

HOSPITAL MANAGEMENT SYSTEM

**A
SYNOPSIS REPORT ON
COMPLETION
OF
HOSPITAL MANAGEMENT SYSTEM
IN**



SUBMITTED BY:-

AAYUSH KUMAR

SAURABH OJHA

SUBMITTED TO:-

ANUJ KUMAR



ABSTRACT

In today's fast-changing world of healthcare, it's really important to manage hospitals well. This summary talks about a Hospital Management System (HMS), which is like a smart computer program that helps hospitals with many things like keeping track of patients, scheduling appointments, managing money, handling supplies, and looking after staff.

The HMS brings all these tasks together in one place, making things easier for hospital staff. For example, it keeps all patient records organized and easy to find, which helps doctors and nurses make better decisions about treatments. It also helps patients by allowing them to book appointments online easily.

Another useful thing about the HMS is how it helps with money matters. It makes billing and paying for hospital services smoother, reducing mistakes and making sure everyone gets paid correctly. It also helps with managing hospital supplies so that there's always enough of what's needed, but not too much to waste.

Lastly, the HMS helps with managing hospital staff by handling things like work schedules, paying salaries, and checking how well everyone is doing their jobs. By taking care of these tasks, it helps the hospital run smoothly and keeps everyone focused on providing the best care possible to patients.

In short, a Hospital Management System is like a helpful friend to hospitals, making sure everything runs smoothly so that patients get the best care possible.

OVERVIEW

Hospitals are like big teams working together to help people feel better when they're sick or hurt. But keeping everything running smoothly in a hospital can be tricky with so many patients, doctors, and jobs to manage. That's where a special tool called a Hospital Management System, or HMS for short, comes in to help.

The HMS is like a super-smart computer program designed to make life easier for everyone in the hospital. It helps keep track of patients, schedules appointments, manages money, handles supplies, and takes care of the hospital's staff. It's like having a helpful assistant that never gets tired or makes mistakes.

In this paper, we'll take a closer look at how an HMS works and how it helps hospitals do their job better. We'll explore the different tasks it can do and how it makes life easier for both hospital staff and patients. So, let's dive in and discover how this amazing tool is changing the way hospitals work for the better.

Hospital Management System is used to manage all the the hospital operations such as add,view or delete patients, doctors and staff and the bill associated with them.

REQUIREMENTS

Language : JAVA

Framework : Hibernate

IDE : INTELLIJ IDEA

SYSTEM REQUIREMENTS : 512MB RAM 2GB atleast ROM

ENTITY

The Project include following entities:

1. Patient
2. Doctor
3. Staff
4. Bill
5. Payment

Patients

- **Attribues**

1. p_id (Primary Key)
2. fname
3. lname
4. gender
5. disease
6. admitstatus
7. age
8. phone_no

- **Relationship**

1. Each patient can have only one billing record, and each billing record is associated with exactly one patient. (One-to-One)
2. Each patient can have multiple payment records, but each payment record is associated with exactly one patient.(One-to-Many)

```
mysql> desc patient;
```

Field	Type	Null	Key	Default	Extra
p_id	varchar(255)	NO	PRI	NULL	
admitstatus	varchar(255)	YES		NULL	
age	int	YES		NULL	
disease	varchar(255)	YES		NULL	
fname	varchar(255)	YES		NULL	
gender	varchar(255)	YES		NULL	
lname	varchar(255)	YES		NULL	
phone_no	varchar(255)	YES		NULL	

8 rows in set (0.00 sec)

Doctors

- **Attribues**

1. d_id (Primary Key)
2. d_name
3. qualification
4. specilization
5. availability

- **Relationship**

1. A doctor can be assigned to multiple patients, and a patient can be attended by multiple doctors. This relationship is represented by the doctors and patients tables.(Many-to-Many)

Field	Type	Null	Key	Default	Extra
d_id	varchar(255)	NO	PRI	NULL	
availability	varchar(255)	YES		NULL	
d_name	varchar(255)	YES		NULL	
qualification	varchar(255)	YES		NULL	
specialization	varchar(255)	YES		NULL	

5 rows in set (0.00 sec)

Staff

- **Attribues**

1. staffId (Primary Key)
2. fname
3. lname
4. gender
5. designation
6. age
7. salary

- **Relationship**

1. Staff members (like nurses or administrators) can be associated with multiple patients, and a patient can interact with multiple staff members. This relationship is represented by the staff and patients tables.(Many-to-Many)
2. A staff member can work with multiple doctors, and a doctor can have interactions with multiple staff members.(Many-to-Many)

```
mysql> desc staff;
```

