

Title: AI- Enabled Healthcare Database System

Name: Ali Alsharif

Student ID: 918063606

SFSU email: aalsharif@mail.sfsu.edu

Github username: AliAlsharif99

Checkpoints	Date Submitted
Checkpoint 1 v1	09/17/2023
Checkpoint 2	10/04/2023

Project Description:

The AI-enabled Healthcare database system is a comprehensive solution designed for managing healthcare data efficiently. It stores patient information, doctor schedules, medical records, and research data. Professionals shall find it user-friendly, thanks to AI tools that help with data retrieval, personalized recommendations, and medical image analysis.

This system's standout feature is AI integration, which automates tasks and offers personalized insights. It enhances data management and supports professionals in their decision-making processes. For instance, it speeds up medical image analysis, making it easier for doctors to detect issues accurately. The system also has features such as appointment scheduling, medication calculation, and it shall aid medical researchers in data analysis. In conclusion, it's a powerful tool that combines AI with healthcare data management to improve patient care and research efforts.

2 software tools that would benefit from using the database system:

Picture Archiving and Communication System (PACS)

PACS is used for storing, managing, and distributing medical images such as X-rays, MRIs, and CT scans. Integrating AI-enabled Healthcare Database System with PACS would significantly enhance its capabilities in medical image analysis and storage.

Clinical Decision Support Systems (CDSS)

CDSS tools assist healthcare providers in making clinical decisions by providing evidence-based recommendations. Integrating AI-enabled Healthcare Database System with CDSS would enhance the decision support capabilities in data retrieval, and AI-powered recommendations.

Table of contents:

Cover Page	1
Use-cases	3
Functional Requirements	5
ERD	11
Entity Description	12
Constraints Description.....	16
EED	20
Non-functional Requirements.....	21

Use-case: Image-based Diagnosis

Actor: Radiologist (John)

Description: John is a radiologist working in a research lab of a well-known hospital. He is tired of analyzing patient's medical scans and can't work in a faster pace because he can't risk misdiagnosing a case. To improve his and his team's work rate, they suggested the use of AI tools to help them detect anomalies in a faster, more precise way.

Use-case: Treatment plans

Actor: Physician (Mohammed)

Description: Mohammed has been a physician for many years. He oversees many cases such as common illnesses and patients with cancer. Creating a good treatment plan requires him to invest a lot of time and hard work to get the best treatment plan for his patients. The harder cases such as his cancer patients need more care with their treatment plans since their illnesses are life threatening so it drains his energy studying the patient's treatment responses and genetic profiles. Having a personalized treatment plan using AI to study the patient's genetic profiles will improve his performance and efficiency in creating treatment plans.

Use-case: Overbooking Appointments

Actor: Medical Office Scheduler (Sherry)

Description: Sherry takes a lot of phone calls for scheduling appointments in the hospital. She doesn't mind taking the same amount of phone calls, but her issue is looking into each doctor's schedule every time she gets a phone call for an appointment. She gets frustrated when she can't find appropriate schedule times or the doctor's is fully booked. Sherry recommends the use of a software that can prevent overbooking of appointments, ensuring that practitioners have sufficient time for each patient encounter. The system will consider appointment durations and the nature of medical visits to make it more efficient.

Use-case: Medication Calculation

Actor: Nurse (Sereen)

Description: Sereen is a recovery room nurse, and she looks over many patients. Patients are prescribed medications from the doctors, and she gives her patients their medication in specific times of the day. Since she works with a lot of patients, efficiently managing the patients time of medication and calculating the doses can be exhausting and repetitive. She recommends the use of a software where she can have automated medication reminders and input the patient information and condition to get the best medication time and dosage according to the patient case, so it doesn't have to be necessary to consult the physician, which would make both their work more efficient.

