

# **Project Proposal**

## **On**

## **Hospital Management System**

**Guided By:** Mr. Anuj Kumar

**Created By:** Silki Chauhan , Vikul Saharan

**AFid:** AF04991241 and AF04991727

**Batch Code:** ANP-D2405

**Course Code:** ITPR

## **Table Of Contents**

# **Index**

- 1.** Title of the Project.....
- 2.** Introduction.....
- 3.** Objective
- 4.** Project Category
- 5.** Analysis
  - Modules and Description
  - Database Design
  - ER Diagram
  - Data Flow Diagram
- 6.** Complete Structure
  - Process Logical Diagram
- 7.** Platform Used
  - Hardware Requirement
  - Software Requirement
- 8.** Future Scope
- 9.** Bibliography

## **INTRODUCTION**

The Hospital Management System (HMS) is a digital solution designed to manage all hospital activities efficiently. It helps in handling patient records, doctor information, appointment scheduling, staff details, medicines, and billing in a smooth and organized way.

By replacing manual work with an automated system, it reduces errors, saves time, improves accuracy, and ensures better patient care.

This system provides a centralized platform where all hospital data can be accessed quickly and securely, making hospital operations faster and more efficient.

This system is specifically designed to enhance hospital productivity, improve data accuracy, and support healthcare staff in delivering better patient services.

## **OBJECTIVE**

The primary objectives of the Hospital Management System are:

1. To digitalize patient registration and medical records
2. To automate doctor scheduling and appointment handling
3. To maintain accurate billing and payment records
4. To reduce manual workload and minimize human errors
5. To provide secure and role-based access to hospital staff
6. To enable fast and reliable data retrieval whenever required

## **PROJECT CATEGORY**

Language: Java

Database: MySQL

Connectivity: JDBC

IDEs: Eclipse

Tools: MySQL Server

# ANALYSIS

## Modules and Description

### 1. Patient Management Module

- Add new patient
- View all patients
- Search patient by ID
- Remove patient by ID

### 2. Doctor Management Module

- View all doctors
- Search doctor by ID

### 3. Appointment Management Module

- Book appointment
- Check patient exists
- Check doctor availability
- View all appointments

### 4. Medicines Management Module

- Add medicine
- Update stock
- View all medicines

## 5. Billing Module

- Create patient bill
- Calculate total amount
- Manage payment status

## 6. User/ Admin Module

- Secure login system for administrator
- Admin can add/remove users
- Role-based access

## 7. Staff Module

- Add staff
- View staff
- Search patient by ID

# Database Design

Main tables used:

## PATIENTS Table Design:

Fields	Datatype	Properties
patient_id	int	Primary key, not null, auto_increment
Patient_name	varchar(100)	not null
age	int	check
gender	varchar(100)	not null
Blood_group	Varchar(10)	Not null
contact	varchar(100)	not null,unique
Disease	Varchar(100)	Null allowed
Registration_date	date	Not null
address	varchar(100)	not null

## DOCTORS Table Design:

<b>Fields</b>	<b>Datatype</b>	<b>Properties</b>
doctor_id	int	Primary key, not null, auto_increment
Doctor_name	varchar(100)	not null
specialization	varchar(100)	not null
doctor_fees	int	Not null
Phone	Varchar(15)	Not null, unique
Email	Varchar(100)	Unique
Experience	Int	check
availability	Varchar(50)	Not null

## APPOINTMENT Table Design:

<b>Fields</b>	<b>Datatype</b>	<b>Properties</b>
appointment_id	int	Primary key, not null, auto_increment
patient_id	int	Foreign key, not null
doctor_id	int	Foreign key, not null
appointment_date	date	not null
Appointment_time	time	Not null
remarks	Varchar(255)	null

### MEDICINE Table Design:

Fields	Datatype	Properties
medicine_id	int	Primary key, not null, auto_increment
Medicine_name	varchar(100)	not null
quantity	int	not null, check
price	decimal(10,2)	not null, check
company	Varchar(100)	Not null
Expiry_date	Date	Not null

### BILL Table Design:

Fields	Datatype	Properties
bill_id	int	Primary key, not null, auto_increment
patient_id	int	Foreign key, not null
doctor_fee	decimal(10,2)	not null
room_charges	decimal(10,2)	Default 0
medicine_charges	decimal(10,2)	Default 0
other_charges	decimal(10,2)	Default 0
total_amount	decimal(10,2)	not null
payment_status	varchar(20)	not null
bill_date	date	not null
Appointment_id	int	Foreign key
Consultation_fee	double	Not null

