

COMSATS UNIVERSITY ISLAMABAD, ABBOTTABAD CAMPUS

DBMS Semester Project

Hospital Management System

Submitted By:

Hanzla Nouman FA21-BSE-015-4A

Laiba Binta Tahir FA21-BSE-019-4A

Muhammad Irfan FA21-BSE-027-4A

Arfah Ali FA21-BSE-080-4A

Submitted To:

Dr. Rab Nawaz jadoon

Dated: 10 July 2023

Abstract:

The Hospital Management System is a comprehensive solution designed to address the challenges faced by hospitals in managing various aspects of their operations, including appointment booking, patient records management, and maintaining medical history. This project aims to develop an efficient and user-friendly system by leveraging modern technologies.

The frontend of the system is built using ReactJS and JavaScript, providing a seamless and interactive user experience for patients, doctors, and administrators. The backend is developed in Node.js, ensuring a robust and scalable infrastructure to handle the system's functionalities. The data is stored in a MySQL database, allowing for efficient and secure management of patient information.

The system's primary objective is to simplify the appointment booking process for patients while providing doctors and administrators with a comprehensive platform to manage patient data effectively. Patients can easily book appointments, view their medical history, and update their personal information. Doctors and administrators have access to patient records, enabling them to provide accurate diagnoses, prescribe medications, and maintain detailed medical histories.

By utilizing a well-designed user interface and efficient data management techniques, the Hospital Management System aims to enhance the overall efficiency of hospital operations. It enables hospitals to handle a large volume of patient data securely, facilitating seamless communication between patients, doctors, and administrators.

The successful implementation of the Hospital Management System will empower hospitals to improve their overall workflow, enhance patient care, and streamline administrative tasks. It provides a foundation for efficient data management and ensures a more seamless experience for all stakeholders involved in the healthcare process.

Table Of Contents

[Hospital Management System 6](#_Toc139797261)

[1. Introduction 6](#_Toc139797262)

[2. System Architecture 6](#_Toc139797263)

[3. Database Design 6](#_Toc139797264)

[4. Functionality 8](#_Toc139797265)

[4.1 Patient Side Features: 8](#_Toc139797266)

[4.2 Doctor Side Features: 8](#_Toc139797267)

[5. Implementation 8](#_Toc139797268)

[6. Testing 9](#_Toc139797269)

[7. Deployment and User Documentation 9](#_Toc139797270)

[8. Challenges and Lessons Learned 9](#_Toc139797271)

[9. Conclusion 10](#_Toc139797272)

[10. DDL statements 10](#_Toc139797273)

List Of Images

Figure 1 ER Diagram for ..............................................………………………………………07

Figure 2 Relational Schemas for HMS………………………………………….……………07

Glossary

|  |  |
| --- | --- |
| HMS | Hospital management system |
| ER Diagram | Entity Relationship diagram |
| DBMS | Database management system |
| DDL | Data definition language |

# Hospital Management System

## Introduction

- The Hospital Management System is a project developed for a Semester 4 DBMS course.

- The aim of the project is to demonstrate the management of various activities involved in the day-to-day operations of hospitals using a database.

- The project focuses on activities such as appointment booking, doctor scheduling, patient diagnoses, and medical history management.

## 2. System Architecture

- The Hospital Management System follows a client-server architecture.

- The frontend is developed using React.js, while the backend is built with Node.js and Express.

- The MySQL database is used to store the system's data.

- The system utilizes an Entity-Relationship (ER) diagram to model the relationships between different entities.

## 3. Database Design

- The database design for the system is based on the provided ER diagram.

- It includes tables for patients, doctors, appointments, medical history, diagnoses, and prescriptions.

- The schema design ensures data integrity and maintains relationships between entities.

