PPT

Slide 1: Title

HMS

Slide 2: Team members name with Registration number

In roll number order

Slide 3 and 4: Abstract

Write it in points. (complete the abstract in 2 slides, skip unimportant points if it does not fit)

Hospital Management System (HMS) is a crucial component of the healthcare industry, as it streamlines various processes and enhances the overall efficiency of healthcare organizations. This system integrates various functions, such as patient management, medical records, appointment scheduling, billing, and inventory management, into a single platform. The use of HMS leads to improved patient care, reduced costs, and increased productivity. This study focuses on the design and implementation of a comprehensive HMS, with a user-friendly interface and advanced features to meet the needs of healthcare organizations. The system is developed using modern technologies, such as cloud computing and database management, to ensure scalability, reliability, and security. The results of this study demonstrate the feasibility and effectiveness of the developed HMS, and its potential to improve the quality of healthcare delivery and patient outcomes.

This project Hospital Management system includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs. The software has the facility to give a unique id for every patient and stores the clinical details of every patient and hospital tests done automatically. It includes a search facility to know the current status of each patient. User can search details of a patient using the id. The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

Slide 5: General features of database application

In Bullet points

Patient side features:

- There is a separate interface for patients.
- Patients have a separate login.
- Patients can book appointments.
- Patients can view/update/cancel already booked appointments if necessary.
- Patients are able to see complete diagnoses, prescriptions, and medical histories.

Doctor side features:

- There is a separate interface for doctors.
- Doctors have a separate login.
- Doctors are able to access patient history and profile, and add to patient history.
- Doctors are able to give diagnosis and prescriptions.

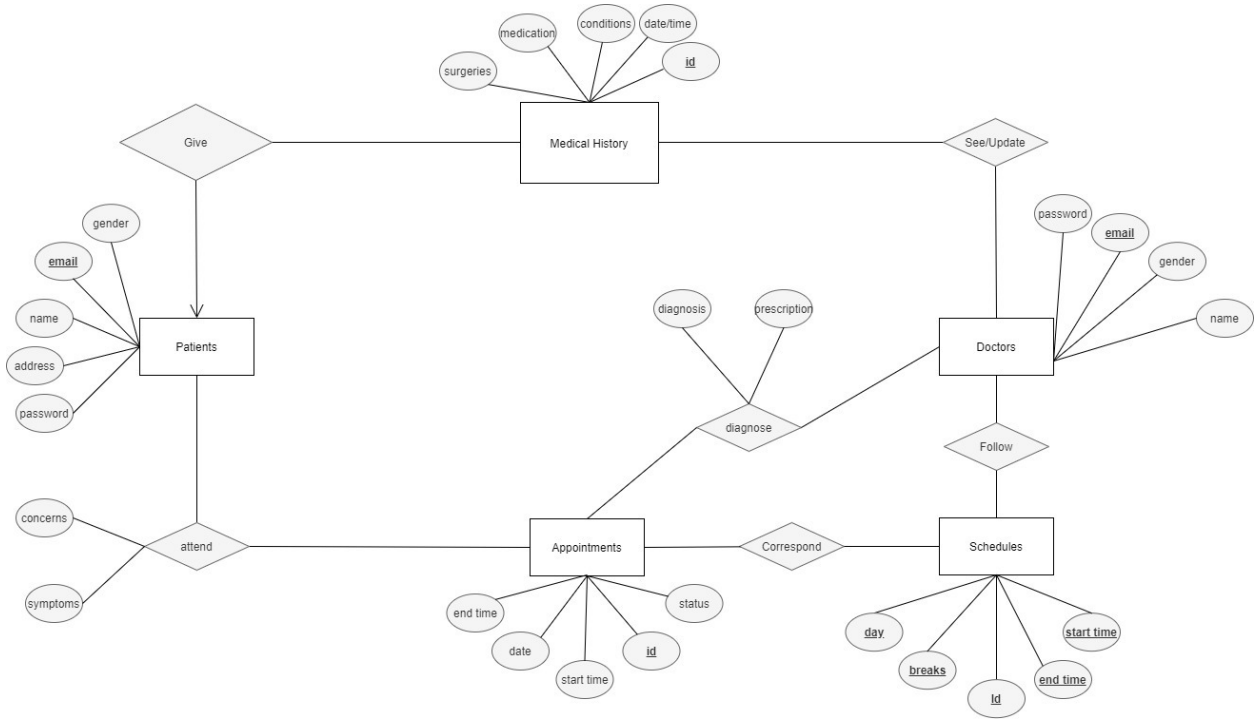
Slide 6: Unique features of database application
In Bullet points
Patient side features:

- Patients can give previous medical history.
- Canceled appointments create free slots for other patients.
- The system avoids clashes of appointments with other patients. Each patient is therefore ensured his/her slot.
- Patient medical history is only available to the doctor with whom the appointment is booked to ensure privacy.

