

[HOSPITAL MANAGEMENT SYSTEM]

-GROUP K

Project submitted to the
SRM University – AP, Andhra Pradesh
for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology/Master of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by (Group-K)

V. HARSHA VARDHAN AP22110010677)

V. MOHIT SAI SURYA (AP22110010704)

C. LOKESH (AP22110010676)

K. UMESH(AP22110010681)

J. SATWIK(AP22110010688)



Under the Guidance of

Prof. Aurobindo Behara

SRM University-AP
Neerukonda, Mangalagiri, Guntur
Andhra Pradesh - 522 240
[April, 2024]

Certificate

Date: 20-04-2024

This is to certify that the work present in this Project entitled "**HOSPITAL
MANAGEMENT SYSTEM**" been carried out by

[V. HARSHA VARDHAN, V. MOHIT SAI SURYA, C. LOKESH, K. UMESH, J. SATWIK]

under my/our supervision. The work is genuine, original, and suitable for
submission to the SRM University – AP for the award of Bachelor of
Technology/Master of Technology in **School of Engineering and Sciences**.

Supervisor

(Signature)

Prof. Aurobindo Behara

Designation,

Affiliation.

Acknowledgements

Firstly, we would like to appreciate the contribution and concern of our instructor who helped us in accomplishing this project in a systematic manner. The guidance of our instructor helped us a lot and it also keeps us motivated at every step. In this way, the encouragement given by *Prof. AUROBINDO BEHARA* helped us in achieving our objective. Therefore, we would like to thank our supervisors for the guidance and concern.

Secondly, we would like to thank our team members who guided us and helped us through his effective management, knowledge and skills in making this software a success. They motivated us at every step, which helped in boosting up our morale. Their planning and support helped us in coping up with all the hurdles we faced during the development process. We would like to mention the contribution of those people who helped us in gathering the information regarding the **HOSPITAL MANAGEMENT SYSTEM** With the help of their concern and trust, data was collected authentically.

