Project Proposal

On

Hospital   Management

System

Guided By: Anuj Kumar

Created By: Shasi, Shourya Sahu
           AF04991714, AF04991814

Batch Code: ANP-D2406

Course Code: ITPR-CSE32

Table Of Contents

2. Introduction

The Hospital Management System(HMS) is a simple software project made to help hospitals manage their daily work.

In many hospitals, work like patient registration, appointment booking, billing, and medicine records is done manually, which takes time and can cause mistakes.

This project helps to store all important information in a computer so the hospital can work faster and more accurately.

It makes the hospital system more organized and reduces paperwork. The project uses for building a simple and user-friendly desktop interface, **MySQL** for storing the data permanently, and **Maven** to manage the project dependencies. Overall, this system improves the working speed of the hopital, reduces paperwork, and helps, maintain all records in an organized way.

3. Objectives

1. To store patient information safely in a computer.
2. To help in booking appointments quickly.
3. To maintain doctor details like specialization and timings.
4. To maintain proper records of patients for future reference.
5. To automatically generate bills for patients.
6. To keep all data in one place so it is easy to search and update.
7. To help hospital staff easily update patient details without rewriting forms.
8. To support future expansion, like adding more modules (pharmacy, ward management)

4. Project Category

1. **Application type**: Command line base.
2. **Domain:** Database management system.

3. **Technology used:**
   - ✔ Java (Core java).
   - ✔ MySQL database.
   - ✔ Maven (Build Tool)
   - ✔ JDBC Connectivity

4. **Project Level:** Mini project
5. **Architecture:** 3-Tier architecture (UI -> Business logic/DAO Layer ->Database)
6. **Purpose:** To manage hospital data quickly, securely and efficiently.

5. Analysis

Modules and Description

1. Patient Module:

- New patient registration with essential demographic details.
- Update and modify patient information (name, age, gender, contact, address).
- Maintain unique patient IDs for accurate identification.
- Store and retrieve patient profiles quickly for hospital operations.

2. Doctor Module:

- Store details of all doctors including specialization and availability.
- Manage doctor schedules to support appointment booking.
- Assign doctors to patients based on department or condition.
- Maintain contact and professional information for internal use.

3. Appointment Scheduling Module:

- Book appointments between patients and doctors.
- Check time-slot availability to avoid conflicts.
- Maintain history of scheduled, completed, or cancelled appointments.
- Improve workflow by organizing daily appointment lists.

4. Billing & Payment Module:

- Generate patient bills for consultations, tests, and treatments.
- Maintain service records with associated charges.
- Calculate total payable amount automatically.
- Track paid and unpaid bills for financial accuracy.

Database Design

Database tables include:

Login Table:

Patients Table:

This table stores all information related to Patient .

Field Name	Data type	Description
patient-ID	Int (primary key)	Unique-ID for each patient
Name	Varchar	Name of the patient
Age	Varchar	Patient age
Gender	Varchar	Male/Female
Contact	int	Phone number
Address	Varchar	address

Doctors Table:

Stores doctor names, specialization, schedule, etc.

Field Name	Data Type	Description
Doctor-Id	INT (PK)	Unique doctor ID
Name	VARCHAR	Doctor name
Specialization	VARCHAR	Doctor specialist
Contact	VARCHAR	Contact number
Fee	INT	Consultation fee

Appointment Table:

Books appointments without clash in timings.

