Introduction of Design Document

Purpose of the Document

The purpose of this design document is to provide a comprehensive blueprint for the development of the **Hospital Management System (HMS)**. It outlines the system's overall design, architecture, components, data flow, and interaction patterns. This document serves as a guide for developers, testers, system architects, and stakeholders to ensure the system is implemented according to the functional and non-functional requirements.

Scope of the System

The Hospital Management System is intended to streamline and digitize hospital operations, including patient registration, appointment scheduling, staff and doctor management, billing, inventory control, electronic health records (EHR), and laboratory test handling. The system is designed to support role-based access, ensuring that users such as doctors, nurses, receptionists, and administrative staff can access the appropriate modules relevant to their roles.

Intended Audience

This document is intended for a variety of stakeholders involved in the development and implementation of the HMS. This includes:

- **Software Developers** – to guide implementation according to architectural decisions.
- **Test Engineers** – to verify compliance with design and functional expectations.
- **System Architects** – to review consistency, scalability, and modularity.
- **Project Managers & Clients** – to understand the system structure and design roadmap.

Overview of the Document

This design document includes diagrams, models, and technical specifications of the HMS. It begins with the **Entity Relationship Diagram (ERD)** to depict the data model, followed by **Sequence Diagrams** to illustrate use-case interactions. The document then covers system **Architecture**, **Class Diagrams**, and **Database Design** to provide in-depth technical structure. Later sections address **Interface Design**, real-world **Test Cases**, and a **Glossary** to clarify key terms. The design ensures that the HMS is reliable, secure, maintainable, and scalable for hospitals of various sizes.

Sequence Diagrams

Definition of Sequence Diagram

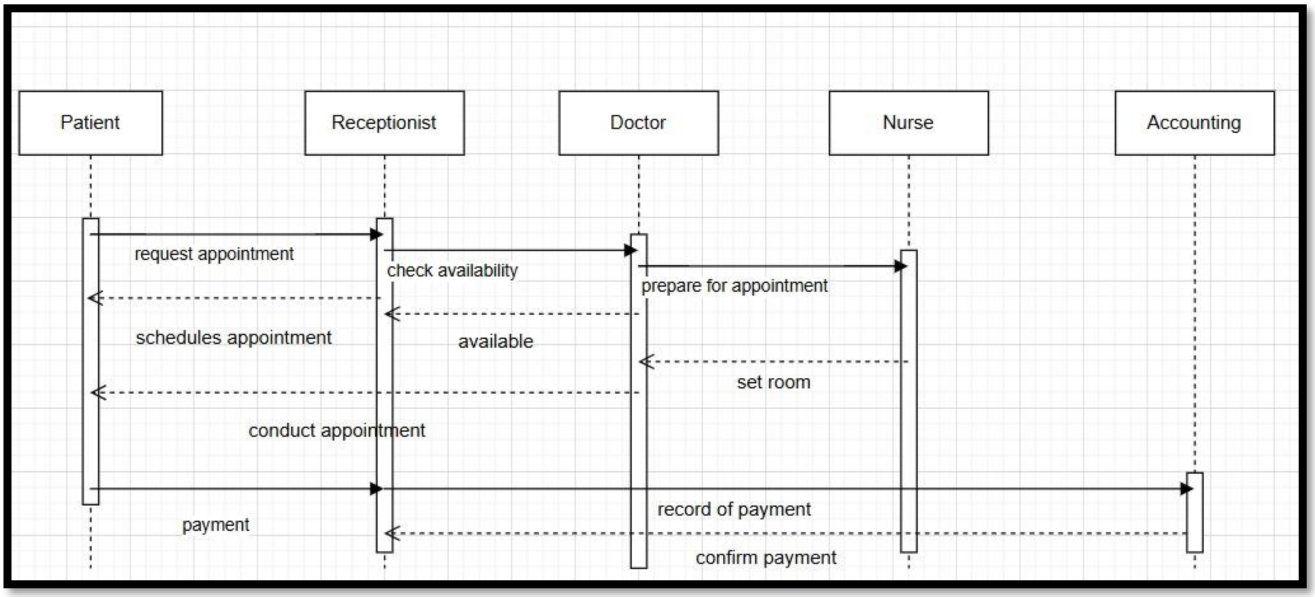
A **Sequence Diagram** is a type of **UML (Unified Modeling Language)** diagram that models the **interaction between system components (objects or actors)** over time. It focuses on the **order of messages exchanged** to carry out a specific functionality or use case. Each diagram shows how and in what sequence processes operate and interact to complete a task.