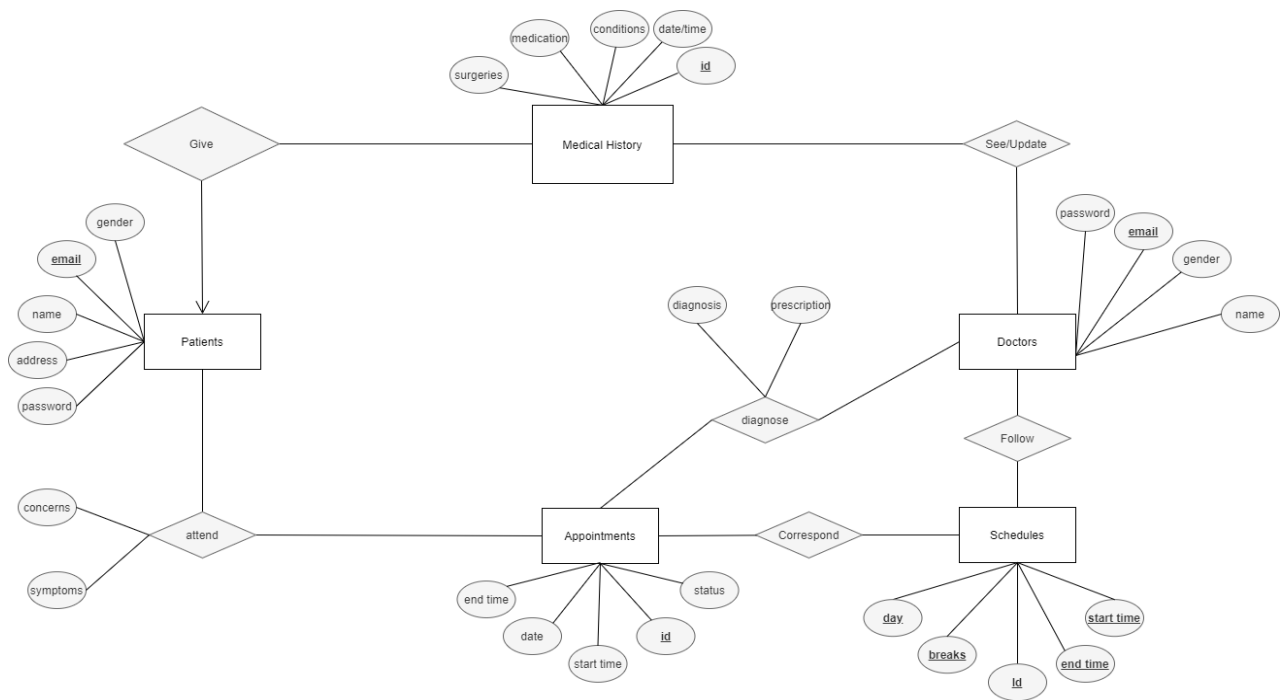
Table of Contents

Certificate.....	1
Acknowledgements.....	2
Abstract.....	4
ER Diagram.....	5
Relational Schema.....	6
List of Tables.....	7
Introduction.....	1
Methodology.....	3
Implementation.....	5
Conclusion Remark.....	11
References.....	11.

Abstract

Hospitals interact with a lot of people in a day and there are various activities involved in day-to-day operations of hospitals, for example booking of appointments, managing doctor schedules, managing patient diagnoses, managing medical histories of patients, etc. The aim of this project is to show how data related to these tasks can be made easier to manage using databases. The use of a DBMS ensures data integrity, security, and scalability, enabling the system to handle large volumes of data efficiently. The graphical user interface (GUI) provides a user-friendly experience for both administrators and end-users, improving overall productivity and patient care. Overall, the Hospital Management System Database Project aims to modernize and optimize hospital operations, leading to better healthcare delivery and patient outcomes.

