

**PROJECT PROPOSAL**  
**ON**  
**Hospital Management System**

**Guided by:-**Mr. Anuj Kumar

**Created by:-**Anupam Tiwari

**AFid:-** AF04992195

**Student Name :-** Aakash

**AFid:-**AF05009017

**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 is a simple and smart software that helps a hospital manage its daily work in an organized way. It keeps important information in one place such as patient details, doctor information, appointments, admissions , and billing. This system reduces paperwork, saves time, avoids mistakes, and makes the entire hospital process faster and smoother.

**Objective:-**

- To design a simple, fast, and secure command-line application.
- To eliminate manual paperwork in hospitals.
- To maintain systematic records of patients and doctors.
- To store appointment details and track availability.
- To support patient admissions and billing operations.
- To provide data consistency using JDBC + MySQL.
- To demonstrate layered architecture in a Java project.

**Project Category:-** It is a Command-Line Hospital Management System created using Core Java, JDBC, and MySQL, focusing on CRUD operations, data storage, appointments, admissions, and billing.

# **ANALYSIS:**

## **A. Modules and Their Description**

### **1. Patient Management Module**

Manages details of all patients.

Functions:

Add patient

View all patients

Store medical problem description

Maintain contact & address details

### **2. Doctor Management Module**

Stores details of hospital doctors.

Functions:

Add doctor

View all doctors

Save specialization & fees

### **3. Appointment Management Module**

Handles appointments between patients & doctors.

Functions:

Book appointment

View all appointments

Store date & medical issue

## 4. Admission Module

Maintains inpatient admission records.

Functions:

Admit patient

Track room type

Maintain admit/discharge dates

## 5. Billing Module

Handles billing operations for patients.

Functions:

Generate bill

Store paid, total & balance amounts

## B. Database Design

### 1.PATIENT TABLE

| Field Name          | Data Type     | Description               | Constraints                |
|---------------------|---------------|---------------------------|----------------------------|
| patient_id          | INT           | Unique ID of patient      | Primary Key, AutoIncrement |
| name                | VARCHAR(100 ) | Patient full name         | Not Null                   |
| age                 | INT           | Patient age               | Not Null                   |
| gender              | VARCHAR(10)   | Male/Female/Other         | Not Null                   |
| mobile              | BIGINT        | Patient contact number    | Not Null                   |
| address             | VARCHAR(255)  | Home address              | Not Null                   |
| problem_description | VARCHAR(255)  | Patient's medical problem | Not Null                   |

### 2.DOCTOR TABLE

| Field Name     | Data Type    | Description         | Constraints               |
|----------------|--------------|---------------------|---------------------------|
| doctor_id      | INT          | Unique ID of doctor | Primary Key,AutoIncrement |
| name           | VARCHAR(100) | Doctor name         | Not Null                  |
| specialization | VARCHAR(100) | Field of expertise  | Not Null                  |
| mobile         | BIGINT       | Contact number      | Not Null                  |
| fees           | INT          | Consultation fees   | Not Null                  |

### 3.APPOINTMENT TABLE

| Field Name       | Data Type    | Description                          | Constraints                |
|------------------|--------------|--------------------------------------|----------------------------|
| appointment_id   | INT          | Appointment ID                       | Primary Key, Autoincrement |
| patient_id       | INT          | Linked patient                       | Foreign Key                |
| doctor_id        | INT          | Linked doctor                        | Foreign Key                |
| appointment_date | DATE         | Appointment Date                     | Not Null                   |
| problem          | VARCHAR(255) | Problem mentioned during appointment | Optional                   |

### 4.ADMISSION TABLE

| Field Name     | Data Type   | Description         | Constraints |
|----------------|-------------|---------------------|-------------|
| admission_id   | INT         | ID of admission     | Primary Key |
| patient_id     | INT         | Admitted patient    | Foreign Key |
| room_type      | VARCHAR(50) | ICU/Private/General | Not Null    |
| admit_date     | DATE        | Date of admission   | Not Null    |
| discharge_date | DATE        | Leaving date        | Nullable    |