Purpose in HMS

In the context of the **Hospital Management System (HMS)**, sequence diagrams help visualize how users (e.g., receptionists, doctors, lab technicians) interact with the system to perform tasks like patient registration, scheduling appointments, or accessing medical records.

Key Components of a Sequence Diagram

- **Actors**: External entities that interact with the system (e.g., Patient, Doctor, Receptionist)
- **Objects/Components**: Internal system parts (e.g., Registration Module, Appointment Ser
- **Lifelines**: Vertical dashed lines that represent the lifespan of each actor or object during the interaction.
- **Messages**: Arrows showing communication between actors and system components
- **Activation Bars**: Vertical rectangles on lifelines that represent the time during which an object is performing a process.



Class Diagram

Definition of Class Diagram

A **Class Diagram** is a core component of the **UML (Unified Modeling Language)** used to represent the **static structure of a system**. It defines the system's classes, their **attributes**, **methods (operations)**, and the **relationships** among objects.

Purpose in HMS

In the **Hospital Management System (HMS)**, the class diagram provides a blueprint for implementing the system's functionalities using object-oriented programming. It helps in:

- Identifying reusable components and logical grouping.
- Defining interactions between different modules like patients, staff, billing, and inventory.
- Establishing data encapsulation and method responsibilities for each class

Class	**Attributes (Data)**	**Methods (Functions)**
Patient	patientID, name, age, gender, contactInfo,	register(), viewProfile(), updateInfo()
Doctor	doctorID, name, specialty, schedule, contactInfo	viewSchedule(), prescribe(), viewPatient
Appointment	appointmentID, date, time, patientID, doctorID	book(), cancel(), reschedule()
Billing	billID, patientID, amount, paymentStatus, dateIssued	generateBill(), calculateTotal(),
Medicine	medID, name, quantity, expiryDate, price	updateStock(), issueMedicine()
LabTest	testID, patientID, testType, result, doctorID	orderTest(), enterResult(), viewResult
Staff	staffID, name, role, department, contact	login(), assignTask(), updateProfile()
User	userID, username, password, role	login(), logout(), changePassword()