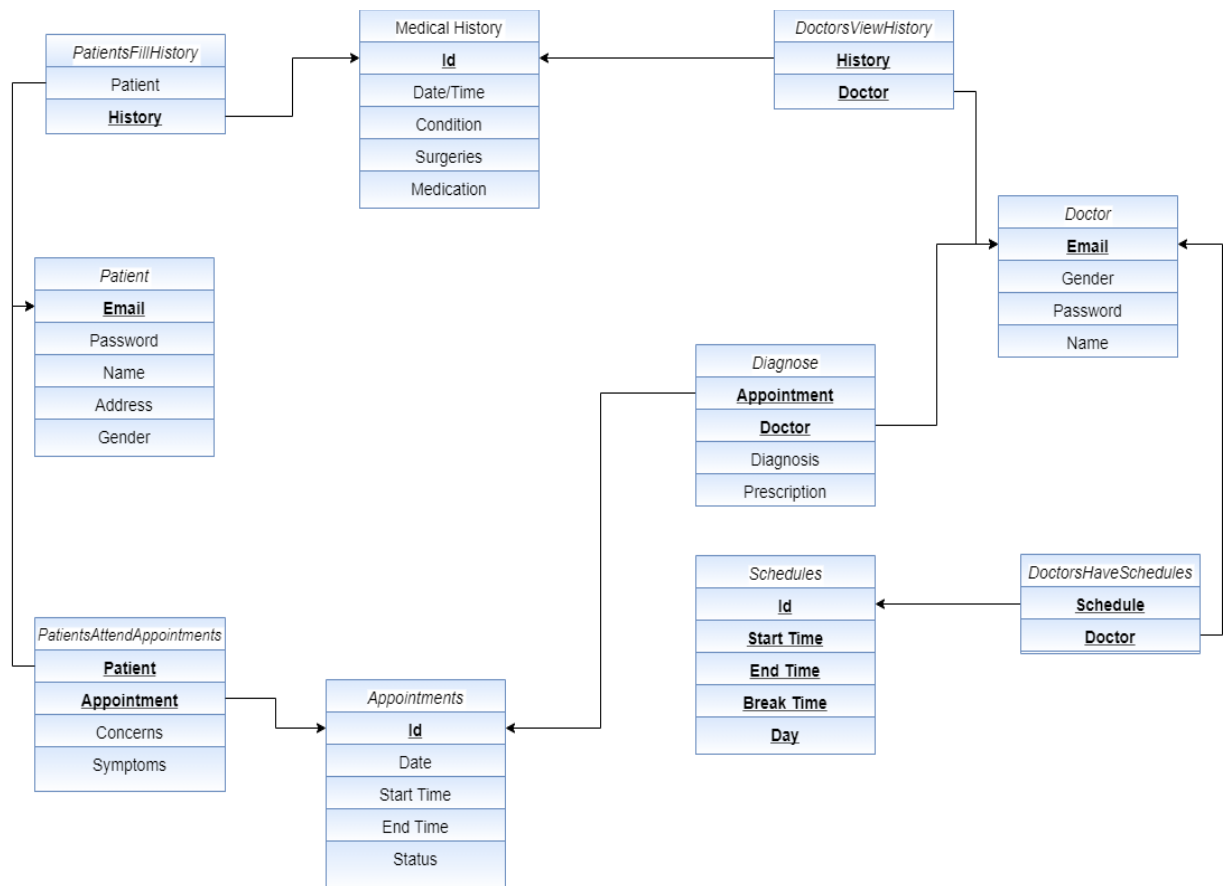
ENTITY RELATIONAL DIAGRAM



ENTITIES

– Patients, Appointments, Doctors, Schedules, Medical History

RELATIONAL SCHEMA DIAGRAM



List of Tables

Patient Table:

	email	password	name	address	gender	
	rakesh@gmail.com	hrishikesh13	Rakesh	Gujarat	male	
	ramesh@gmail.com	hrishikesh13	Ramesh	Tamil Nadu	male	
	suresh@gmail.com	hrishikesh13	Suresh	Karnataka	male	
	NULL	NULL	NULL	NULL	NULL	

Medical History Table:

	id	date	conditions	surgeries	medication	
	1	2019-01-14	Pain in abdomen	Heart Surgery	Crocin	
	2	2019-01-14	Frequent Indigestion	none	none	
	3	2019-01-14	Body Pain	none	Iodex	
	NULL	NULL	NULL	NULL	NULL	

Doctor Table:

	email	gender	password	name	
	hathalye7@gmail.com	male	hrishikesh13	Hrishikesh Athalye	
	hathalye8@gmail.com	male	hrishikesh13	Hrishikesh Athalye	
	NULL	NULL	NULL	NULL	

Appointment Table:

	id	date	starttime	endtime	status	
	1	2019-01-15	09:00:00	10:00:00	Done	
	2	2019-01-16	10:00:00	11:00:00	Done	
	3	2019-01-18	14:00:00	15:00:00	Done	
	NULL	NULL	NULL	NULL	NULL	

Patients Attend Appointments Table:

	patient	appt	concerns	symptoms	
	rakesh@gmail.com	3	nausea	fever	
	ramesh@gmail.com	1	none	itchy throat	
	suresh@gmail.com	2	infection	fever	
	NULL	NULL	NULL	NULL	

Schedule:

	id	starttime	endtime	breaktime	day	
	1	09:00:00	17:00:00	12:00:00	Friday	
	1	09:00:00	17:00:00	12:00:00	Saturday	
	1	09:00:00	17:00:00	12:00:00	Sunday	
	1	09:00:00	17:00:00	12:00:00	Tuesday	
	2	09:00:00	17:00:00	12:00:00	Friday	
	2	09:00:00	17:00:00	12:00:00	Wednesday	
	NULL	NULL	NULL	NULL	NULL	

Patients Fill History Table:

	patient	history	
	rakesh@gmail.com	3	
	ramesh@gmail.com	1	
	suresh@gmail.com	2	
	NULL	NULL	