Doctor side features:

- The system takes into consideration doctor schedules and does not allow appointments when a doctor is already busy or has a break.
- Doctors are able to modify diagnosis and prescriptions.

Slide 7: Architecture diagram
png will be inserted when completed
Slide 8: ER diagram
png will be inserted when completed



Slide 9: Frontend design (UI) and software used
Frontend technologies : React.js

UI design hand-drawn sketches:
<insert images by extracting it from the pdf>

Slide 10: Backend (database design) and software used
Backend technologies : Node.js, Express
<insert image by extracting it from the pdf>

Slide 11: Type of connectivity used for database access
Database connectivity acts as the communication interface between the software and the underlying database of the application.
In this project, we are using the relational database connectivity which is used to publish data from a relational database or to deliver data to a relational database. Relational database connection types use ODBC or native database drivers to access data at run time, and they use JDBC to access metadata at design time.
Any of the following connections can be used:
Microsoft SQL Server. Connects to databases though ODBC or through the native database driver.
Oracle. Connects to databases through the native database driver. (most probably we will use Oracle server connection as it is being implemented in the lab as well)
IBM DB2. Connects to databases through the native database driver.

Slide 12: References
https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=56446b88f60668a39dd19da3c47b760b8eb8dfe2 (for architecture diagram)

https://www.edrawsoft.com/article/er-diagrams-for-hospital-management-system.html (for ER diagram)

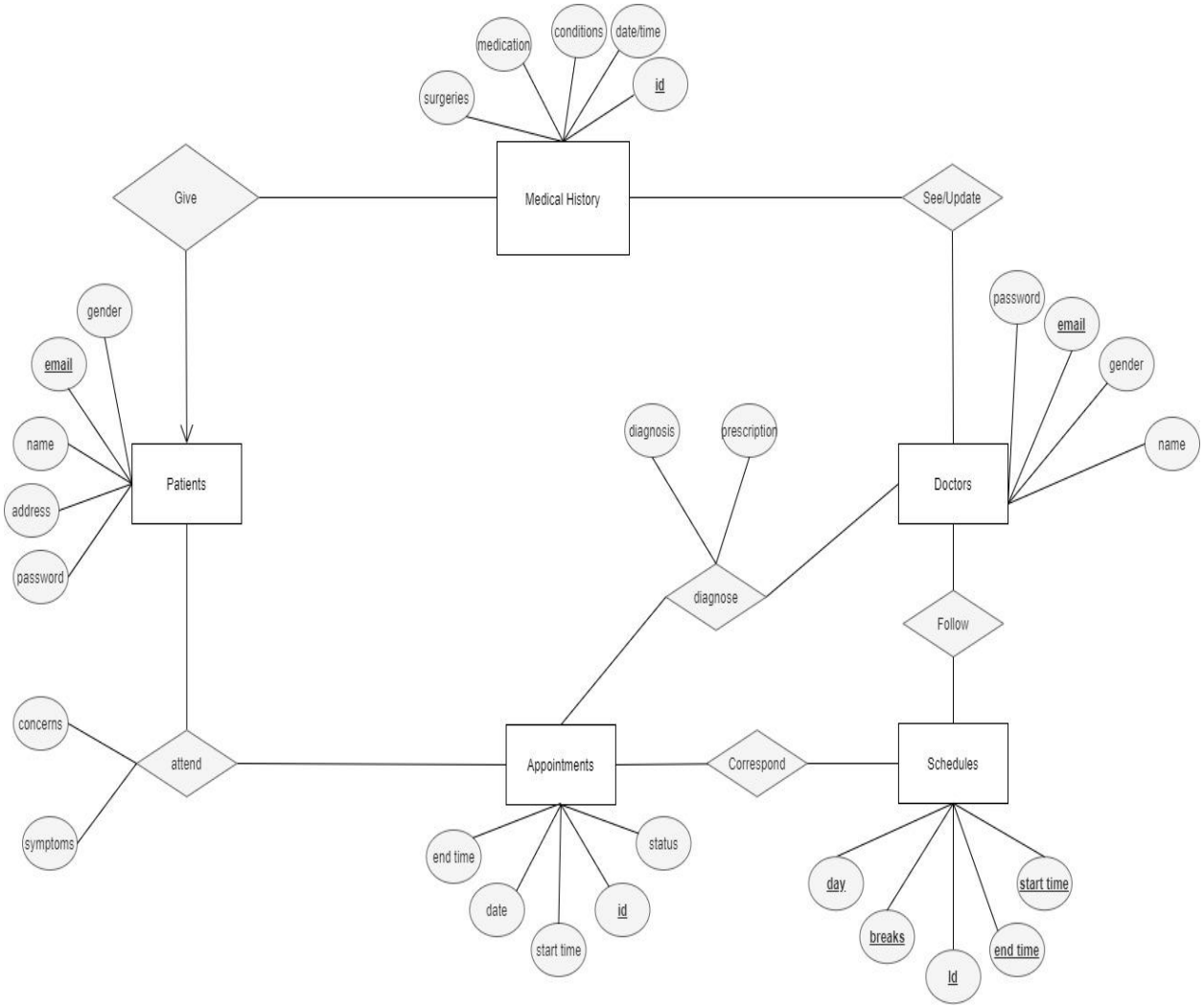
https://www.indiatoday.in/india/story/tamil-nadu-hospital-patients-data-sold-online-dark-web-aiims-server-issue-2304872-2022-12-03 (Data of 1.5L patients sold online)