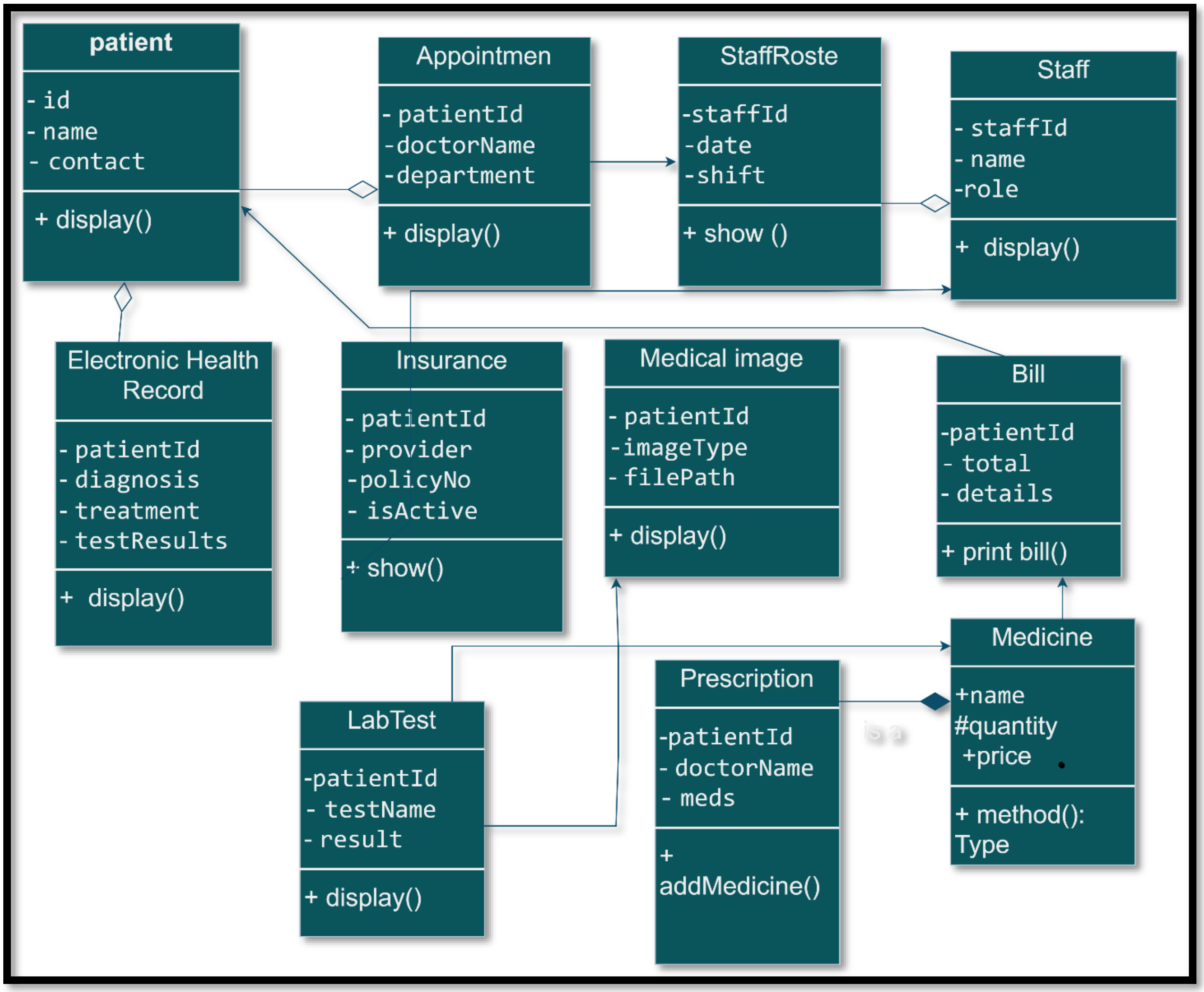


Fig ; Class diagram for HMS

Interface Design

Definition of Interface Design

Interface Design refers to the visual layout and interaction structure between users and the software system. In the context of the **Hospital Management System (HMS)**, it includes the graphical user interfaces (GUIs) through which various users (e.g., doctors, nurses, receptionists, lab technicians, administrators) interact with the system. A well-designed interface ensures usability, accessibility, and efficiency in completing healthcare-related tasks.

Purpose in HMS

The goal of the HMS interface design is to provide a clean, intuitive, and role-specific experience that simplifies hospital operations. Since the system is used by users with varying levels of technical skill, the interface must be:

- **User-friendly** and easy to navigate.
- **Responsive** and accessible across multiple devices.
- **Role-based**, displaying only relevant modules and actions.
- **Secure**, ensuring sensitive data is accessible only to authorized users.

Key Interfaces in HMS

Below are the primary user interfaces designed for HMS:

1. ***Login Interface***

- Fields: Username, Password, Role selection
- Features: Forgot password, secure login
- Redirects users to appropriate dashboard based on role

2. ***Patient Registration Interface***

- Form for entering patient information (name, DOB, contact, medical history)
- Validations for required fields
- Submit, Clear, and Edit options

3***. Appointment Scheduling Interface***

- Calendar view for selecting date/time
- Doctor availability display
- Confirmation popup after scheduling

4. ***Doctor Dashboard***

- View schedule
- Access to patient history and EHR
- Add diagnosis, prescribe medication

5. ***Billing Module Interface***

- Add services, medicines, room charges
- Auto calculation of total amount
- Generate and print invoice