Diagnose Table:

	appt	doctor	diagnosis	prescription	
	1	hathalye7@gmail.com	Bloating	Ibuprofen as needed	
	2	hathalye8@gmail.com	Muscle soreness	Stretch morning/night	
	3	hathalye8@gmail.com	Vitamin Deficiency	Good Diet	
	NULL	NULL	NULL	NULL	

DocsHaveSchedules Table:

	sched	doctor	
	1	hathalye7@gmail.com	
	2	hathalye8@gmail.com	
	NULL	NULL	

DoctorViewsHistory Table:

	history	doctor	
	1	hathalye7@gmail.com	
	2	hathalye8@gmail.com	
	3	hathalye8@gmail.com	
	NULL	NULL	

Introduction

This code is for a Hospital Management System (HMS) database project. It includes the creation of several tables to store information about patients, medical history, doctors, appointments, schedules, diagnoses, and interactions between patients and doctors. The data is populated with sample entries to demonstrate how the database would function in a real-world scenario. It covers aspects such as patient details, medical history, appointments, diagnoses, and doctor schedules, allowing for effective management and tracking of healthcare-related information.

Database Creation:

CREATE DATABASE HMS; This command creates a new database named HMS.

USE HMS; This command selects the HMS database for use, meaning subsequent commands will be applied to this database.

Table Creation:

CREATE TABLE Patient(...); Defines a table named Patient to store patient information such as email, password, name, address, and gender.

CREATE TABLE MedicalHistory(...); Defines a table named MedicalHistory to store medical history records including conditions, surgeries, and medications.

CREATE TABLE Doctor(...); Defines a table named Doctor to store doctor details like email, gender, password, and name.

CREATE TABLE Appointment(...); Defines a table named Appointment to store appointment information including date, start time, end time, and status.

CREATE TABLE PatientsAttendAppointments(...);: Defines a table to track which patients attend which appointments and their concerns and symptoms.

CREATE TABLE Schedule(...);: Defines a table to store doctor schedules including start time, end time, break time, and day.

CREATE TABLE PatientsFillHistory(...);: Defines a table to track which patients fill out which medical history forms.

CREATE TABLE Diagnose(...);: Defines a table to store diagnoses made during appointments along with prescriptions.

CREATE TABLE DocsHaveSchedules(...);: Defines a table to link doctors with their schedules.

CREATE TABLE DoctorViewsHistory(...);: Defines a table to track which doctors view which medical histories.

Data Insertion:

INSERT INTO ... VALUES ...;: Inserts sample data into each table to demonstrate how the system would store actual information. For example, patient details, medical histories, doctor information, appointment details, patient-doctor interactions, etc.

Project Introduction:

This code serves as the backbone for a Database Management System (DBMS) project focused on hospital management. The tables and data structures are designed to manage patient records, medical histories, appointments, doctor schedules, diagnoses, and interactions between patients and doctors effectively. It's a foundation that a software application can build upon to create a functional hospital management system, allowing for efficient data storage, retrieval, and management within a healthcare setting.

Methodology

Project Initiation:

Define the project scope, objectives, and goals.

Identify stakeholders, including hospital staff, administrators, doctors, and patients.

Set up the development team, including database designers, developers, and testers.

Requirements Gathering:

Conduct meetings with stakeholders to gather system requirements.

Define user stories or use cases to capture functional and non-functional requirements.

Prioritize requirements based on their importance and impact on the system.

Database Design:

Collaborate with database designers to design the database schema based on the gathered requirements.

Define tables, relationships, attributes, and constraints.

Ensure data integrity, normalization, and security considerations are addressed.

Iterative Development:

Adopt an iterative approach to development, breaking down the project into manageable tasks or sprints.

Use incremental development to deliver functionality in stages, allowing for feedback and adjustments.

Develop backend logic for managing patient records, appointments, medical histories, doctor schedules, and interactions.

Feedback and Iteration:

Gather feedback from stakeholders and end-users post-deployment.

Identify areas for improvement or additional features based on feedback.

Iterate on the system to incorporate enhancements and address any issues or challenges encountered.