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC80879/ (obtaining proper medical history of the patient to make informed clinical decisions)

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2779965/

REPORT FOR REVIEW 2:

ENTITY-RELATIONSHIP DIAGRAM:



The above shown ER diagram has the following entities:

- Medical History: Id, date/time, conditions, medication and surgeries.
- Doctors: Email, password, gender and name.
- Schedules: Start time, end time, Id, breaks, and day.
- Appointments: Id, status, start time, date and end time.
- Patients: Email, gender, name, address and password.

The above ER diagram has the following relationships:

- Give: Patients give the medical history.
- See/Update: Doctors see or update the medical history of the patient.
- Follow: Doctors follow schedules.
- Correspond: Appointments correspond to schedules.
- Diagnose: Doctors diagnose patients in the appointments.
- Attend: Patients attend the appointments

The first step is to identify the entity sets. As per the requirements, we will have some main entities. For example, if we are required to observe which patient goes to which hospital or whether they have a previous record or not, or if you simply want to analyze the number of patients a doctor treats.

The second step is to map out the attributes of the entities, (including the identification of key attributes). The key attributes for each entity have been listed below:

Medical History: **Id**, date/time, conditions, medication, and surgeries.
Doctors: **email**, password, gender, and name.
Schedules: **start time**, **end time**, **Id**, **breaks**, and **day**.
Appointments: **ID,** status, start time, date, and end time.
Patients: **email**, gender, name, address, and password.