Field	Type	Null	Key	Default	Extra
staffId	varchar(255)	NO	PRI	NULL	
age	int	NO		NULL	
designation	varchar(255)	YES		NULL	
firstName	varchar(255)	YES		NULL	
gender	varchar(255)	YES		NULL	
lastName	varchar(255)	YES		NULL	
salary	double	NO		NULL	

```
7 rows in set (0.00 sec)
```

Bill

- **Attribues**

1. b_id(Primary Key)
2. basicCharges
3. additionalCharges
4. medicationCharges
5. roomCharges
6. doctorFees
7. totalCharges
8. patient_id(Foreign Key)

- **Relationship**

1. Each billing record is associated with a staff member who handles the billing process. This relationship ensures that a billing record has a responsible staff member.(One-to-One)

```
mysql> desc bill;
```

Field	Type	Null	Key	Default	Extra
billId	varchar(255)	NO	PRI	NULL	
additionalCharges	double	NO		NULL	
basicCharges	double	NO		NULL	
doctorFees	double	NO		NULL	
medicationCharges	double	NO		NULL	
roomCharges	double	NO		NULL	
totalCharges	double	NO		NULL	
patient_id	varchar(255)	YES	MUL	NULL	
staff_id	varchar(255)	YES	MUL	NULL	

```
9 rows in set (0.00 sec)
```

Payment

- **Attributes**

1. p_id (Primary Key)
2. bill_id (Foreign Key)
3. paymentMethod
4. amount

- **Relationship**

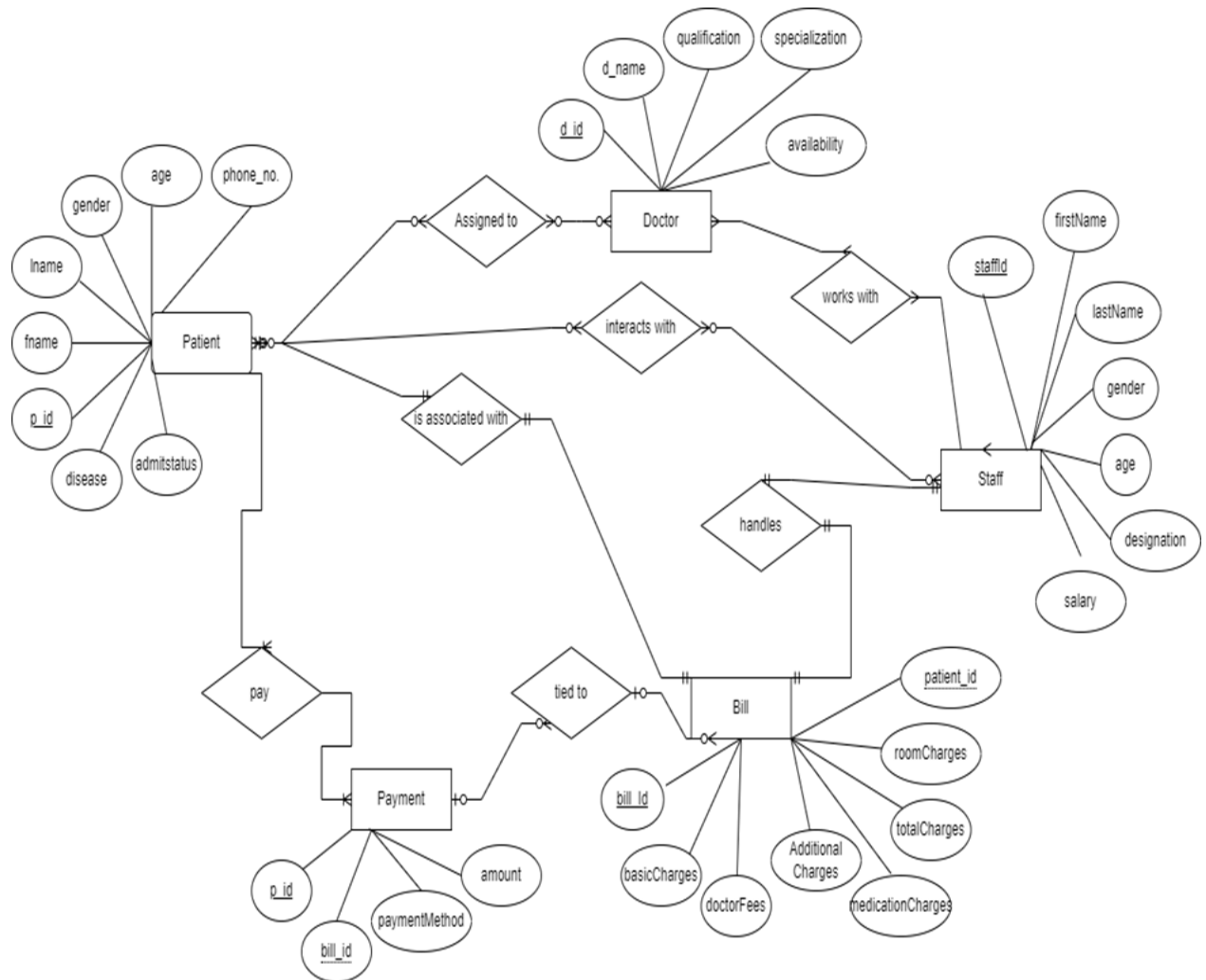
- a. many payment records is tied with one record in the Billis table.(Many- to-One)

```
mysql> desc payment;
```