### 5.BILLING TABLE

| Field Name   | Data Type | Description          | Constraints |
|--------------|-----------|----------------------|-------------|
| bill_id      | INT       | Bill number          | Primary Key |
| patient_id   | INT       | Related patient      | Foreign Key |
| total_amount | INT       | Total charges        | Not Null    |
| paid_amount  | INT       | Amount paid          | Not Null    |
| balance      | INT       | Remaining amount     | Calculated  |
| bill_date    | DATE      | Bill generation date | Not Null    |

## C.ER DIAGRAM

The ER diagram displays all key entities and their essential attributes.

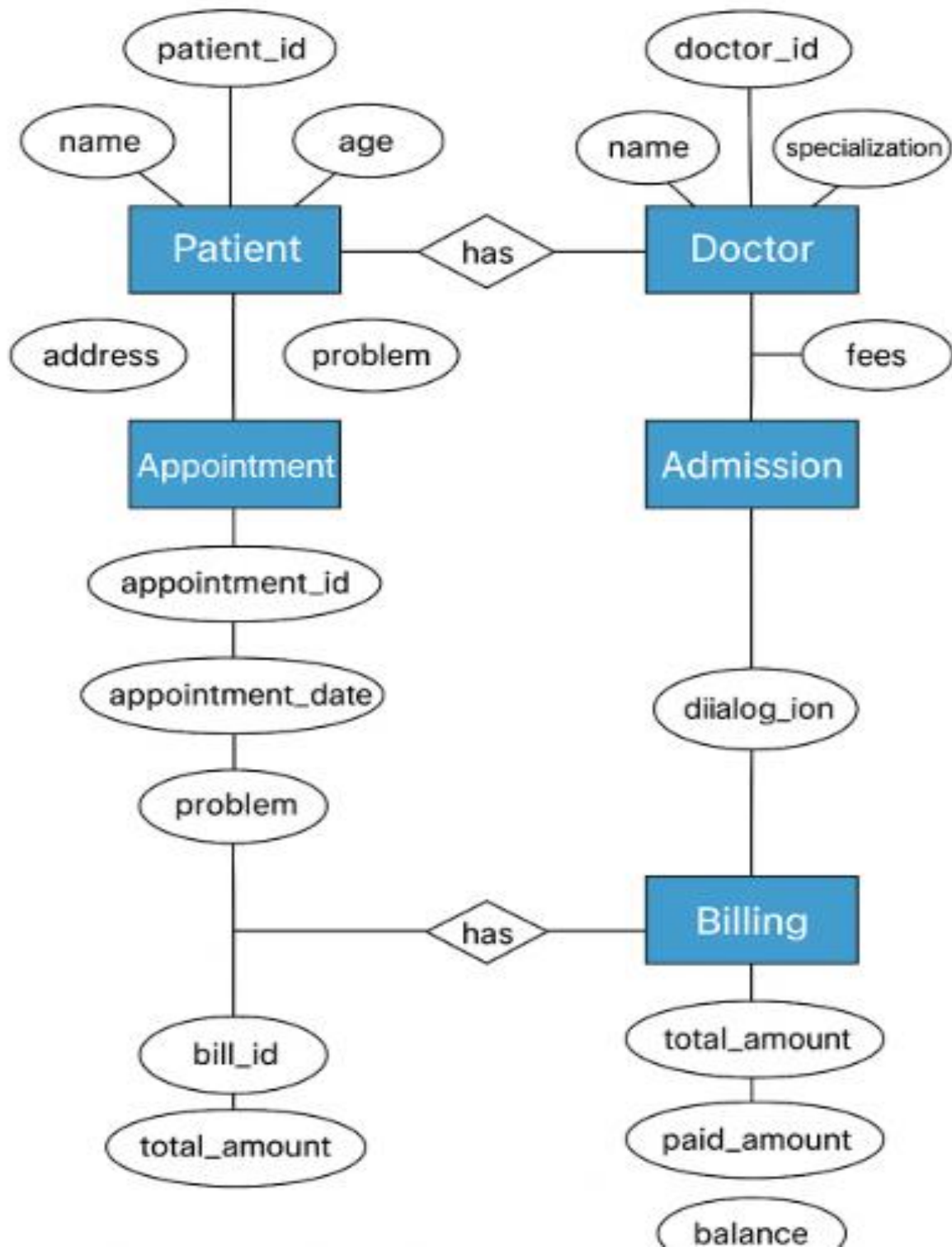
It clearly defines the relationships between different database tables.

