PROJECT GROUP – 10

HOSPITAL MANAGEMENT SYSTEM

DATABASE DESIGN AND FINAL ERD: Database Purpose, Business Problems, Business Rules, Design Requirements, Design Decisions, and Entity Relationship Diagram.

TEAM MEMBERS

- Anjali Sajeevan
- Deepika Kumari Jha
- Madhurima Chatterjee
- Sayali Nitin Chaudhary
- Shreya Uppalapati
- Pooja Singh Kumari

DATABASE PURPOSE

The purpose of the project is to design a Hospital Database System. It helps in providing enhanced facility management and decision making. It avoids errors, tracks patient details, and generates MIS (Management Information System) reports on demand for better decision making and coordination among departments. The hospital has a dedicated department for COVID-19 patient treatment and research.

BUSINESS PROBLEMS ADDRESSED

1. The age column is kept up to date by computing the column using the date of birth and the current date.
2. A report is generated to analyze the departments to which each staff member is assigned.
3. A report is generated to analyze the most COVID affected region.
4. The status of the availability of all wardrooms is monitored.
5. A report for the number of inpatient and outpatient per day is visualized and no outpatient will be assigned a wardroom.
6. A report is visualized to analyze the most common disease of every state in India.
7. A report is visualized to determine the percentage of the most common medical tests undertaken by patients.
8. A report is visualized to obtain the quantity of each stock available in the inventory.
9. A report is visualized to obtain the age range of patients with respect to their diagnosis.
10. A report is visualized to obtain the quantity available for each blood group.

BUSINESS RULES BASED ON SCHEMAS

Patient:

1. A patient can consult more than one doctor for different health issues.
2. A patient can undergo multiple tests and will get the respective test results.
3. A patient may or may not have health insurance.
4. An appointmentID is a token number assigned to patients, it can either be assigned on the spot or booked in advance.
5. A patient can request more than one Test.
6. Staff working with StaffCode “LAB” is responsible for carrying out the test.
7. TestResult Description will be provided to the respective Patient.

Blood Bank:

1. Multiple donors can be registered and donate blood to the BloodBank.
2. Multiple patients can accept the blood from BloodBank.
3. BloodBank should keep BloodDonationRecord, AcceptanceRecord and BloodStock records.

Doctors:

1. Doctor can have one or more Patients.
2. A Doctor can be assigned to only one department.

Departments:

1. Department can have one or more doctor.
2. A staff member is allocated to only one or more departments.
3. A department can have one or more staff members.

COVID:

1. At any given time at least one doctor will be assigned to the COVID19Department.
2. At any given time at least one Staff will be assigned to the COVID19Department.
3. At any given time one or more inventory will be assigned to the COVID19Department.

Medicine:

1. A patient can get prescribed various medicines.

DESIGN REQUIREMENTS

1. Crow’s Foot Notation used for the ER Diagram.
2. Specified the primary key and foreign key fields in each entity by denoting PK and FK next to the attribute.
3. The relationship types between the entities were established.
4. A line between the fields of each table shows the relationships between each table. This line points directly to the fields in each table that are used to form the relationship.
5. Associative entities were created to avoid many to many relationship type.
6. The entities were normalized to 3NF.

DESIGN DECISIONS

Entity Name	Why Entity Included	How Entity is related to Other Entities
Patient	The Patient is one of the core entities of the Hospital Management System. The purpose of this entity is to collect Patient-related data i.e. personal information, diagnosed along with consulting doctors, Insurance, etc. If a patient is admitted to the hospital then the respective ward number with entry and the exit date is also mentioned in the Patient data. In COVID19 pandemic patient's personal information, diagnosis, symptoms and medication can help in future decision making of precautionary majors.	There is one to many relationship from patient to AcceptanceRecord. There is one to many relationship with associative entity PatientMedicine There is zero to many relationship with Insurance. There is one to many relationship with associative entity PatientTest. There is optional zero to 1 relationship from Patient to Wardrooms. There is one to many relationship from Patient to PatientRegistration.
PatientTest	To keep the records of tests associated with each patient and its result, the PatientTest entity is created as an associative entity between Patient, TestDetails and Staff. PatientID and TestID are used to create a composite Primary key. StaffID is a foreign key.	There are many to one relation from PatientTest to Patient,TestDetails and Staff entity, with many side towards the PatientTest.
PatientMedicine	To keep track of the medicines prescribed to each patient and its availability, the PatientMedicine entity is used as an associative entity between the Patient and Medicine entity. The PatientID and MedicineID are used to create a composite primary key.	There is many to one relationship between PatientMedicine to Patient There is many to one relationship between PatientMedicine to Medicine.