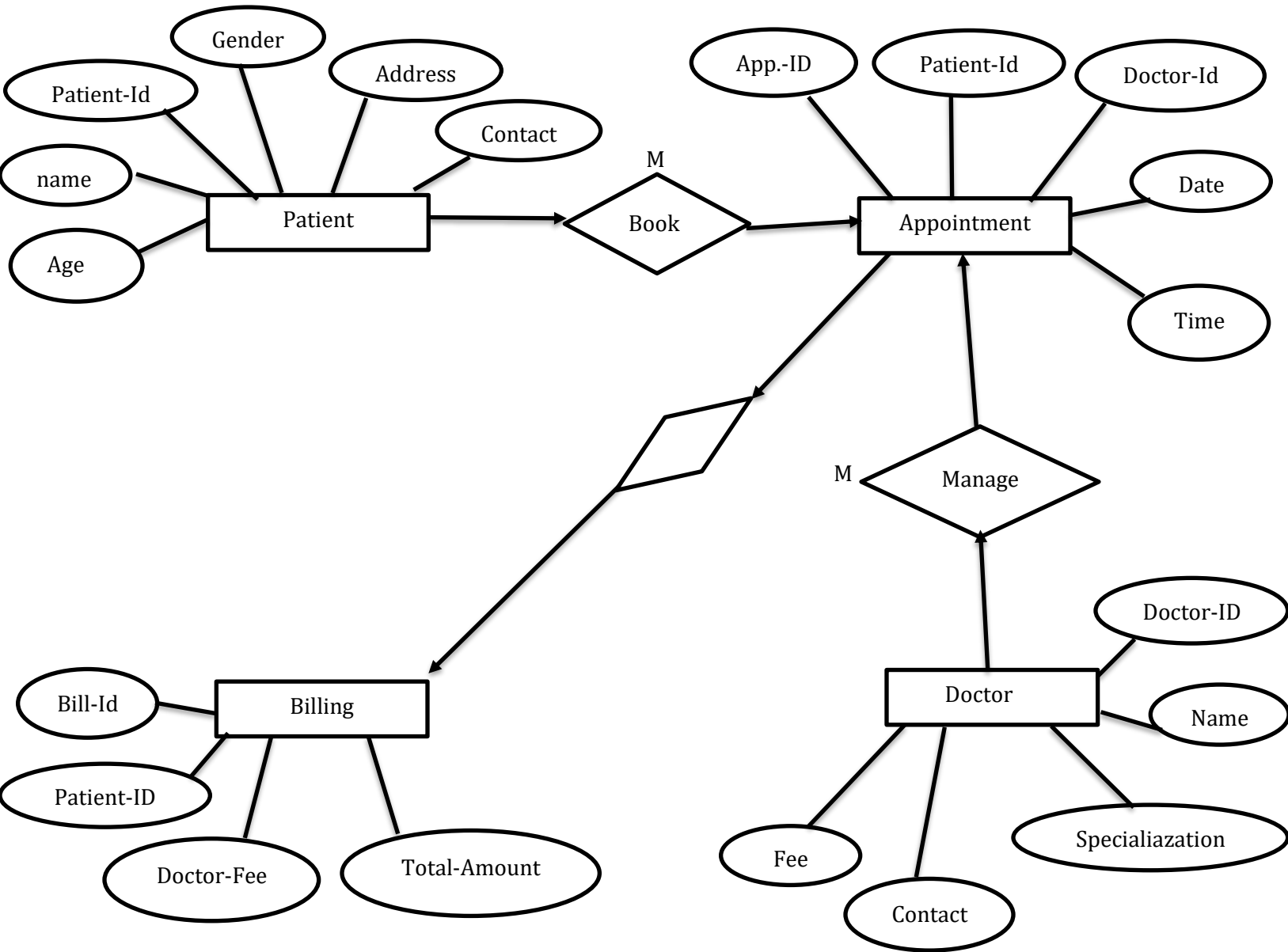| Field Name	Data Type	Description
Appointment-ID	Int (primary key)	Appointment number
Patient-ID	Int (Foreign key)	Linked to patients table
Doctor-ID	Int (Foreign key)	Linked to doctors table
Appointment-time	DATE	Time
Appointment-Date	DATE	Appointment date

Billing Table:

Creates final bill for the patient.

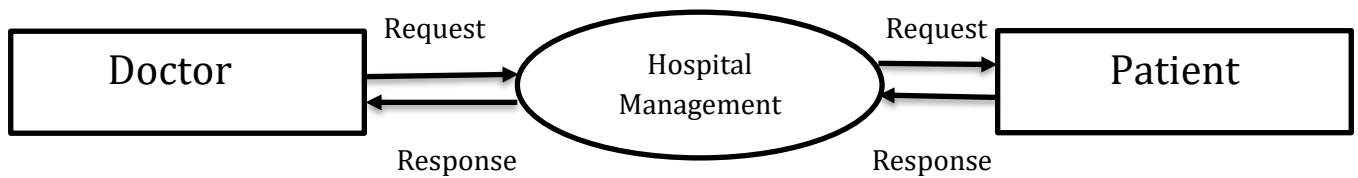| Field Name	Data Type	Description
Bill-ID	Int (Primary key)	Bill no
Patient-ID	Varchar	Patient linked to this bill
Doctor-fee	Varchar	Consultation charge
Total-Amount	INT	Final total amount

ER Diagram

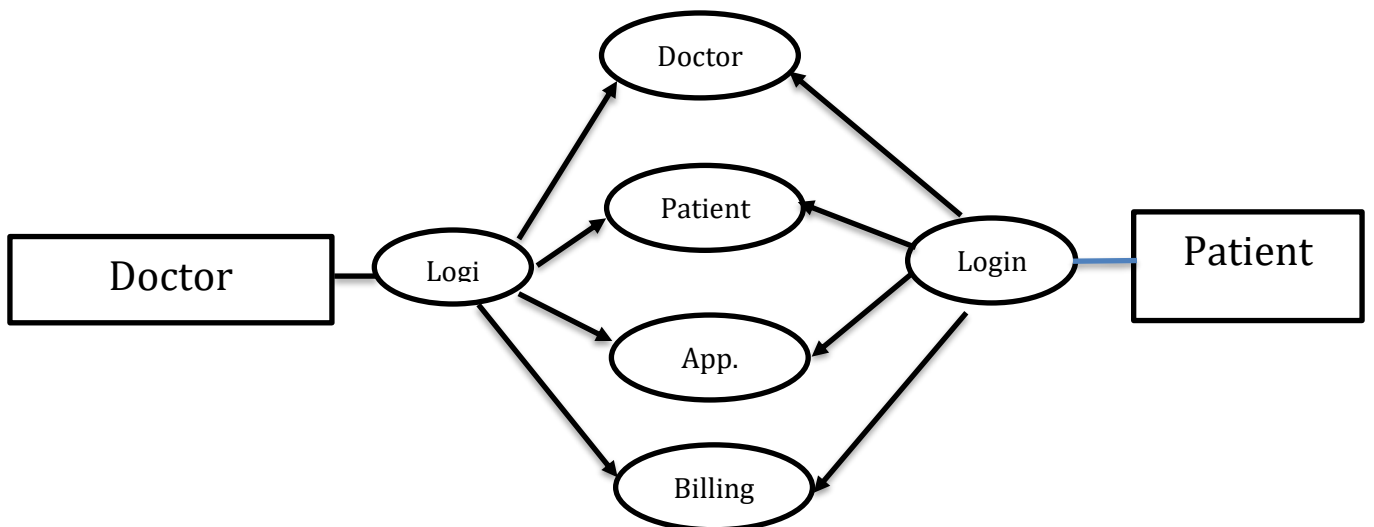The ER diagram represents relationships between Doctors, patients, Appointments, Billing .