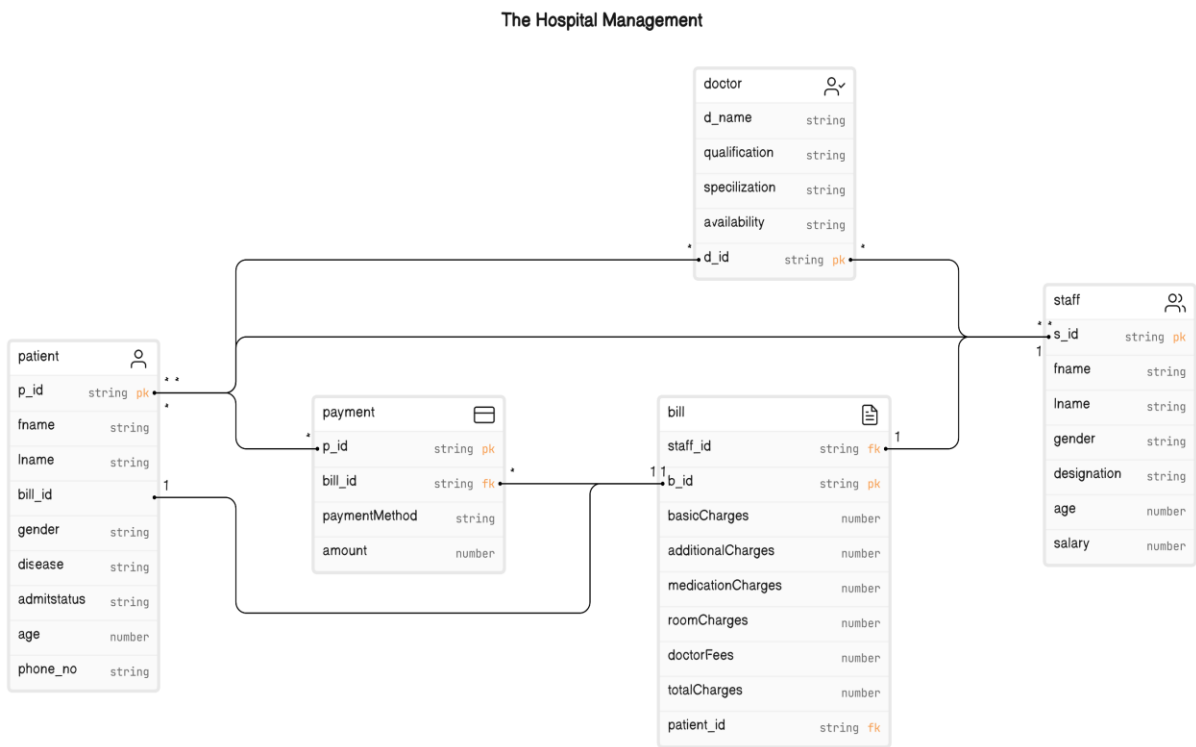
Field	Type	Null	Key	Default	Extra
paymentId	varchar(255)	NO	PRI	NULL	
amount	double	NO		NULL	
paymentMethod	varchar(255)	YES		NULL	
billId	varchar(255)	YES	MUL	NULL	

```
4 rows in set (0.00 sec)
```