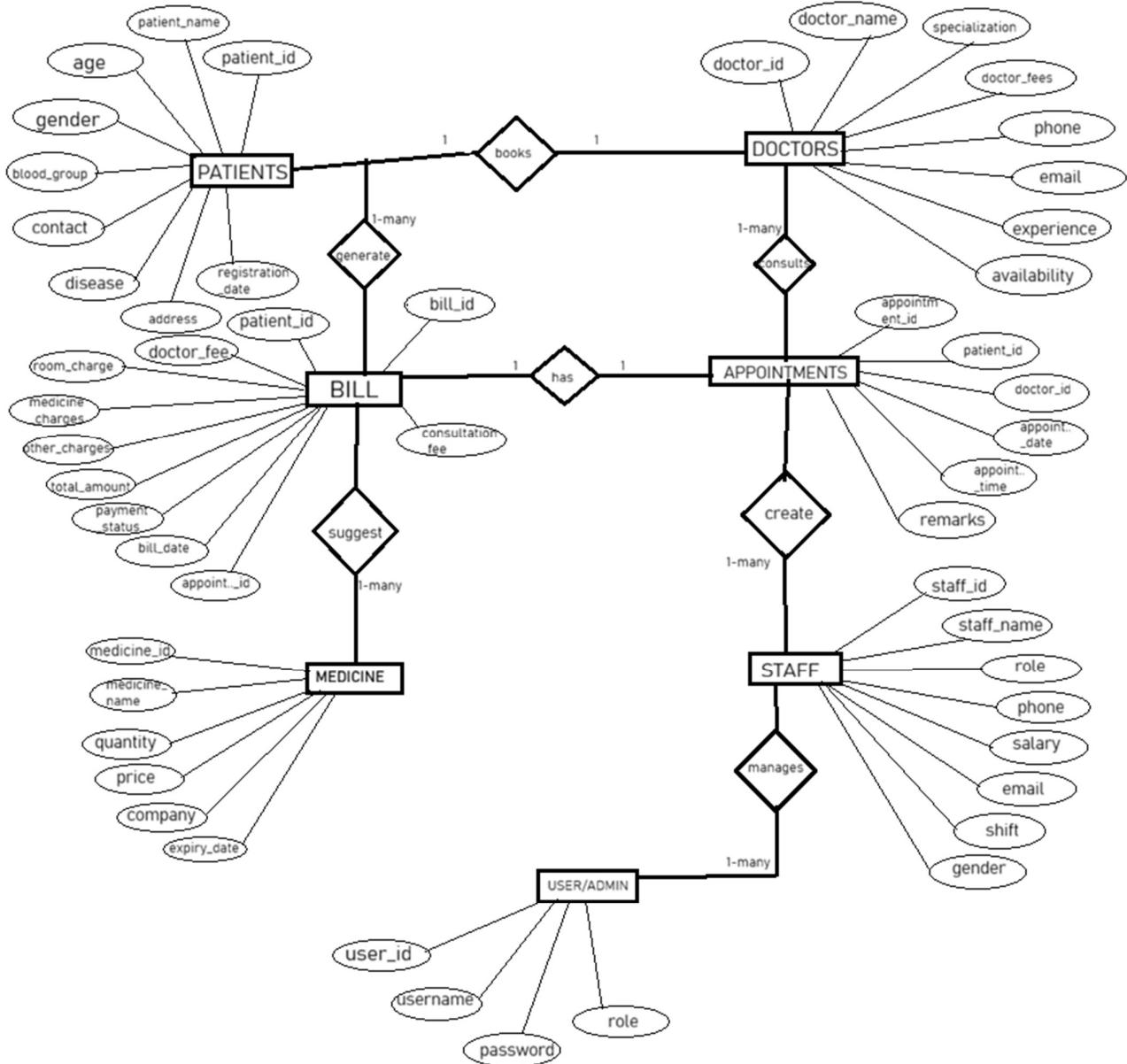
### STAFF Table Design:

<b>Fields</b>	<b>Datatype</b>	<b>Properties</b>
staff_id	int	PRIMARY KEY, AUTO_INCREMENT, NOT NULL
Staff_name	varchar(100)	not null
role	varchar(50)	not null
phone	varchar(15)	not null,unique
salary	decimal(10,2)	not null
Email	Varchar(100)	Unique
Shift	Varchar(50)	Not null
Gender	Varchar(10)	Not null

### USER/ADMIN Table Design:

<b>Fields</b>	<b>Datatype</b>	<b>Properties</b>
user_id	int	Primary key, not null
username	varchar(50)	Unique not null
password	varchar(100)	not null
role	varchar(30)	Default 'admin'

