Patient Medical Data Analytics System for Doctors

A Tiered Data Management Application for Medical Information and Statistical Insight

Description:

The Patient Records and Analytics System for Doctors is a tiered data management application designed to help doctors efficiently manage, add, view, update, and analyze patient medical/demographic data across multiple hospitals in a confidential manner aligned with patient privacy guidelines. This allows doctors to analyze ongoing patient cases to view medical trends efficiently to support research.

The system allows doctors to perform full CRUD operations (adding, viewing, updating, and deleting patient records) while maintaining structured relationships between patients, doctors, hospitals, conditions, medications, and immunizations.

In addition to record management, the system provides analytical capabilities such as identifying the most common conditions, tracking prescriptions, and visualizing demographic and disease trends through data-driven charts.

This application focuses on improving accessibility of patient data for doctors without the addition of administrative details to support patient data analysis by offering an organized backend database and a clear, interactive user interface for data visualization and statistical insight.

Use Cases

System: Patient Records and Analytics System for Doctors

Primary Actor: Doctor

1. **Authenticate User**
   Goal: Securely log in to the system.
   Description: The doctor enters credentials (password). The system verifies identity and grants access to features and patient data. All doctors in this hospital system have access to this database with the same password to add, view, update and analyze patient data in a confidential manner.
2. **Patient Lookup**
   Goal: Quickly retrieve a patient's information by entering patient ID.
   Description: The doctor searches by patient ID to view demographics, blood type, immunizations, conditions, medications, hospitals, and treating doctors of a specific patient.
3. **Add Patient/Record**
   Goal: Create a new patient record.

Description: The doctor enters demographics; links conditions, medications, immunizations, doctors, and hospitals.

4. **Update Patient/Record**
   Goal: Edit existing patient data.
   Description: The doctor updates demographics, adds or removes conditions or medications, updates immunizations, and adjusts hospital/doctor associations.
5. **Delete Patient/Record**
   Goal: Remove or archive a patient record.
   Description: The doctor deletes or archives inactive or outdated records to keep data organized.
6. **View Patient Data**
   Goal: View patient data easily.
   Description: The doctor views patient information in organized tables:
   a. Patients by Hospital
   b. Patients by Medical Condition
   c. Patients by Doctor
   d. Patients by Blood Type
7. **Analyze Disease/Demographic Prevalence**
   Goal: Extract statistical insights.
   Description: The doctor runs analytics to view:
   a. Top 5 most prevalent conditions per hospital
   b. Most prevalent age per condition
   c. Most prevalent gender for a condition
   d. Most prevalent blood type per condition
   e. Number of patients that took vaccine

Database Design — E/R Description

Entities (6)

1. **Patient** *(patient_id, medical_record_number, birth_year, gender, blood_type)*
2. **Doctor** *(doctor_id, name, specialty, hospital_id)*
3. **Hospital** *(hospital_id, name, city)*
4. **Condition** *(condition_id, name)*
5. **Medication** *(medication_id, name)*
6. **Vaccine** *(vaccine_id, name)*

Relationships (6 total; 5 many-to-many)

Patient — Doctor: M:N
A patient can be treated by many doctors; a doctor treats many patients.

Patient — Hospital: M:N
A patient can visit many hospitals; a hospital serves many patients.

Doctor — Hospital: N:1
Each doctor works at exactly one hospital; a hospital employs many doctors.

Patient — Condition: M:N
Patients can have multiple conditions; conditions can apply to many patients.
Attributes:

- `diagnosis_date` **— when the condition was diagnosed**
- `doctor_id` **— the diagnosing doctor (foreign key)**

Patient — Medication: M:N
Patients can take multiple medications; medications can be prescribed to many patients.
Attributes:

- **`prescription_date` — when the medication was prescribed**
- **`doctor_id` — the prescribing doctor (foreign key)**

Patient — Vaccine (Immunization): M:N
Patients can receive multiple vaccines; each vaccine can be given to many patients.
Attributes:

- **`admin_date` — date vaccine was administered**

Relation Specification (ER → Tables)

Patient

- `patient_id` (PK)
- medical_record_number
- `birth_year`
- `gender`
- `blood_type`

Doctor

- `doctor_id` (PK)
- `name`

- specialty
- hospital_id (FK → Hospital.hospital_id)

Hospital

- hospital_id (PK)
- name
- city

Condition

- condition_id (PK)
- name

Medication

- medication_id (PK)
- name

Vaccine

- vaccine_id (PK)
- name

PatientDoctor — join for M:N (Patient ↔ Doctor)

- `patient_id` (FK → Patient.patient_id)
- `doctor_id` (FK → Doctor.doctor_id)
- **PK:** (patient_id, doctor_id)

PatientHospital — join for M:N (Patient ↔ Hospital)

- `patient_id` (FK → Patient.patient_id)
- `hospital_id` (FK → Hospital.hospital_id)
- **PK:** (patient_id, hospital_id)

PatientCondition — join for M:N (Patient ↔ Condition)

- `patient_id` (FK → Patient.patient_id)
- `condition_id` (FK → Condition.condition_id)
- `diagnosis_date`
- `diagnosed_by_doctor_id` (FK → Doctor.doctor_id)
- **PK:** (patient_id, condition_id, diagnosis_date)

PatientMedication — join for M:N (Patient ↔ Medication)

- `patient_id` (FK → Patient.patient_id)
- `medication_id` (FK → Medication.medication_id)

- `prescription_date`
- `prescribed_by_doctor_id` (FK → Doctor.doctor_id)
- **PK:** (patient_id, medication_id, prescription_date)

PatientImmunization — join for M:N (Patient ↔ Vaccine)

- `patient_id` (FK → Patient.patient_id)
- `vaccine_id` (FK → Vaccine.vaccine_id)
- `admin_date`
- **PK:** (patient_id, vaccine_id, admin_date)