Medicine	The medicine entity contains the details of the medicine, the cost and the quantity available in stock.	There is one to many relationship between Medicine to PatientMedicine.
Insurance	If the patient is having health insurance, then to maintain insurance details with respect to that patient the insurance entity is added in the database. It includes PatientID, InsuranceID, Insurance Provider and coverage of bill payment from that insurance.	There is an optional one to one relationship between Insurance and Patient i.e. Patient may or may not own an insurance.
WardRooms	Entity WardRoom is used to maintain data related to the patient who is admitted in the hospital along with PatientID, floor, type and Status to keep track of all allocated ward for a patient, its location and status	There is an optional one to one relationship between Patient and Wardrooms. A patient may or may not be admitted to the hospital. An admitted patient will be allocated WardRooms.
TestDetails	This entity is used to keep a record of all the available tests along with its details in the hospital. It has an attribute TestID that is used to uniquely identify a test	There is one to many relationship between TestDetails to associative entity PatientTest , with many towards PatientTest
Doctor	The doctor is one of the core entities in the Hospital Management System. The entity has a list of all the doctors available in the hospital along with their details including the assigned department.	There is one to many relationship from Doctor to PatientRegistration. There is many to one relationship from Doctor to COVID19Department. There is many to one relationship from Doctor to Department The entity is related to the PatientRegistration entity which helps in getting information about an appointment with the patient. It also connects to the department entity that helps us to determine which department the doctor is assigned.
Department	This entity maintains a record of different departments, names of the department, number of doctors	The Department Entity is connected to the Doctor and DepartmentStaff Entity.

	and wards assigned to a particular department.	There is one to many relationships between departments and doctor. There is one to many relationships between department and staff.
DepartmentStaff	This is an associative entity created to avoid many-to-many relationship between Department and Staff	There is one to many relationship between Department and DepartmentStaff There is one to many relationship between Staff and DepartmentStaff.
Staff	This entity holds personal data (such as Full name, contact number, etc.), biodata (such as dob, gender, etc.) and Income information like joining date and salary) of all the staff members working in the hospital. Examples of staff members could be nurses, technicians, housekeeping, etc.	There is one to many relationship between Staff and associative entity PatientTest. There is one to many relationship between Staff and associative entity DepartmentStaff There is one to many relationship from COVID19Department to Staff
BloodBank	This entity holds the information about BloodBank like BloodID and BloodBankName, State, City, Zip Code.	There are one to many relationships with BloodDonationRecord,BloodStock , AcceptanceRecord.
BloodStock	BloodStock entity holds the information about BloodStock available in BloodBank along with information like BloodGroup, BloodQuantity, BestBefore(Expiry Date). BloodStock is connected with BloodBank via foreign key BloodBankID.	There is one to many relationship between BloodBank and BloodStock as there will be multiple stocks of blood in BloodBank.
BloodDonationRecord	DonationRecord entity holds the information about all the blood donations carried out in BloodBank. The attributes like DonationRecordID, DonarName, DonationDate,	There is one to many relationship between BloodBank and BloodDonationRecord. It is connected with BloodBank via foreign key BloodBankID