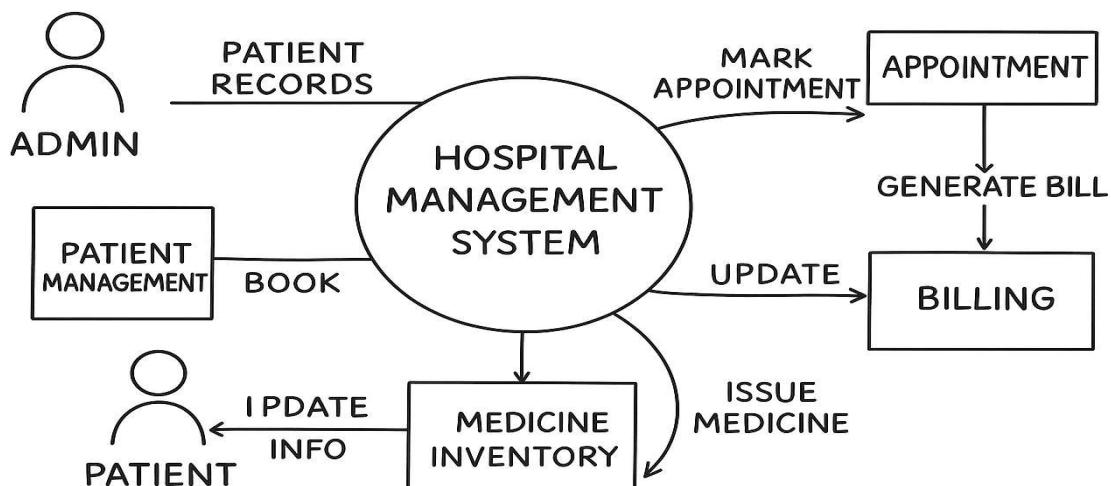
# ER DIAGRAM



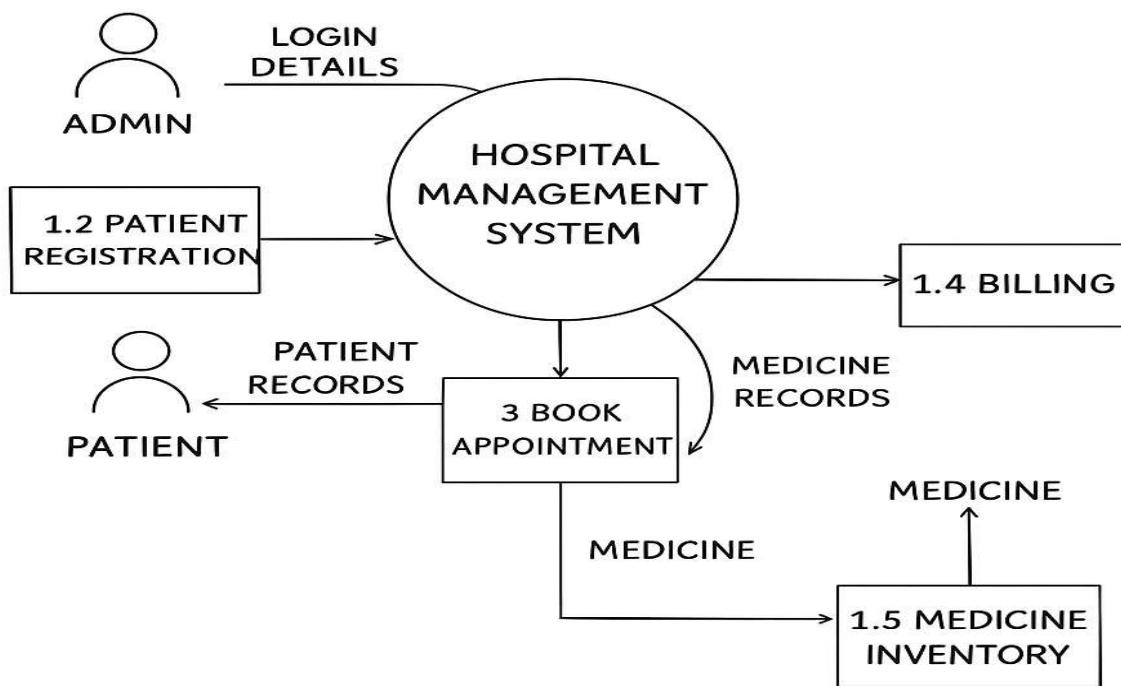
# DATA FLOW DIAGRAM (DFD)