6. ***Pharmacy Inventory Interface***

- Add/edit medicine stock
- Dispense medicine to patients
- Alerts for low stock

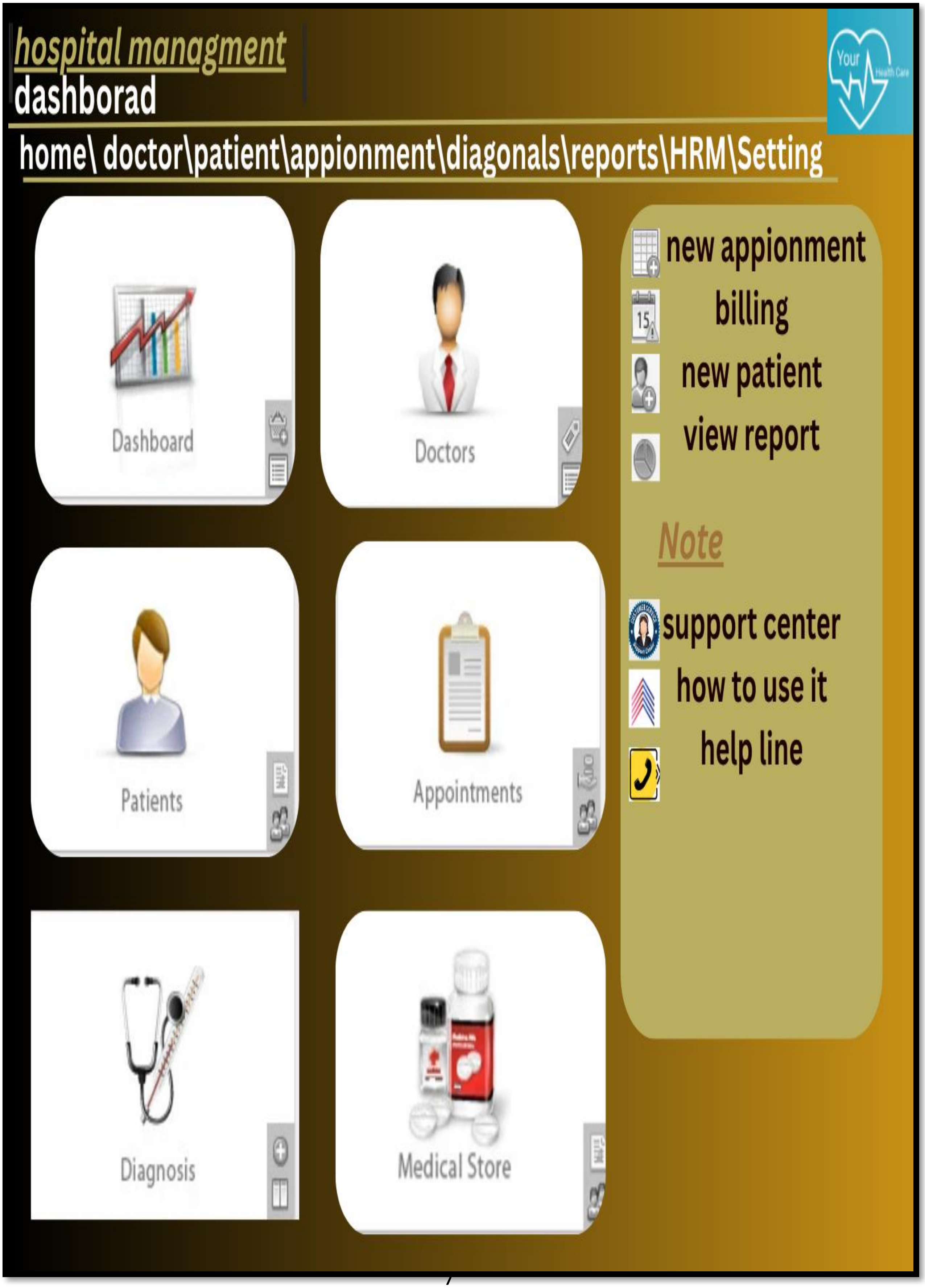
7. ***Lab Test Interface***

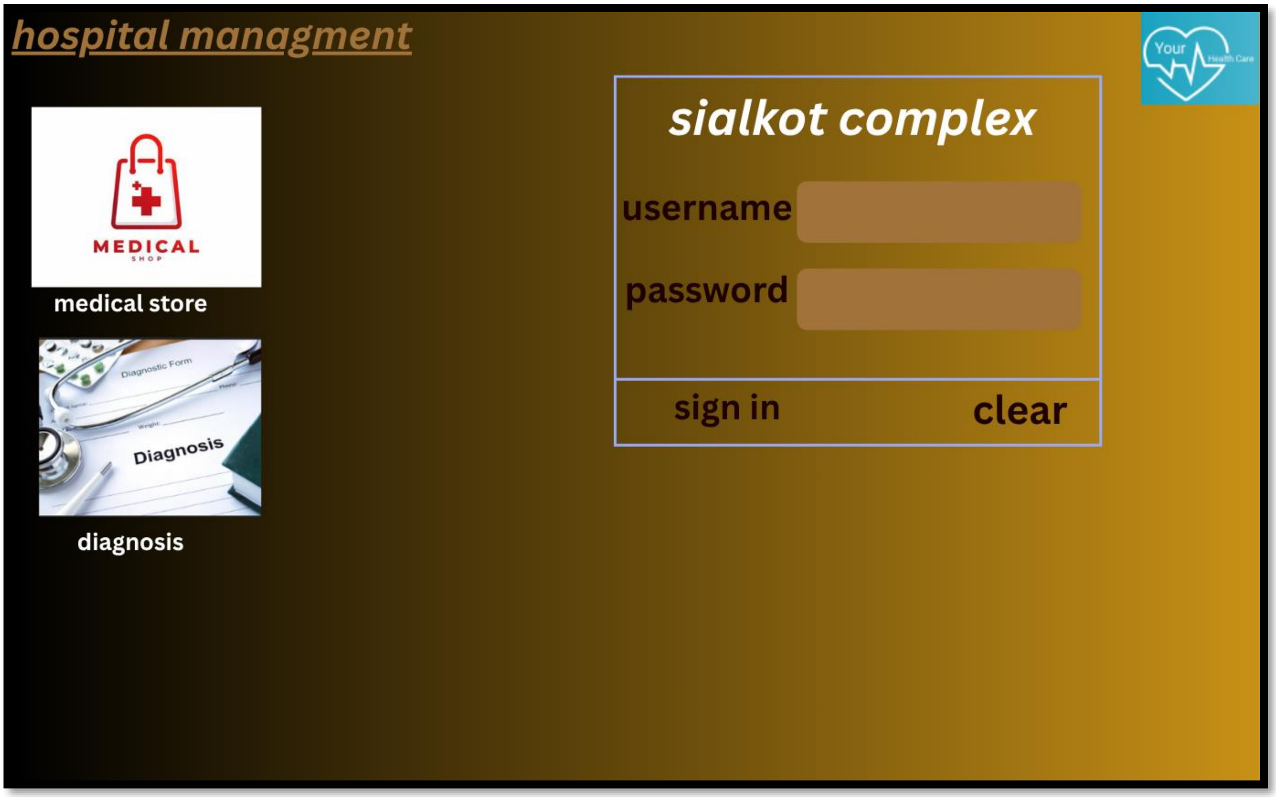
- View test orders
- Enter test results
- Upload and view reports