	DonationBloodGroup, BloodAmount will help to keep details of Donors and Donated blood amounts in BloodBank.	
AcceptanceRecord	AcceptanceRecord entity holds the information about the acceptance of blood from any patient carried out in the hospital's BloodBank. The attributes like PatientID AcceptanceDate, AcceptedBloodGrp, AcceptedBloodAmt will help to keep track of AcceptanceRecord of blood by patients.	There is one to many relationship between BloodBank and AcceptanceRecord. There is one to many relationship between Patient and AcceptanceRecord. It is connected with BloodBank entity via foreign key BloodBankID. It is connected with the Patient entity via foreign key PatientID.
PatientRegistration	This entity stores appointments details of patients. It includes the DoctorID assigned to a particular patient along with the date of the appointment.	There is a one to one relationship with Patient entity. There is one to many relationship with Doctor entity.
COVID19Department	This entity includes the StaffID, DoctorID and InventoryCode. This is a special department allocated for COVID19 pandemic in order to keep track of all Staff, Doctors and inventory items assigned to COVID department to take precautionary steps.	There is one to many relationship with doctor entity. i.e. COVID19Department can have one or more doctor assigned. It has one to many relationship with Staff and Inventory i.e. COVID19Department can have one or more staff assigned
Inventory	This entity stores information about all the equipment available in the hospital with the quantity. If the quantity attribute in Inventory becomes zero, then the hospital must order for more equipment.	It has one to many relationship with COVID19Department , with many towards Inventory.

ENTITY RELATIONSHIP DIAGRAM

Patient

- PK PatientID
- Diagnosis
- InPatient/OutPatient
- BloodGroup
- EntryDate
- ExitDate
- Symptoms
- PatientFirstName
- PatientLastName
- Gender
- GuardianFirstName
- GuardianLastName
- DateofBirth
- Age
- GuardianRelation
- PhoneNumber
- EmailAddress
- StreetName
- City
- ZIPCode

PatientMedicine

- PK FK PatientID
- PK FK MedicineID

Insurance

- PK InsuranceID
- FK PatientID
- Provider
- Coverage

PatientTest

- PK FK PatientID
- PK FK TestID
- FK StaffID
- TestResultDescription
- Date

TestDetails

- PK TestID
- TestName
- TestDescription

BloodBank

- PK BloodBankID
- BloodBankName
- State
- City
- Zipcode

BloodStock

- PK StockID
- FK BloodBankID
- BloodGroup
- Status
- Quantity
- BestBefore

Medicine

- PK MedicineID
- Medicinename
- DrugClass
- Symptoms
- Cost
- QuantityAvailable

Inventory

- PK InventoryCode
- InventoryType
- Brand
- StockQuantity

WardRooms

- PK RoomID
- FK PatientID
- Floor
- Type
- Status

COVID19Department

- PK FK StaffID
- PK FK DoctorID
- FK InventoryCode

AcceptenceRecord

- PK AcceptanceRecordID
- FK PatientID
- FK BloodBankID
- AcceptanceDate
- AcceptancePlace
- AcceptedBloodGroup
- AcceptedBloodAmount

PatientRegistration

- PK AppointmentID
- FK DoctorID
- FK PatientID
- Date

Staff

- PK StaffID
- StaffCode
- StaffDesignation
- StaffFirstName
- StaffLastName
- ReportTo
- Gender
- JoiningDate
- Salary
- DateOfBirth
- Age
- EmailAddress
- PhoneNumber
- State
- City
- ZIPCode

BloodDonationRecord

- PK DonationRecordID
- FK BloodBankID
- DonorName
- DonorState
- DonorCity
- DonorZIPcode
- DonationDate
- DonationBloodGroup
- BloodAmount

Doctor

- PK DoctorID
- FK DepartmentID
- DoctorsFirstName
- DoctorsLastName
- Specialization
- Gender
- StreetName
- City
- ZIPCode
- ContactDetails
- WorkingHours
- Salary
- EmailAddress
- DateofBirth
- Age

Department

- PK DepartmentID
- DepartmentName
- NumberOfDoctors
- NumberOfAssignedWards

DepartmentStaff

- PK FK DepartmentID
- PK FK StaffID