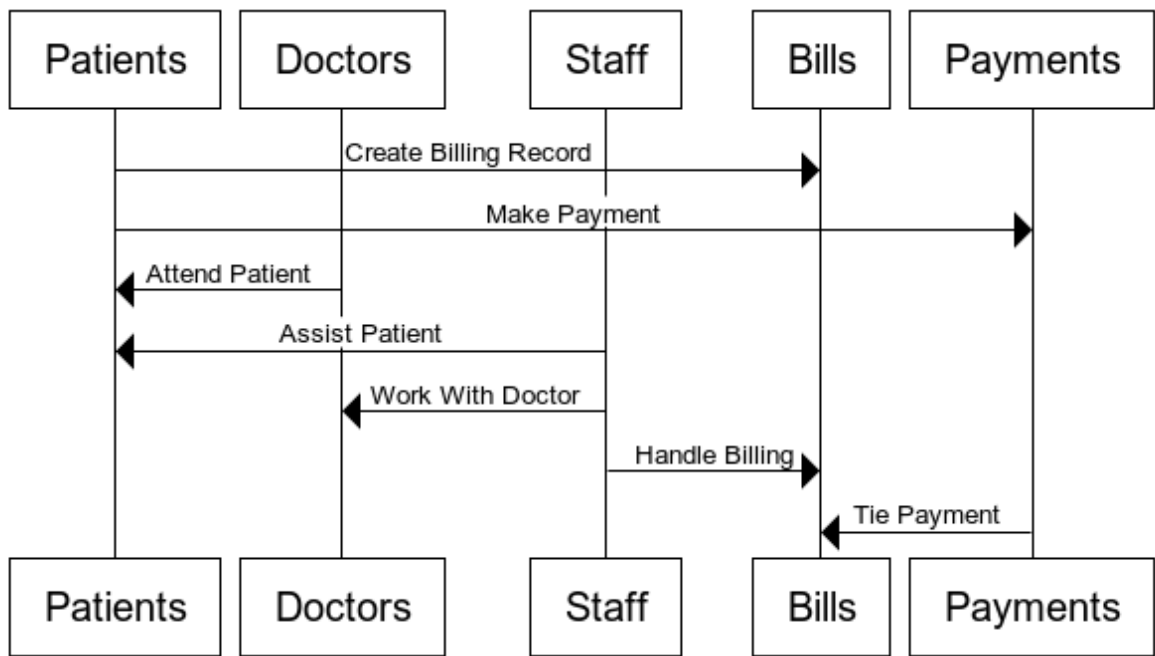
ER DIAGRAM



CLASS DIAGRAM



SEQUENCE DIAGRAM



OUTPUT

MAIN

Hospital Management System

1. Manage Patients
2. Manage Doctors
3. Manage Staff
4. Manage Bills
5. Manage Payments
6. Exit

PATIENT MENU

Patient Menu

1. Add Patient
2. Update Patient
3. Delete Patient
4. View Patient Details
5. View All Patient Details
6. Back to Main Menu

DOCTOR MENU

Doctor Menu

1. Add Doctor
2. Update Doctor
3. Delete Doctor
4. View Doctor Details By ID
5. View All Doctor Details
6. Back to Main Menu

STAFF MENU

Staff Menu

1. Add Staff
2. Update Staff
3. Delete Staff
4. View Staff Details By ID
5. View All Staff Details
6. Back to Main Menu

BILL MENU

Bill Menu

1. Add Bill
2. Update Bill
3. Delete Bill
4. View Bill Details
5. Back to Main Menu

PAYMENT MENU

Payment Menu

1. Add Payment
2. Update Payment
3. Delete Payment
4. View Payment Details By ID
5. View All Payment Details
6. Back to Main Menu

SUMMARY

Hospitals are bustling communities with numerous patients, doctors, and tasks to handle. To streamline operations, Hospital Management Systems (HMS) act as intelligent assistants, aiding in patient management, appointment scheduling, financial transactions, inventory control, and staff administration. This paper explores the functionalities and benefits of an HMS, focusing on its role in enhancing hospital efficiency and patient care.

The requirements for developing an HMS include using Java language, Hibernate framework, and IntelliJ IDEA as the Integrated Development Environment (IDE). The system's minimum system requirements are 512MB RAM and at least 2GB of ROM.

Entities within the HMS encompass Patients, Doctors, Staff, Bills, and Payments. Patients' attributes include personal details, disease status, and admission status. The relationship between patients and billing records is one-to-one, while patients can have multiple payment records. Doctors' attributes consist of name, qualifications, specialization, and availability, with a many-to-many relationship between doctors and patients. Staff attributes include personal details and designation, with staff-patient and staff-doctor relationships established as many-to-many.

The Bill entity records various charges associated with patient care, while Payments track payment details, both tied to patients and bills. The ER Diagram, Class Diagram, and Sequence Diagram visually represent the relationships and interactions between these entities.

The output of the HMS includes menus for Patient, Doctor, Staff, Bill, and Payment management, ensuring smooth navigation and user-friendly interaction with the system. Through these menus, hospital staff can efficiently manage various aspects of hospital operations, ultimately leading to improved patient care and organizational effectiveness.