Primary and foreign keys are used to maintain data accuracy and linkage.

It acts as the structural blueprint for creating the complete database.

### Descriptions:

- A patient can have many appointments.
- A doctor can attend many appointments.
- A patient can have multiple admissions.
- A patient can have multiple bills.





## Data Flow Diagram:-

**DFD LEVEL-0:-** The Level-0 Data Flow Diagram (DFD), also called the **Context Diagram**, represents the **entire Hospital Management System (HMS) as a single process**. It shows how the system interacts with external entities and how data flows between them.

At this level, **internal processes are not shown**. Only **inputs and outputs** between the system and external entities are represented.

### Purpose of Level-0 DFD:-

- Provides a **high-level view** of the entire system
- Shows **system boundaries**
- Identifies **external entities** interacting with the system
- Helps understand **overall data flow**
- Acts as a **base for Level-1 and Level-2 DFDs**

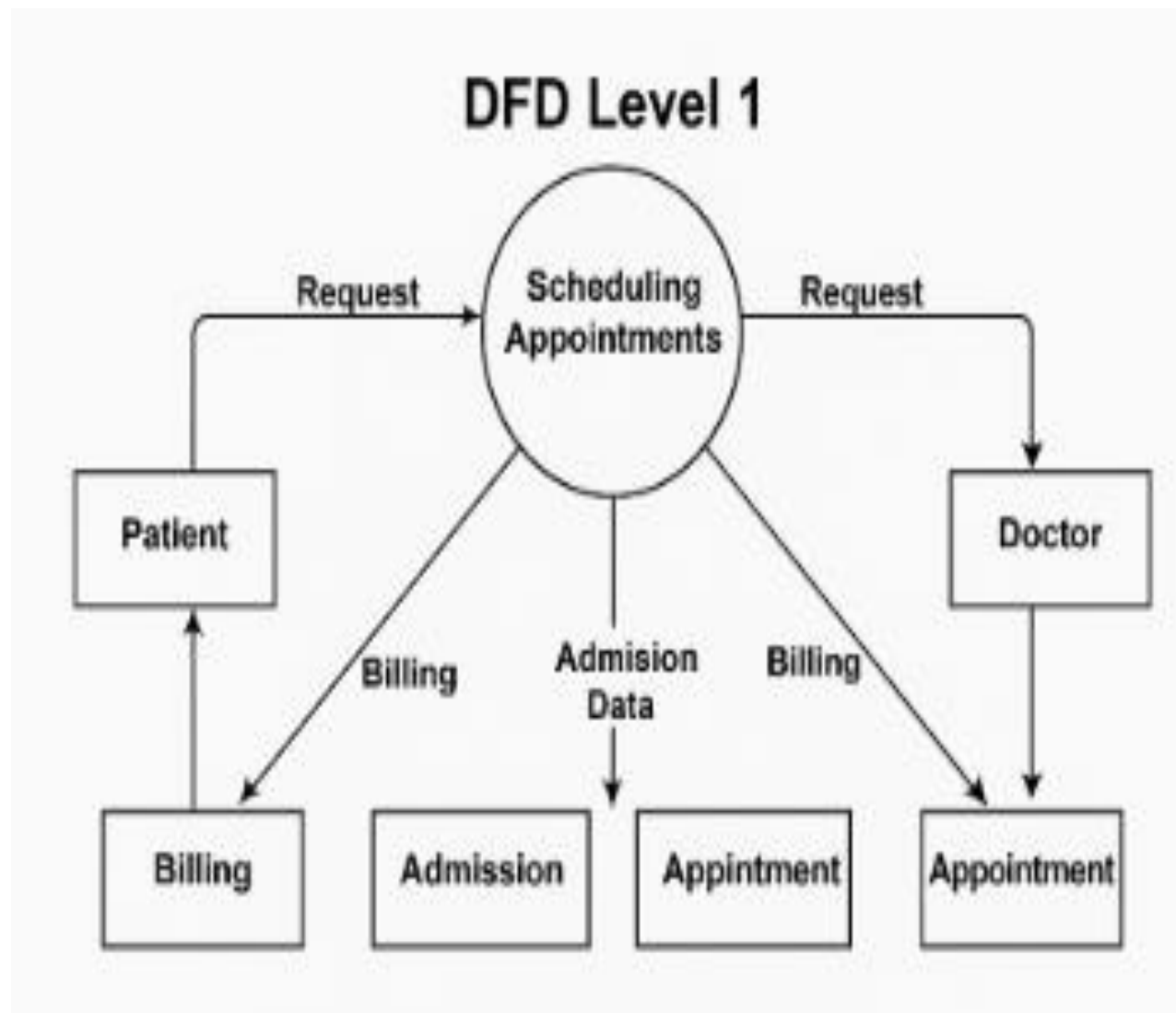
## Zero level DFD



**DFD LEVEL-1:-** The **Level-1 Data Flow Diagram (DFD)** is an expansion of the Level-0 diagram. It breaks the **Hospital Management System** into its **major functional modules** and shows how data flows between external entities, processes, and data stores

**Purpose of Level-1 DFD:-**

- Shows **major system functions**
- Explains how data moves between processes and data stores
- Helps understand **logical flow of information**
- Acts as a foundation for **Level-2 DFDs**