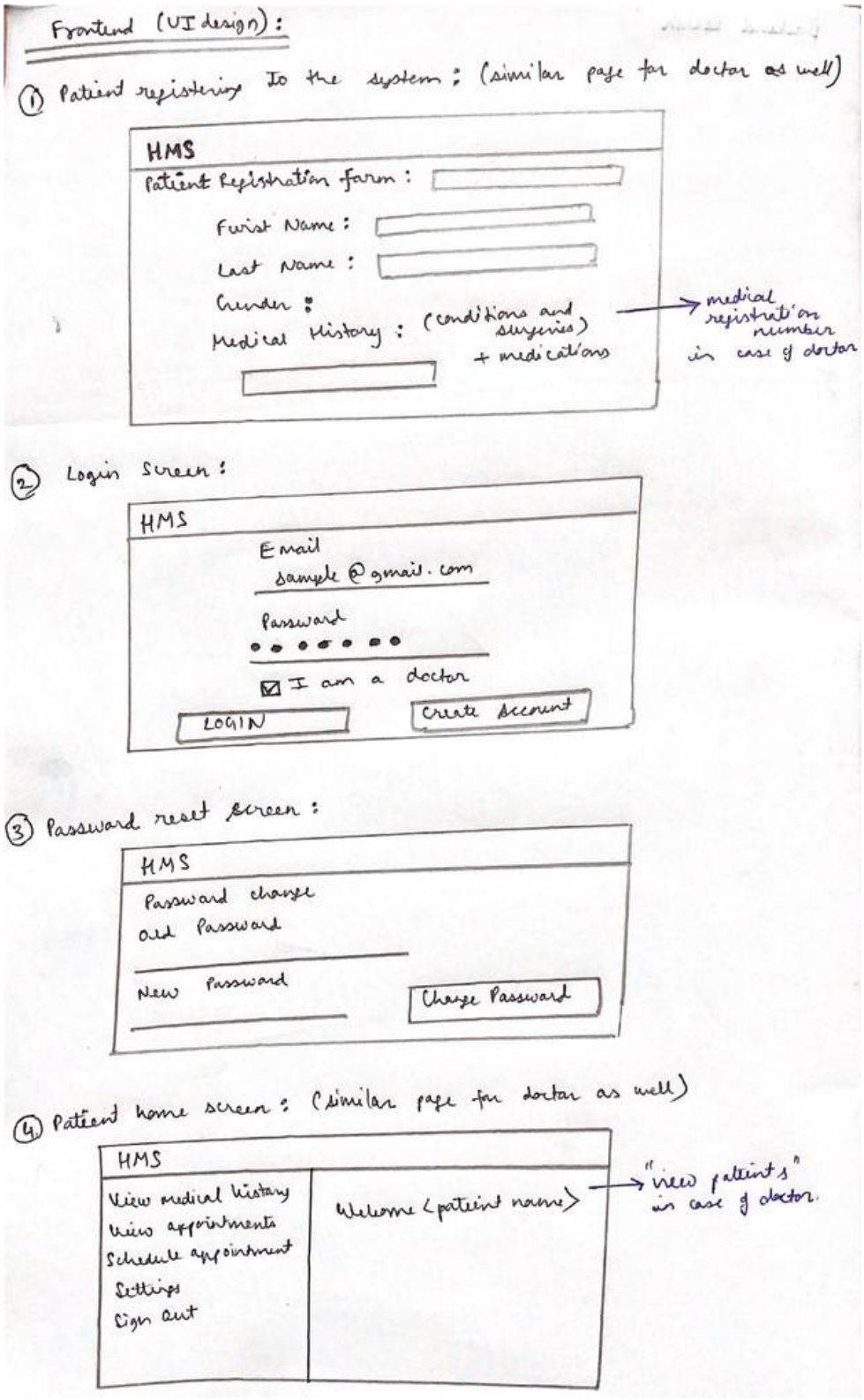
Identify the type of relationship that exists between the entities. This can be done by identifying primary and foreign keys. For example, if the hospital table makes a foreign key reference to the patient ID of the patient table, then both of them will be joined together. Based on the cardinalities, you have to place the appropriate sign.

Once we have identified all the relationships, it is time to map out the lines.

- Since a hospital has multiple patients, it is a one-to-many relationship.
- Since a single hospital has many doctors, the relationship that exists is one-to-many.
- Since a doctor is associated with many patients, it is a one-to-many relationship.
- Since a single patient has multiple medical records, it is a one-to-many relationship.

The last step is to combine all the relationships and draw a complete ER diagram.

FRONT-END UI DESIGN:



(5) Patient viewing appointments:

HMS

Date of Appointment	Start	End	Concerns	Symptoms	Status	
15/01/2019	9:00	10:00	none	itchy throat	Done	See diagnosis
23/10/2020	15:00	16:00	fever	cough	Done	Cancel
17/10/2023	11:00	12:00	indigestion	Acidity	Not Done	

↳ (similar page for doctor also, "diagnose", "prescribe medication buttons"

(6) Doctor giving diagnosis:

HMS

Diagnosis :

Prescription :

Submit Diagnosis

(7) Doctor viewing patient history:

HMS

Search by name

Submit

Name

Ramesh

Suresh

Profile

Medical Profile

Medical Profile

SOFTWARE USED:

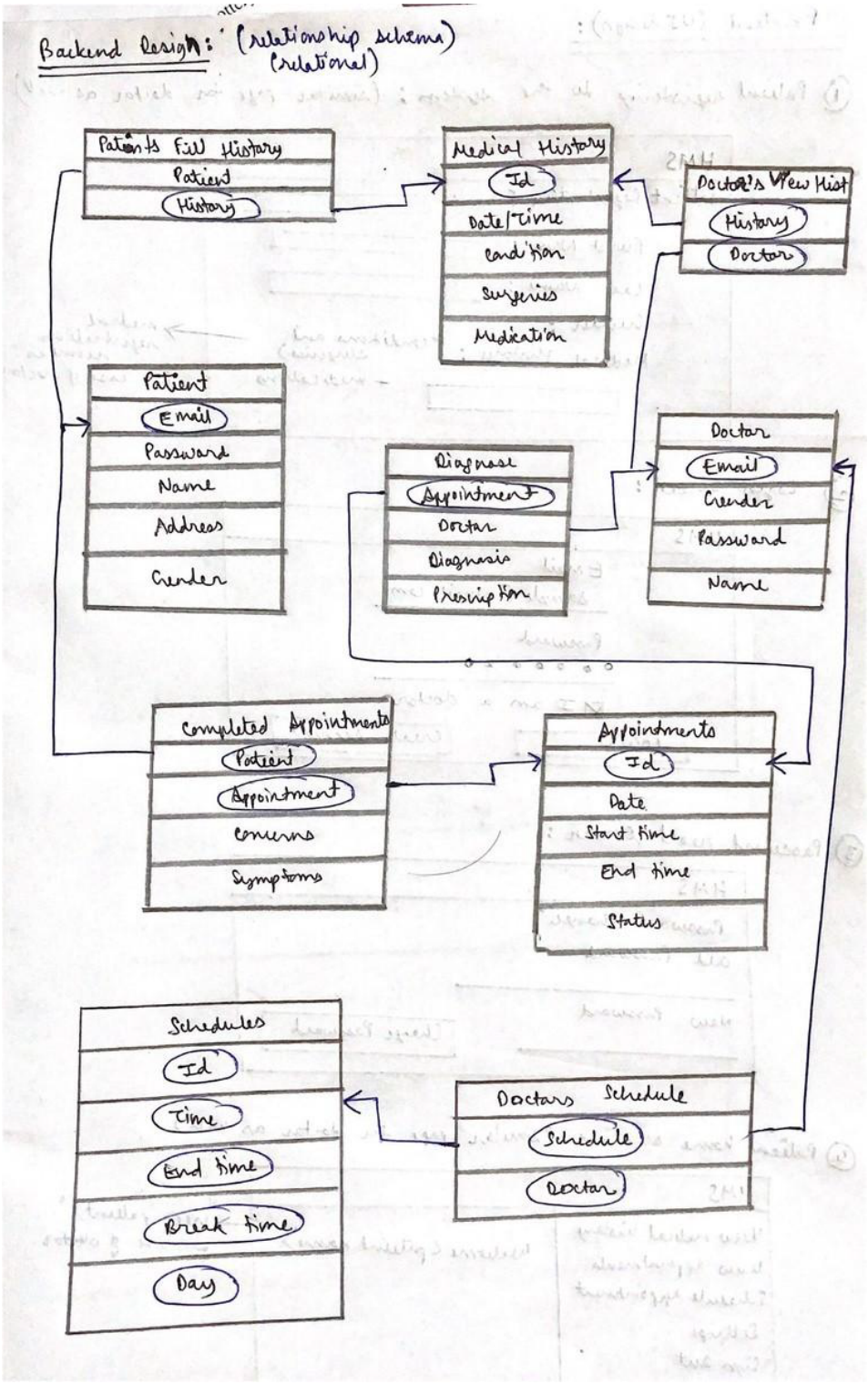
- React.JS

EXPLANATION:

Since we have not yet started implementing the project, we have sketched the UI designs in with a pencil to show an estimated look of the final web application.

- Patient/Doctor registering for the system: Includes the fields of patient registration number provided by the hospital itself, first name, last name, gender and an option to enter the medical history.
- Login screen: Includes the fields of email and password, but also includes two checkboxes having the options of "I am a doctor" and "I am a student". Simultaneously, two buttons will be displayed, login and create an account.
- Password reset screen: Prompts the user to reset the password by entering the new password and then entering the new password again to make sure there are no typing errors.
- Patient or doctor home screen: Displays a welcome message for the user and shows an array of available web services on a left panel, the services being:
    1. View medical history
    2. View appointments
    3. Schedule appointments
    4. Settings
    5. Sign out
- Patient viewing appointments: Shows a tabular view of the previous and upcoming appointments and finally, two buttons of "see diagnosis" and "cancel" will also be provided for convenience. The table includes:
    1. Date of Appointment
    2. Start time
    3. End time
    4. Concerns
    5. Symptoms
    6. Status
- Doctor giving diagnosis: Two separate text fields will be prompted for the doctor to enter the diagnosis and the prescription medications, and then a submit diagnosis button to save the diagnosis to the database.
- Doctor viewing patient history: Here the doctor can search the patient history by name by submitting the full name. On a successful search, the name and hyperlinked medical profile will be displayed.

BACK-END DESIGN:



SOFTWARE USED:

- Node.JS, Express