Design Considerations

- **Navigation:** Tab-based or sidebar navigation with clear module separation.
- **Accessibility:** Color contrast, font size options, and keyboard navigation support.
- **Error Handling:** User-friendly error messages and input validations.
- **Feedback:** Visual indicators (loading spinners, confirmations) for system responses.

The **User Interface (UI)** of the Hospital Management System (HMS) is designed to be intuitive, user-friendly,and role-specific, ensuring seamless interaction for all categories of users, including doctors, nurses, receptionists, administrators, and patients. The UI follows a clean, responsive layout with clearly labeled navigation menus, forms, buttons, and dashboards tailored to each role's responsibilities. For instance, a doctor’s dashboard provides quick access to patient records, appointments, and lab reports, while a receptionist’s interface focuses on registration and appointment scheduling. The interface incorporates validation checks, dropdown menus, calendar pickers, and confirmation pop-ups to reduce user errors and improve efficiency.


hospital managment
dashborad
Your Health Care
home\ doctor\patient\appionment\diagonals\reports\HRM\Setting
Dashboard
Doctors
Patients
Appointments
Diagnosis
Medical Store
new appionment
billing
new patient
view report
Note
support center
how to use it
help line



hospital managment

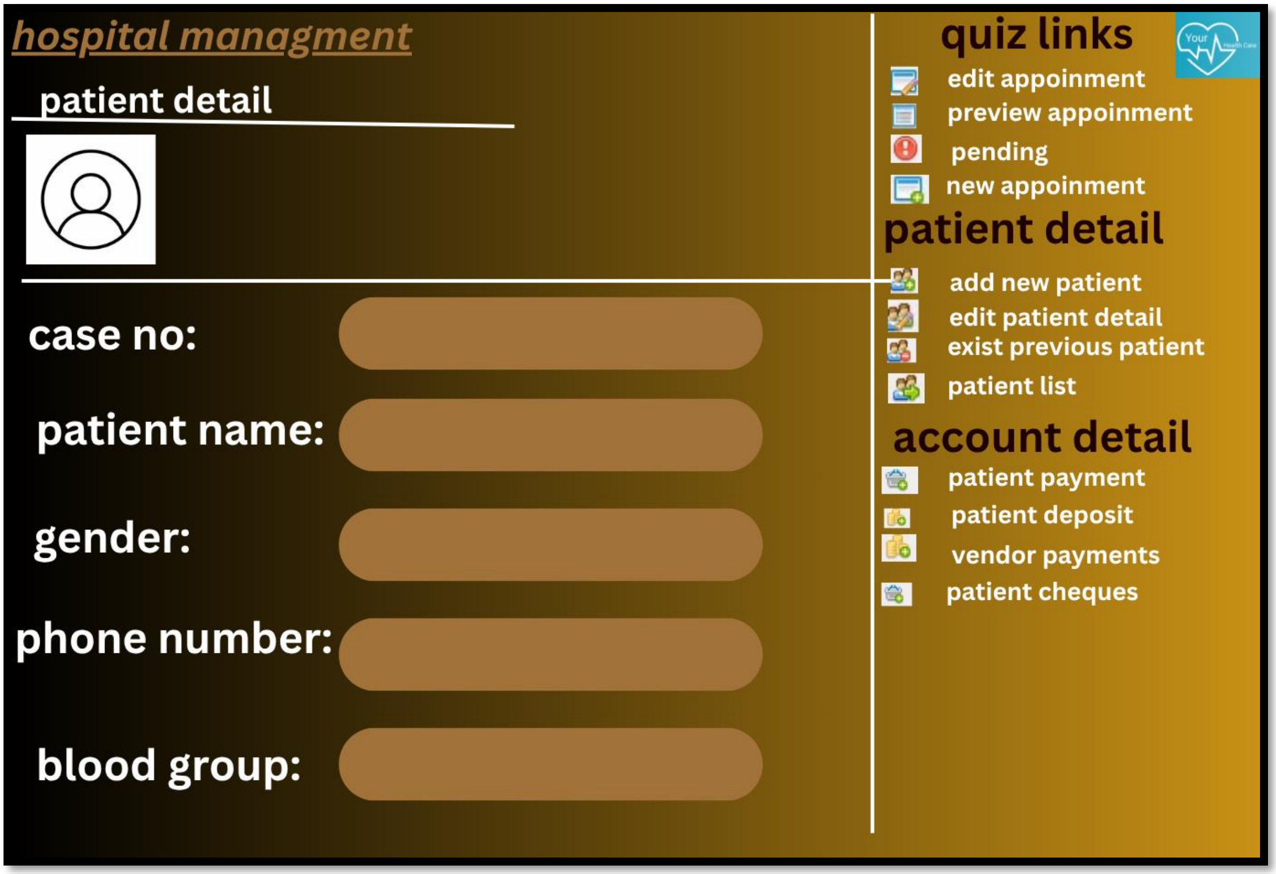
medical store

diagnosis

sialkot complex

username

password



hospital managment

patient detail

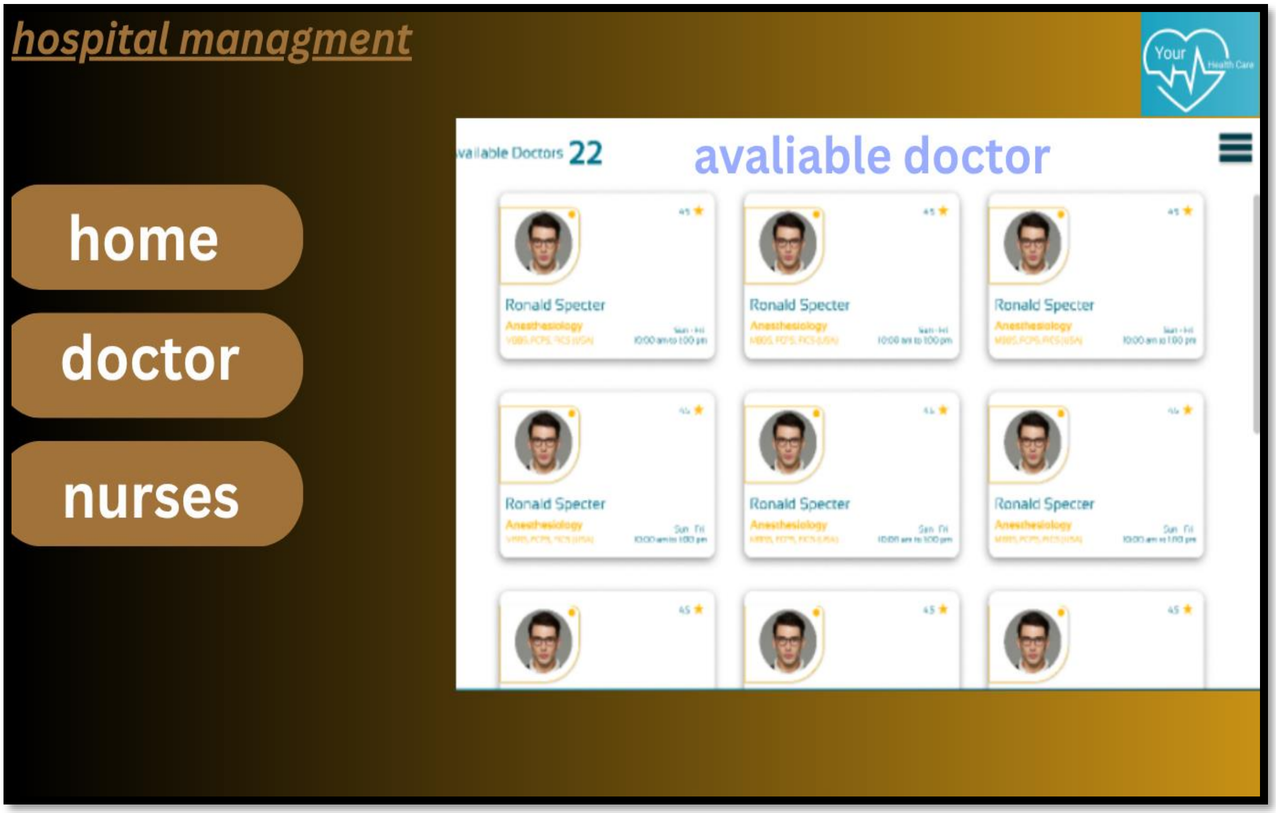
case no:

patient name:

gender:

phone number:

blood group:



hospital managment

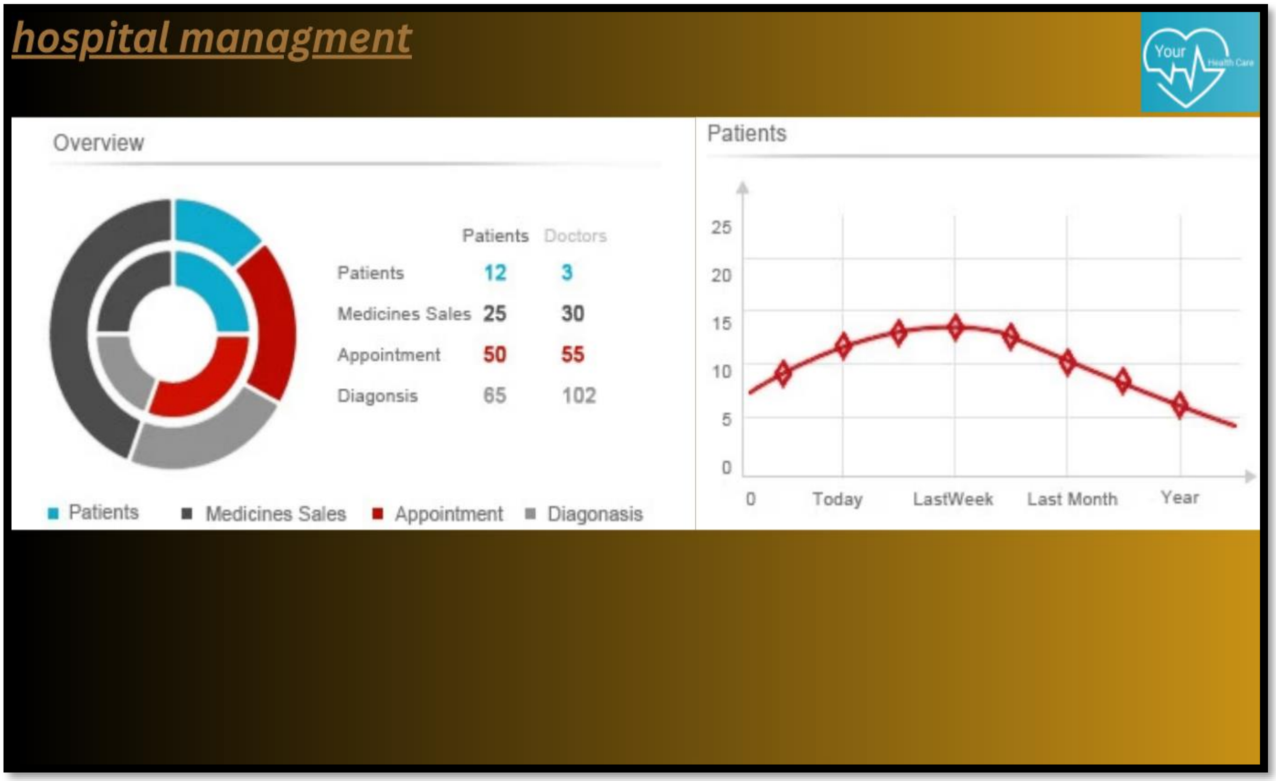
home

doctor

nurses

vailable Doctors 22

avaliable doctor



hospital managment

Overview

	Patients	Doctors
Patients	12	3
Medicines Sales	25	30
Appointment	50	55
Diagonsis	65	102

Test Cases

Test cases are structured scenarios or sets of conditions used to **verify that a software system functions correctly** according to its requirements. Each test case specifies:

- **What is being tested**
- **How to test it (steps)**
- **What the expected result should be**
- **Whether the system passes or fails the test**

Test cases are essential in software quality assurance. They ensure that:

- All features work as intended
- Bugs or errors are identified before deployment
- System behavior is consistent under different conditions