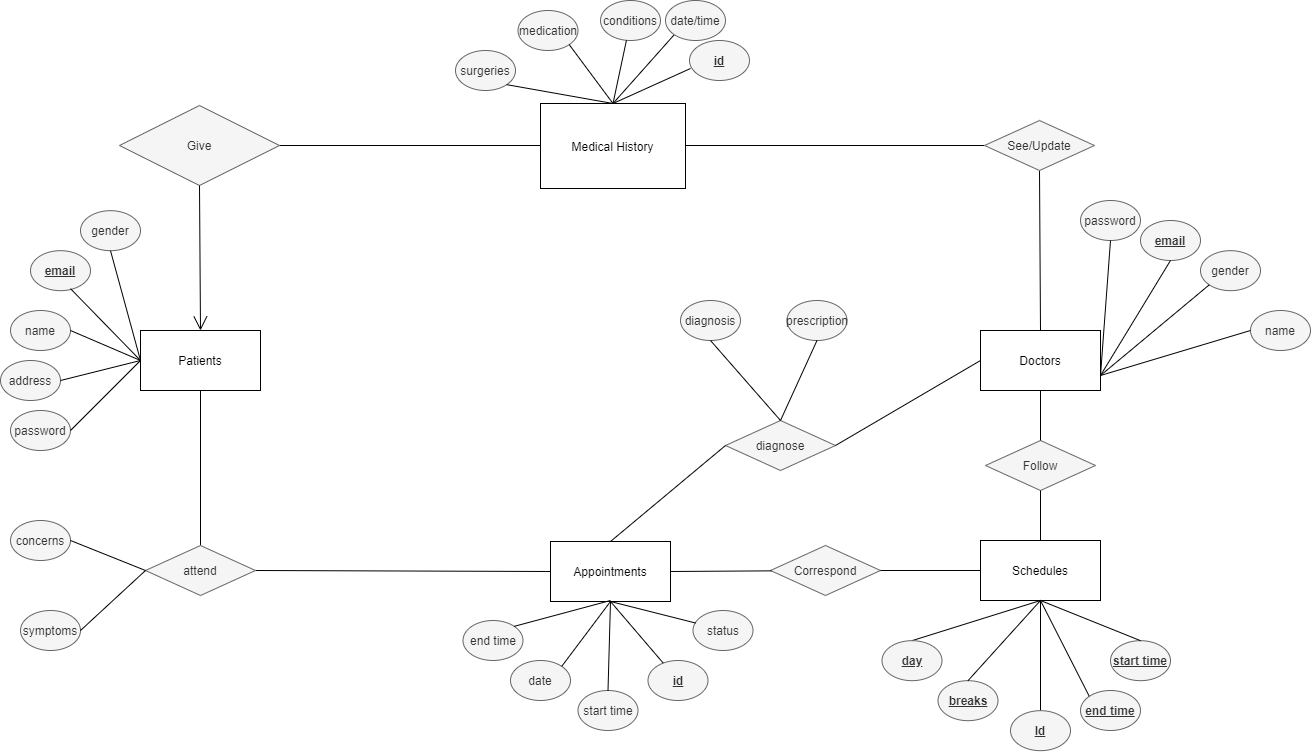


Figure 1 ER Diagram for HMS

A diagram of a work flow

Description automatically generated

Figure 2 Relational Schemas for HMS

## 4. Functionality

### 4.1 Patient Side Features:

- Patients have a separate interface and can log in to the system.

- Patients can book appointments with doctors.

- They can provide their previous medical history.

- Patients have the ability to view, update, or cancel their already booked appointments.

- Cancelled appointments create available slots for other patients.

- The system prevents appointment clashes and ensures each patient gets their dedicated slot.

- Patients can view their complete diagnosis, prescriptions, and medical history.

- Patient medical history is accessible only to the doctor with whom the appointment is booked, ensuring privacy.

### 4.2 Doctor Side Features:

- Doctors have a separate interface and can log in to the system.

- The system considers doctor schedules and prevents appointment bookings when a doctor is busy or on a break.

- Doctors can access patient history and profiles, and add to the patient's medical history.

- They can provide diagnoses and prescriptions to patients.

- Doctors have the ability to modify existing diagnoses and prescriptions.

## 5. Implementation

- The Hospital Management System is implemented using React.js for the frontend.

- The backend is developed with Node.js and Express, providing the necessary APIs for data retrieval and manipulation.

- The system follows a software development methodology (e.g., Agile, Waterfall) to ensure organized development.

- Notable design patterns or architectural choices are implemented for modularity and scalability.

## 6. Testing

- The system undergoes various testing phases, including unit testing, integration testing, and acceptance testing.

- Testing frameworks and tools are utilized to ensure the system's functionality and reliability.

- The results of the testing phase are documented, and any issues encountered are addressed and resolved.

## 7. Deployment and User Documentation

- Instructions are provided for installing and configuring the system.

- User documentation and manuals are created to guide hospital staff on system usage.

## 8. Challenges and Lessons Learned

- The project development process may have encountered challenges, such as technical difficulties or design complexities.

- Each challenge is discussed, along with the strategies employed to overcome them.

- Valuable lessons learned during the development process are reflected upon, along with suggestions for future improvements.

## 9. Conclusion

- The Hospital Management System successfully demonstrates the management of hospital activities using a database.

- The system provides convenient features for patients and doctors, ensuring efficient appointment scheduling, medical history management, and diagnosis and prescription processes.

- Future enhancements or expansions that can be considered for the system are discussed.

## 10. DDL statements

**Table: Patient**

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Constraints** |
| email | varchar(50) | PRIMARY KEY |
| password | varchar(30) | NOT NULL |
| name | varchar(50) | NOT NULL |
| address | varchar(60) | NOT NULL |
| gender | varchar(20) | NOT NULL |

**Table: MedicalHistory**

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Constraints** |
| id | int | PRIMARY KEY |
| date | DATE | NOT NULL |
| conditions | VARCHAR(100) | NOT NULL |
| surgeries | VARCHAR(100) | NOT NULL |
| medication | VARCHAR(100) | NOT NULL |

**Table: Doctor**

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Constraints** |
| email | varchar(50) | PRIMARY KEY |
| gender | varchar(20) | NOT NULL |
| password | varchar(30) | NOT NULL |
| name | varchar(50) | NOT NULL |

**Table: Appointment**

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Constraints** |
| id | int | PRIMARY KEY |
| date | DATE | NOT NULL |
| starttime | TIME | NOT NULL |
| endtime | TIME | NOT NULL |
| status | varchar(15) | NOT NULL |

**Table: PatientsAttendAppointments**

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Constraints** |
| patient | varchar(50) | NOT NULL |
| appt | int | NOT NULL |
| concerns | varchar(40) | NOT NULL |
| symptoms | varchar(40) | NOT NULL |

**Table: Schedule**

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Constraints** |
| id | int | NOT NULL |
| starttime | TIME | NOT NULL |
| endtime | TIME | NOT NULL |
| breaktime | TIME | NOT NULL |
| day | varchar(20) | NOT NULL |

**Table: PatientsFillHistory**

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Constraints** |
| patient | varchar(50) | NOT NULL |
| history | int | NOT NULL |

**Table: Diagnose**

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Constraints** |
| appt | int | NOT NULL |
| doctor | varchar(50) | NOT NULL |
| diagnosis | varchar(40) | NOT NULL |
| prescription | varchar(50) | NOT NULL |

**Table: DocsHaveSchedules**

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Constraints** |
| sched | int | NOT NULL |
| doctor | varchar(50) | NOT NULL |

**Table: DoctorViewsHistory**

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Constraints** |
| history | int | NOT NULL |
| doctor | varchar(50) | NOT NULL |