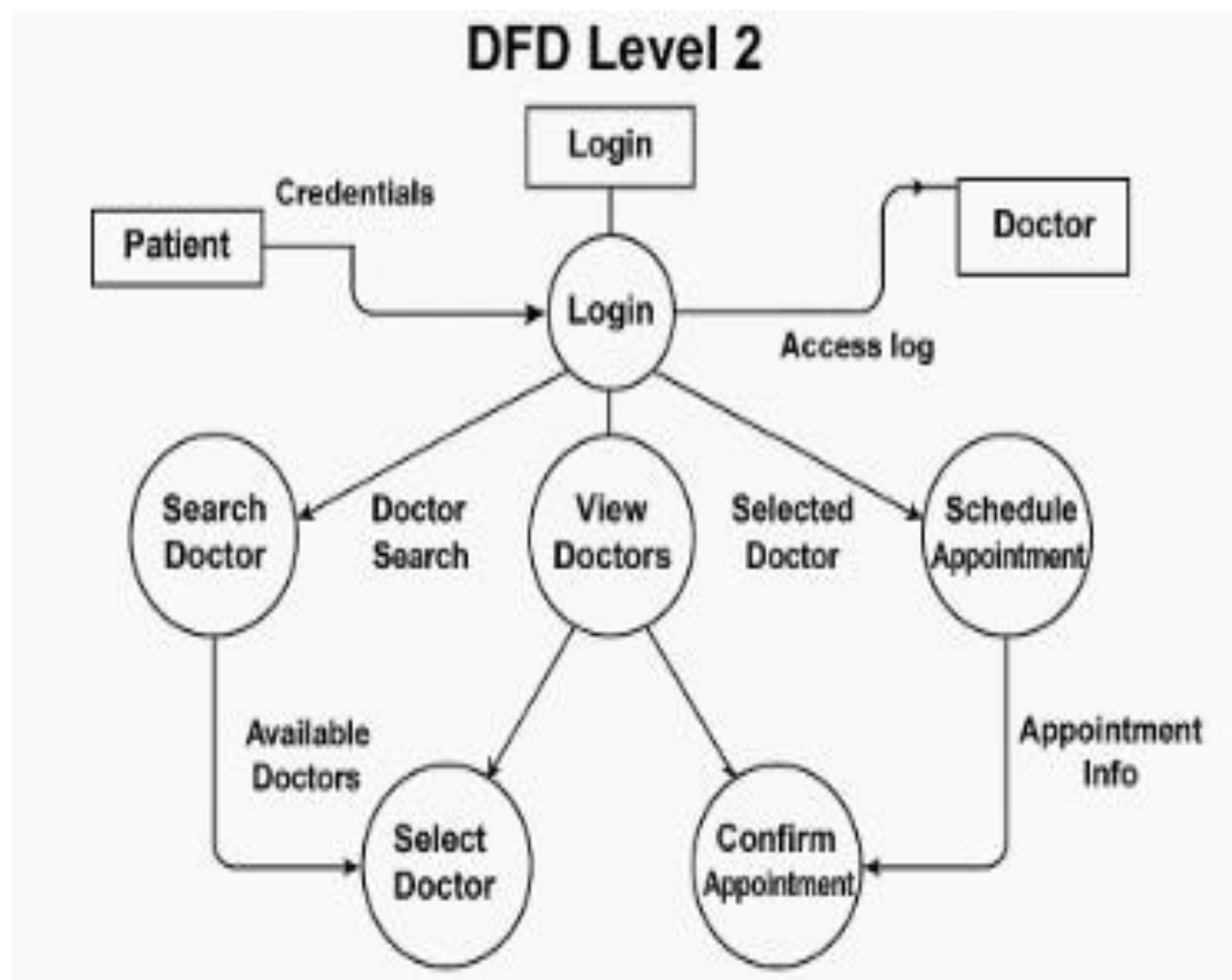


**DFD LEVEL–2:-** The Level-2 Data Flow Diagram (DFD) provides a **detailed decomposition** of the Level-1 processes. Each major module of the Hospital Management System is broken down into **sub-processes** to show how data is processed, stored, and retrieved in detail.

This level helps in understanding the **internal working of each module** and shows how the system ensures **data accuracy, consistency, and efficiency**.

### **Purpose of Level-2 DFD:-**

- Shows **detailed internal processes**
- Explains **how each module functions**
- Identifies **data storage and retrieval points**
- Helps developers during **database and module design**
- Ensures **system reliability and accuracy**



## **Complete Structure:-**

- Patient registers
- Patient books appointment
- Doctor consults patient
- Patient gets admitted (if required)
- Room is allotted
- Treatment is provided
- Bill is generated
- Payment is done
- Patient is discharged

## **Platform Used:-**

### **Hardware Requirement:-**

- Processor: Intel Core i3 or above
- RAM: Minimum 4 GB
- Hard Disk: 500 GB or above
- Keyboard
- Mouse
- Monitor

### **Software Requirement:-**

- Operating System: Windows 10 or above
- Programming Language: Java
- Database: MySQL
- IDE: Eclipse / IntelliJ IDEA
- JDBC Driver

### **Future Scope:-**

- Online appointment booking.
- SMS alerts for reminders and updates.
- Separate login panels for admin, doctor, and staff.
- Pharmacy and lab test modules.
- Online payment and e-billing support.
- Cloud-based backup and data security

### **Bibliography:-**

- Java Programming Documentation – Oracle
- MySQL Reference Manual
- JDBC API Documentation
- Online tutorials and reference materials