By following an Agile methodology, the HMS project can adapt to changing requirements, promote collaboration among team members, and deliver a high-quality, user-centric solution for hospital management.

Implementation:

(DDL)

```
CREATE DATABASE HMS;  
USE HMS;
```

```
CREATE TABLE Patient(  
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  
);
```

```
CREATE TABLE MedicalHistory(  
id int PRIMARY KEY,  
date DATE NOT NULL,  
conditions VARCHAR(100) NOT NULL,  
surgeries VARCHAR(100) NOT NULL,  
medication VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE Doctor(  
email varchar(50) PRIMARY KEY,  
gender varchar(20) NOT NULL,  
password varchar(30) NOT NULL,  
name varchar(50) NOT NULL  
);
```

```
CREATE TABLE Appointment(  
id int PRIMARY KEY,  
date DATE NOT NULL,  
starttime TIME NOT NULL,  
endtime TIME NOT NULL,  
status varchar(15) NOT NULL
```

);

```
CREATE TABLE PatientsAttendAppointments(  
  patient varchar(50) NOT NULL,  
  appt int NOT NULL,  
  concerns varchar(40) NOT NULL,  
  symptoms varchar(40) NOT NULL,  
  FOREIGN KEY (patient) REFERENCES Patient (email) ON  
  DELETE CASCADE,  
  FOREIGN KEY (appt) REFERENCES Appointment (id) ON  
  DELETE CASCADE,  
  PRIMARY KEY (patient, appt)  
);
```

```
CREATE TABLE Schedule(  
  id int NOT NULL,  
  starttime TIME NOT NULL,  
  endtime TIME NOT NULL,  
  breaktime TIME NOT NULL,  
  day varchar(20) NOT NULL,  
  PRIMARY KEY (id, starttime, endtime, breaktime, day)  
);
```

```
CREATE TABLE PatientsFillHistory(  
  patient varchar(50) NOT NULL,  
  history int NOT NULL,  
  FOREIGN KEY (patient) REFERENCES Patient (email) ON  
  DELETE CASCADE,  
  FOREIGN KEY (history) REFERENCES MedicalHistory (id) ON  
  DELETE CASCADE,  
  PRIMARY KEY (history)  
);
```

```
CREATE TABLE Diagnose(  
  appt int NOT NULL,
```

```
doctor varchar(50) NOT NULL,  
diagnosis varchar(40) NOT NULL,  
prescription varchar(50) NOT NULL,  
FOREIGN KEY (appt) REFERENCES Appointment (id) ON  
DELETE CASCADE,  
FOREIGN KEY (doctor) REFERENCES Doctor (email) ON  
DELETE CASCADE,  
PRIMARY KEY (appt, doctor)  
);
```

```
CREATE TABLE DocsHaveSchedules(  
sched int NOT NULL,  
doctor varchar(50) NOT NULL,  
FOREIGN KEY (sched) REFERENCES Schedule (id) ON  
DELETE CASCADE,  
FOREIGN KEY (doctor) REFERENCES Doctor (email) ON  
DELETE CASCADE,  
PRIMARY KEY (sched, doctor)  
);
```

```
CREATE TABLE DoctorViewsHistory(  
history int NOT NULL,  
doctor varchar(50) NOT NULL,  
FOREIGN KEY (doctor) REFERENCES Doctor (email) ON  
DELETE CASCADE,  
FOREIGN KEY (history) REFERENCES MedicalHistory (id) ON  
DELETE CASCADE,  
PRIMARY KEY (history, doctor)  
);
```

(DML)

```
INSERT INTO Patient(email, password, name, address,gender)
```

VALUES

```
('ramesh@gmail.com','hrishikesh13','Ramesh','Tamil Nadu',  
'male'),  
( 'suresh@gmail.com','hrishikesh13','Suresh','Karnataka',  
'male'),  
( 'rakesh@gmail.com','hrishikesh13','Rakesh','Gujarat', 'male')  
;
```

