORM (UNF)

A single table containing patient details, doctor details, appointments, prescriptions, lab reports, billing, and feedback.

Problems:

- Data redundancy
- Repeating groups
- Update and deletion anomalies

FIRST NORMAL FORM (1NF)

- All attributes contain atomic values
- Repeating groups removed
- Separate tables created for Users, Appointments, Medical Records, Lab Tests, Billing, and Feedback

SECOND NORMAL FORM (2NF)

- Partial dependencies removed
- Patient details moved to PATIENTS table
- Doctor details moved to DOCTORS table
- Appointment table contains only appointment-related attributes

THIRD NORMAL FORM (3NF)

- Transitive dependencies removed
- User authentication details stored only in USERS table
- All non-key attributes depend only on the primary key

FINAL TABLE DESIGN (Normalized to 3NF – Attributes with Types)

USERS TABLE	
Field	Type
user_id	INT AUTO_INCREMENT PRIMARY KEY
name	VARCHAR(100)
email	VARCHAR(100) UNIQUE
phone	VARCHAR(20)
password_hash	VARCHAR(255)
role	ENUM('Patient','Doctor','Staff','Admin')
created_at	DATETIME

Department Table	
Field	Data Type
department_id	INT
department_name	VARCHAR
description	TEXT
status	VARCHAR

Patients Table

Field Name	Data Type
patient_id (PK)	INT
user_id (FK)	INT
name	VARCHAR(100)
age	INT
gender	VARCHAR(10)
address	TEXT
blood_group	VARCHAR(5)
medical_history	TEXT

Doctors Table

Field Name	Data Type
doctor_id (PK)	INT
user_id (FK)	INT
name	VARCHAR(100)
specialization	VARCHAR(100)
experience	INT
availability	VARCHAR(20)

Staff Table

Field Name	Data Type
staff_id (PK)	INT
user_id (FK)	INT
name	VARCHAR(100)
department	VARCHAR(50)
role	VARCHAR(50)

MedicalRecords Table

Field Name	Data Type
record_id (PK)	INT
patient_id (FK)	INT
doctor_id (FK)	INT
diagnosis	TEXT
treatment	TEXT
visit_date	DATE

Reports Table

Field Name	Data Type
report_id (PK)	INT
patient_id (FK)	INT
report_type	VARCHAR(50)
generated_date	DATE

LabTests Table

Field Name	Data Type
abtest_id (PK)	INT
patient_id (FK)	INT
test_name	VARCHAR(100)
result	VARCHAR(100)
test_date	DATE

Appointments Table

Field Name	Data Type
appointment_id (PK)	INT
patient_id (FK)	INT
doctor_id (FK)	INT
appointment_date	DATE
time_slot	VARCHAR(20)
status	VARCHAR(30)

Queue Table

Field	Data Type
queue_id	INT
patient_id	INT
department	VARCHAR
queue_number	INT
status	VARCHAR

Billing Table

Field Name	Data Type
bill_id (PK)	INT
patient_id (FK)	INT
total_amount	DECIMAL(10,2)
payment_status	VARCHAR(20)
payment_date	DATE

CanteenOrders Table

Field Name	Data Type
order_id (PK)	INT
patient_id (FK)	INT
item_name	VARCHAR(100)
quantity	INT
order_status	VARCHAR(20)

FEEDBACK TABLE	
Field	Type
feedback_id	INT AUTO_INCREMENT PRIMARY KEY
patient_id	INT (FK)
doctor_id	INT (FK)
rating	INT
comments	TEXT
created_at	DATETIME

PHARMACY TABLE

Field	Type
medicine_id	INT AUTO_INCREMENT PRIMARY KEY
medicine_name	VARCHAR(100)
quantity	INT
price	DECIMAL(8,2)
expiry_date	DATE

PREScriptions TABLE

Field	Type
prescription_id	INT AUTO_INCREMENT PRIMARY KEY
patient_id	INT (FK)
doctor_id	INT (FK)
medicines	TEXT
instructions	TEXT
created_at	DATETIME