## LEVEL-1 DFD

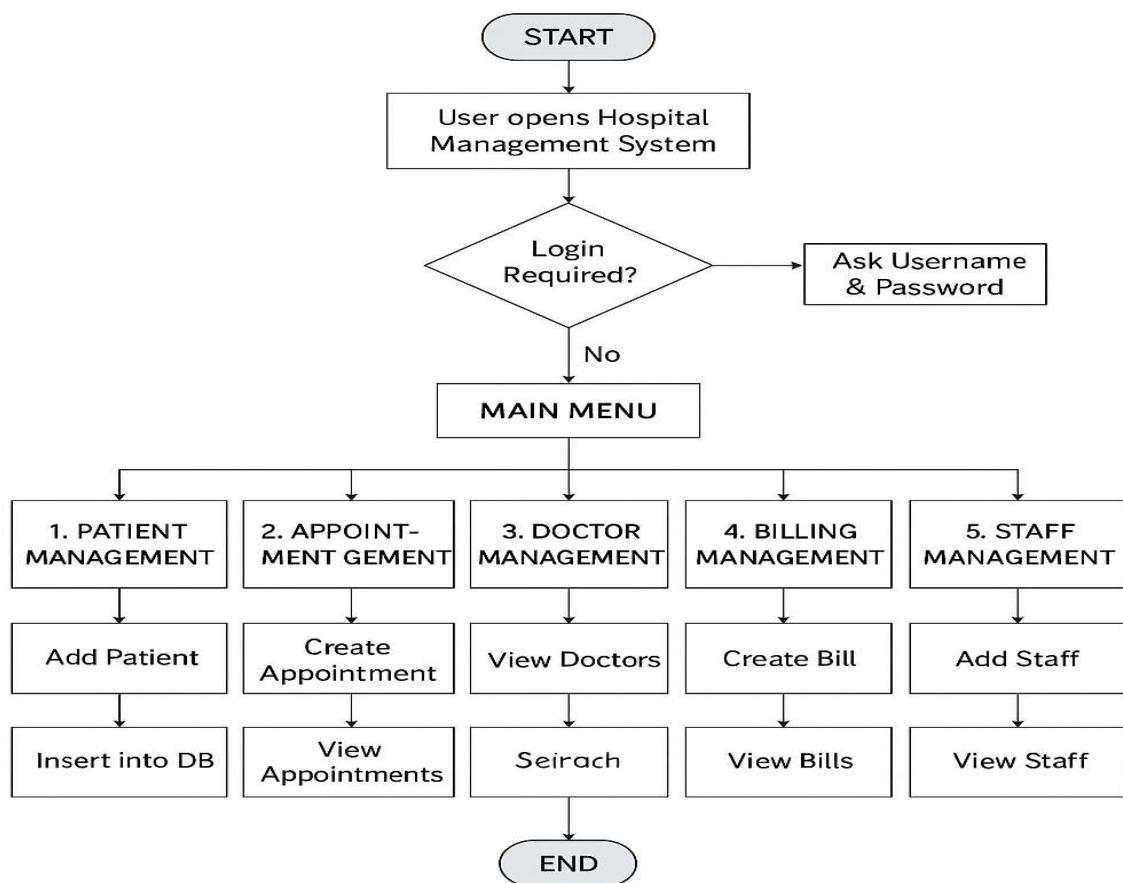


## LEVEL - 02 DFD



# COMPLETE STRUCTURE

- Process Logic Diagram



## **Platform Used**

### **a) Hardware Requirements**

- Processor: Intel i3 or higher
- RAM: Minimum 4GB
- Hard Disk: 500MB free space
- Operating System: Windows/Linux/Mac

### **b) Software Requirements**

- JDK 1.7 or above
- MySQL Server
- MySQL Workbench / phpMyAdmin
- IDE: Eclipse / IntelliJ IDEA / NetBeans
- JDBC Driver
- Plug in:- Maven
- OS: Windows 10/11

## **FUTURE SCOPE**

- Automated Billing with UPI/Online Payments
- Advanced Data Analytics & Reporting
- Cloud Hosting for Remote Access
- Mobile app integration
- Multi-Hospital Integration
- Automated Medicine Inventory System
- Advanced Data Analytics & Reporting

## **BIBLIOGRAPHY**

- Oracle Java Documentation
- MySQL Official Documentation
- JDBC API Guide
- Reference books on DBMS and Java Programming
- Online educational resources (GeeksforGeeks, TutorialsPoint, JavaTPoint)