INSERT INTO

MedicalHistory(id,date,conditions,surgeries,medication)

VALUES

```
(1,'19-01-14','Pain in abdomen','Heart Surgery','Crocine'),  
(2,'19-01-14','Frequent Indigestion','none','none'),  
(3,'19-01-14','Body Pain','none','Iodex')  
;
```

INSERT INTO Doctor(email, gender, password, name)

VALUES

```
('hathalye7@gmail.com', 'male', 'hrishikesh13', 'Hrishikesh  
Athalye'),  
( 'hathalye8@gmail.com', 'male', 'hrishikesh13', 'Hrishikesh  
Athalye')  
;
```

INSERT INTO Appointment(id,date,starttime,endtime,status)

VALUES

```
(1, '19-01-15', '09:00', '10:00', 'Done'),  
(2, '19-01-16', '10:00', '11:00', 'Done'),  
(3, '19-01-18', '14:00', '15:00', 'Done')  
;
```

INSERT INTO

PatientsAttendAppointments(patient,appt,concerns,symptoms)

VALUES

```
('ramesh@gmail.com',1, 'none', 'itchy throat'),  
( 'suresh@gmail.com',2, 'infection', 'fever'),  
( 'rakesh@gmail.com',3, 'nausea', 'fever')  
;
```

```
INSERT INTO Schedule(id,starttime,endtime,breaktime,day)  
VALUES  
(001,'09:00','17:00','12:00','Tuesday'),  
(001,'09:00','17:00','12:00','Friday'),  
(001,'09:00','17:00','12:00','Saturday'),  
(001,'09:00','17:00','12:00','Sunday'),  
(002,'09:00','17:00','12:00','Wednesday'),  
(002,'09:00','17:00','12:00','Friday')  
;
```

```
INSERT INTO PatientsFillHistory(patient,history)  
VALUES  
( 'ramesh@gmail.com', 1),  
( 'suresh@gmail.com', 2),  
( 'rakesh@gmail.com', 3)  
;
```

```
INSERT INTO Diagnose(appt,doctor,diagnosis,prescription)  
VALUES  
(1,'hathalye7@gmail.com', 'Bloating', 'Ibuprofen as needed'),  
(2,'hathalye8@gmail.com', 'Muscle soreness', 'Stretch  
morning/night'),  
(3,'hathalye8@gmail.com', 'Vitamin Deficiency', 'Good Diet')  
;
```

```
INSERT INTO DocsHaveSchedules(sched,doctor)  
VALUES  
(001,'hathalye7@gmail.com'),  
(002,'hathalye8@gmail.com')  
;
```

```
INSERT INTO DoctorViewsHistory(history,doctor)
VALUES
(1,'hathalye7@gmail.com'),
(2,'hathalye8@gmail.com'),
(3,'hathalye8@gmail.com')
;
```

```
SELECT * FROM Patient;
SELECT * FROM MedicalHistory;
SELECT * FROM Doctor;
SELECT * FROM Appointment;
SELECT * FROM PatientsAttendAppointments;
SELECT * FROM Schedule;
SELECT * FROM PatientsFillHistory;
SELECT * FROM Diagnose;
SELECT * FROM DocsHaveSchedules;
SELECT * FROM DoctorViewsHistory;
```

```
//END
```

Concluding Remarks

Developing the software was indeed a tough task for Group-K but we enjoyed it. Moreover, we had also learned that working in a team plays an important role in accomplishing any task. Our *Prof. Aurobindo Behera sir* through his effective management, knowledge and skills looked into the application of the tools and techniques, to describe organize and control the various project processes. His coordination with the software developing team was astounding, her lead Group-K into developing the most successful project.

References:

- 1- Fundamentals of database systems (7th edition, 2015) By Ramez Elmasri, Shamkant B. Navathe.
 - 2- Database Systems: The Complete Book (2nd Edition) 2nd Edition by Hector Garcia-Molina (Author), Jeffrey D. Ullman (Author), Jennifer Widom (Author)
-