Hospital Managment System

Project Title

<Student Name (Roll No)>: Waseem Akram(F22BINFT1E02126)

Department of Information Technology

Faculty of Computing

The Islamia University of Bahawalpur

Meeting Details

Sr No	Details	Date	Supervisor Signature

Summary

Here the author will write summery of the document. They can provide a brief of the background, their findings and requirements.

1. Introduction Error! Bookmark not defined.

1.1 Overview of the System
1.2 Problem Statement
1.3 Purpose of the Project
1.4 Objectives of the System
1.5 Scope of the Project
1.6 Expected Benefits
1.7 Technologies Used

1.1. Requirements Error! Bookmark not defined.

2.1 Functional Requirements

- Patient Management
- Doctor Management

- Generate Bill
- Manage Pharmacy
- Doctor Dashboard
- Admin Management

3.3 Process Flow Diagrams

- Patient Registration Flow
- Appointment Booking Flow
- Billing Process Flow
- Pharmacy Process Flow
- Lab Report Process Flow

3.4 Data Flow Diagrams (Optional)

- DFD Level 0
- DFD Level 1
- DFD Level 2

3.5 ERD (Entity Relationship Diagram)
3.6 Class Diagram (If using OOP)

1.3 References 4

4.1 Books
4.2 Research Papers
4.3 Websites and Online Articles
4.4 Tools Documentation (VS Code, MySQL, etc.)
4.5 APA/MLA Style Referencing (as per university guidelines)

1. Introduction

1.1 Overview

The Hospital Management System (HMS) is a computerized solution designed to manage and organize hospital operations. It helps in handling patient information, doctor schedules, appointments, billing, pharmacy records, and laboratory reports.
The system reduces manual work, eliminates errors, and improves hospital efficiency.

1.2 Problem Statement

Most hospitals use manual registers and paperwork to manage patient records, appointments, and billing. This leads to:

- Lost information
- Slow processing
- Human errors
- Long queues
- Difficulty retrieving old records
  There is a need for a centralized digital system to solve these issues.

1.3 Purpose of the Project

The purpose of HMS is to create a unified platform where hospital data is stored securely and can be accessed quickly.

It simplifies hospital operations and ensures smooth coordination among staff, doctors, and patients.

1.4 Objectives

- To store patient and doctor information in a digital database
- To automate appointment scheduling
- To generate accurate billing
- To maintain pharmacy and lab records
- To provide role-based access for admin, doctors, and receptionists
- To improve patient experience

1.5 Scope

The system covers:

- Patient registration
- Appointment booking
- Doctor availability
- Prescription handling
- Billing system
- Pharmacy inventory
- Lab test management
- Admin controls
  (Advanced modules like OPD, IPD, ambulance system can be included if required.)

1.6 Expected Benefits

- Faster hospital operations
- Reduced paperwork

- Accurate records
- Improved patient satisfaction
- Better data security
- Easy retrieval of patient history
- Automated billing and reporting

1.7 Technologies Used

- **Frontend:** HTML, CSS, JavaScript / React (optional)
- **Backend:** PHP / Python / Java
- **Database:** MySQL
- **Tools:** VS Code, XAMPP/WAMP, GitHub

---

2. Requirements

2.1 Functional Requirements

Patient Management

- Add, update, delete patient profiles
- Maintain patient history
- Search patient records

Doctor Management

- Add doctor details
- Assign specialties
- View doctor schedules

Appointment Scheduling

- Book appointments

- Cancel or update appointments
- Check doctor availability

Billing & Payments

- Generate bills automatically
- Add charges (consultation, lab, pharmacy)
- Issue payment receipts

Pharmacy Module

- Manage medicines
- Track stock
- Generate invoice for medicine purchase

Laboratory Module

- Create lab test requests
- Upload test results
- Attach reports with patient history

Admin Panel

- Manage all users
- View full reports
- Control settings

User Authentication

- Login system
- Role-based access (Admin, Doctor, Receptionist, Pharmacist)

2.2 Non-Functional Requirements

Performance

- System should load within 2–3 seconds
- Handle multiple users at once

Security

- Password protected
- Secure database
- Role-based permissions

Usability

- Clean interface
- Easy navigation

Reliability

- System must work without errors
- Consistent uptime

Scalability

- Add more modules in future

Backup & Recovery

- Database backup
- Restore data in case of system crash

2.3 Hardware Requirements

- Processor: Core i3 or above
- RAM: 4 GB minimum
- Hard Disk: 20 GB free space

- Server or localhost environment

2.4 Software Requirements

- Windows/Linux
- MySQL
- XAMPP / WAMP
- VS Code / NetBeans / IntelliJ
- PHP/Python/Java runtime

---

3. Use Cases and Flow of Processes

3.1 Use Case Diagram

(You can draw using Draw.io)
Actors:

- Admin
- Receptionist
- Doctor
- Patient
- Pharmacist
- Lab Technician

3.2 Use Case Descriptions

Use Case: Register Patient

- Actor: Receptionist
- Steps:

    1. Enter patient details
    2. Save to database
    3. Generate patient ID

Use Case: Manage Appointments

- Actor: Receptionist/Patient
- Steps:
    1. Select doctor
    2. Choose date/time
    3. Confirm booking

Use Case: Doctor Dashboard

- Actor: Doctor
- Steps:
    1. Login
    2. View appointments
    3. Add diagnosis/prescription

Use Case: Generate Bill

- Actor: Receptionist
- Steps:
    1. Add consultation fee
    2. Add tests/medicine charges
    3. Print bill

Use Case: Pharmacy Process

- Actor: Pharmacist

- Steps:
    1. Add medicine
    2. Check availability
    3. Issue medicine
    4. Update stock

Use Case: Lab Report Process

- Actor: Lab Technician
- Steps:
    1. Add test request
    2. Upload report
    3. Attach to patient record

3.3 Process Flow Diagrams

(Explain each flow, add arrows in diagram tool)

- **Patient Registration Flow**
- **Appointment Flow**
- **Billing Flow**
- **Pharmacy Workflow**
- **Lab Workflow**

3.4 Data Flow Diagrams

- **DFD Level 0:** Basic system overview
- **DFD Level 1:** Detailed modules

- **DFD Level 2:** In-depth data movement

3.5 ERD (Entity Relationship Diagram)

Tables:

- Patient
- Doctor
- Appointment
- Billing
- Medicine
- LabTest
- Users (Login)

---

4. References

Use APA format:

Books

- Sommerville, I. *Software Engineering*
- Pressman, R. *Software Engineering: A Practitioner's Approach*

Websites

- https://healthit.gov
- https://developer.mozilla.org
- https://mysql.com

Research Papers

- Articles on hospital automation systems

Tools Documentation

- PHP documentation
- MySQL documentation