Field	**Description**
Test Case ID	Unique identifier for the test case (e.g., TC001)
Test Scenario	Description of what feature or function is being tested
Preconditions	Any setup or data required before running the test
Test Steps	Detailed steps to execute the test
Test Data	Input values needed for the test
Expected Result	What the system should do in response to the inputs
Actual Result	What actually happened when the test was run
Status (Pass/Fail)	Outcome of the test — whether expected and actual results matched
Remarks	Notes about errors, bugs, or observations during testing

Hospital Management System- Test Cases

ID	Test Case	Test Execution	Expected Results	Status	Remarks
1	Patient Registration	Navigate to Patient Registration page → Enter personal details, medical history → Click Submit	Patient successfully registered	Pass	Patient was registered and confirmation received
2	Patient Registration - Missing Info	Navigate to registration → Leave required fields blank → Click Submit	System should prompt for missing required fields	Fail	System accepted form without validating required fields
3	Appointment Scheduling	Navigate to Appointment Scheduling → Select patient, doctor, date and time → Click Confirm	Appointment scheduled successfully	Pass	Appointment listed on both schedules
4	Appointment Scheduling - Overlap	Schedule appointment for Patient A with Doctor X at 10:00 AM → Schedule another for Patient B same time	System should block double-booking	Fail	Appointment allowed despite doctor being unavailable
5	User Login Authentication	Go to login page → Enter valid credentials → Click Login	Dashboard appears based on role	Pass	User redirected to correct role dashboard
6	Staff Login - Unauthorized Access	Attempt to access admin panel using nurse account	Access denied and redirect to dashboard	Fail	Nurse account accessed restricted admin settings

7	Billing Generation	Open billing module → Select patient → Add services and charges → Click Generate	Invoice is generated with breakdown	Pass	Invoice displayed with accurate totals
8	Billing - Incorrect Total	Add consultation + procedure + room charges → System calculates total	Total should match sum of all charges	Fail	Calculated total missed room charges
9	Medicine Inventory Update	Open pharmacy module → Add new medicine stock → Save changes	Inventory updated successfully	Pass	Inventory reflects new stock values
10	Pharmacy Inventory - Negative Stock	Dispense medicine beyond available quantity	System should prevent dispensing	Fail	System allowed negative stock, breaking inventory
11	EHR Storage	Access patient's record → Enter diagnosis and test results → Save record	Data saved and accessible under patient profile	Pass	EHR available under patient view
12	EHR - Unauthorized Record Access	Receptionist tries to view mental health record	Access denied	Fail	Receptionist accessed sensitive medical history
13	Lab Test Order	Doctor opens patient profile → Select lab test and submit order	Test order submitted to lab	Pass	Order appears in lab technician's queue
14	Lab Test Result Entry - Format Error	Doctor enters text in numeric-only field	System should flag invalid format	Fail	System saved text data in numeric field without validation

Glossary

Term	Definition
HMS (Hospital Management System)	A software application designed to streamline and automate operations within a hospital or clinic.
UI (User Interface)	The visual part of the application where users interact with the system via screens, buttons, and forms.
UML (Unified Modeling Language)	A standardized modeling language used to visualize system design, including class and sequence diagrams.
Patient ID	A unique identifier assigned to each patient for tracking and record-keeping purposes.
EHR (Electronic Health Record)	A digital version of a patient's medical history, test results, treatments, and other clinical information.
Module	A separate functional unit within the HMS, such as billing, appointment scheduling, or pharmacy.
Role-Based Access	A security feature that grants different access levels based on the user's role (e.g., doctor, nurse).
Test Case	A documented scenario to verify that a particular feature of the software behaves as expected.
Sequence Diagram	A diagram showing how system components or users interact over time during a specific process.
Class Diagram	A diagram representing the classes, attributes, and relationships in an object-oriented system.
Entity Relationship Diagram (ERD)	A diagram that maps out the data model, showing entities and their relationships in the database.
Appointment Scheduling	The module responsible for booking, managing, and rescheduling appointments between patients and doctors.
Billing Module	A part of the system that manages patient invoices, service charges, and payment statuses.

Inventory Management	The module that tracks medicines and supplies in the hospital, including stock levels and expiration.
Lab Module	The section of the HMS used for ordering, processing, and reporting diagnostic tests and results.
Authentication	The process of verifying a user's identity before granting access to the system.
Authorization	The process of assigning specific permissions to users based on their role or profile.
Database	A structured collection of data used by the HMS to store and retrieve information.
Login Interface	The screen where users enter their credentials to access the system.
Validation	A process that ensures data entered into the system meets required formats or constraints.