Data Flow Diagram

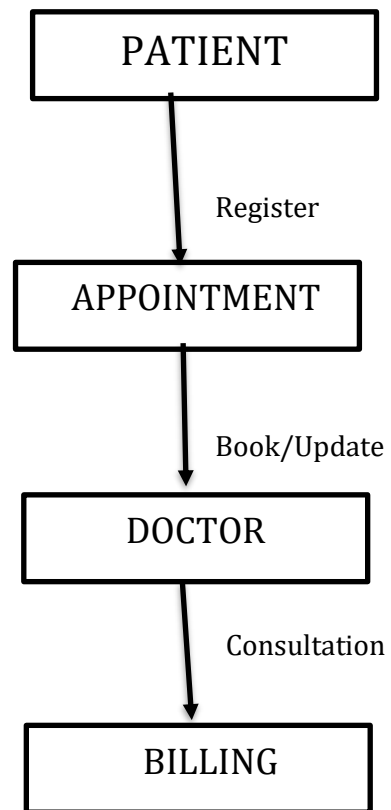DFD shows the flow of data from user to system modules .

Level 0:

Level 1:

6. Complete Structure

Process Logical Diagram

```
            ┌─────────────────────┐
            │      PATIENT        │
            └─────────────────────┘
                     │
                     ▼  Register
            ┌─────────────────────┐
            │    APPOINTMENT      │
            └─────────────────────┘
                     │
                     ▼  Book/Update
            ┌─────────────────────┐
            │      DOCTOR         │
            └─────────────────────┘
                     │
                     ▼  Consultation
            ┌─────────────────────┐
            │      BILLING        │
            └─────────────────────┘
```

7. Platform Used

Hardware Requirements

• Processor:Apple M4 chip

• RAM: 16GB or above

• Storage: Minimum 256GB space

• Operating System: MAC OS.

Software Requirements

1. Java Development Kit (JDK)

- The Java Development Kit provides all the tools required to develop and run Java applications.

- It includes the Java compiler (javac), Java Runtime Environment (JRE), and essential libraries.

- In this project, JDK is used to write and execute the backend logic, forms, and functionalities of the Hospital Management System.

    Why used in this project?

    - ✓ To develop the complete application using Core Java
    - ✓ To compile and run Java programs
    - ✓ Provides libraries for OOP-based development

2. Eclipse IDE

- Eclipse IDE is a popular development environment used for writing, editing, and debugging Java code.
- It offers features like auto-completion, error checking, project structure management, and plugin support.

Why used in this project?

- ✓ Makes Java coding easier with syntax support

✓ Integrated console and debugging tools

✓ Allows well-organized project structure

✓ Supports Maven and plugin extensions

3. Maven (Build Automation Tool)

- Maven is a dependency and project management tool.
  It automatically downloads required libraries (like MySQL Connector) and manages project versions and builds.

Why used in this project?

✓ Simplifies dependency management

✓ Easy project compilation and packaging

✓ Reduces manual configuration

✓ Creates a clean and structured Java project

4. MySQL Database

- MySQL is a relational database system used for storing all library data permanently.
- It stores doctors, patients, book appointment, and billing in tables and allows fast data retrieval.

Why used in this project?

✓ Secure and reliable data storage

✓ Supports complex queries

✓ Easy integration with Java

✓ Ensures data consistency and accuracy

5. MySQL Connector/J (JDBC Driver)

- MySQL Connector/J is the official JDBC driver used to connect Java applications to a MySQL database.

- It allows Java programs to execute SQL queries such as insert, update, delete, and fetch operations.

Why used in this project?

✓ Enables communication between Java program and MySQL database

✓ Sends queries and retrieves results

✓ Essential for performing book issue, return, and management operations

8. Future Scope

• Online appointment booking portal.

• SMS/Email reminders.

• Mobile app for patients.

• Digital medical reports.

• Full automation with RFID & AI diagnosis support.

9. Bibliography
• Java Documentation.
• MySQL Documentation.
• Online Tutorials and Reference Materials like ChatGPT.