Use-case: Research Data Analysis

Actor: Medical researcher (Jose)

Description: Jose is a medical researcher working in a drug testing lab. His research is focused on analyzing big data sets to identify new drug discoveries. He gets burnt out frequently because he's always analyzing big sets of data and it gets tiring for him. He suggests the use of a software that can support him with his analysis to increase his productivity and help him identify trends and patterns for drug discovery.

Functional Requirements:

1-User

- 1.1 A user shall go through identity verification.
- 1.2 A user shall have a username.
- 1.3 A user shall have a password.
- 1.4 A user shall have a unique ID.

2-Registered User

- 2.1 A registered user is a user.
- 2.2 A registered user shall have a role.
- 2.3 A registered user shall have full access to his/her department's database.
- 2.4 A registered user shall be able to manage their schedule.

3-Patient

- 3.1 A patient is a user.
- 3.2 A patient shall have access to their own medical records.
- 3.3 A patient shall have access to their own personal information.
- 3.4 A patient shall have the ability to schedule appointments.
- 3.5 A patient shall have the ability to view their upcoming and past appointments.
- 3.6 A patient shall be able to receive automated notifications.

4-Doctor

- 4.1 A doctor is a registered user.
- 4.2 A doctor shall have a specific role.
- 4.3 A doctor shall have access to patient medical records.
- 4.4 A doctor shall have access to patient diagnostic reports.
- 4.5 A doctor shall have access to patient treatment plans.
- 4.6 A doctor shall be able to update patient records.
- 4.7 A doctor shall be able to add new diagnosis to patients.
- 4.8 A doctor shall be able to update patient treatment plans.
- 4.9 A doctor shall be able to add prescriptions.
- 4.10 A doctor shall have access to AI tools for clinical decision support.
- 4.11 A doctor shall have access to AI tools for aiding in diagnosis.
- 4.12 A doctor shall have access to AI tools for treatment recommendations.
- 4.13 A doctor shall have access to Lab Tests.

5-Researcher

- 5.1 A researcher is a registered user.
- 5.2 A researcher shall have access to research databases and tools for data analysis.
- 5.3 A researcher shall be able to upload research data.
- 5.4 A researcher shall be able to manage research data.
- 5.5 A researcher shall be able to analyze research data.
- 5.6 A researcher shall be able to collaborate with other researchers. (Recursive)
- 5.7 A researcher shall belong to one department.
- 5.8 A researcher shall have access to Medical Records.
- 5.9 A researcher shall have access to Lab Tests.

6-Nurse

- 6.1 A nurse is a registered user.
- 6.2 A nurse shall have access to Medical Records.
- 6.3 A nurse shall have access to medication administration.
- 6.4 A nurse shall have access to patients care plans.
- 6.5 A nurse shall be able to receive automated medication administration reminders.
- 6.6 A nurse shall have access to AI dosage calculation tools.
- 6.7 A nurse shall be able to record patient observations in the database.
- 6.8 A nurse shall be able to record patient vital signs in the database.
- 6.9 A nurse shall have access to Lab Tests.

7- Staff

- 7.1 A Staff is a registered user.
- 7.2 A Staff shall have access to appointment scheduling tools.
- 7.3 A Staff shall help patients in scheduling appointments.
- 7.4 A Staff shall access many patients' billing information
- 7.5 A Staff shall verify patient insurance coverage.

8-System Administrators

- 8.1 A system administrator is a user.
- 8.2 A system administrator shall have highest system access and control.
- 8.3 A system administrator shall be able to manage user accounts.
- 8.4 A system administrator shall be able to manage user roles.
- 8.5 A system administrator shall be able to manage permissions.
- 8.6 A system administrator shall have access to system configuration settings.

9-Hospital

- 9.1 A hospital shall contain many departments.
- 9.2 A hospital shall have an address.
- 9.3 A hospital shall have a name.

10-Departments

- 10.1 Each department shall belong to one hospital. (aggregation)
- 10.2 A department shall have many doctors.
- 10.3 A department shall have many nurses.
- 10.4 A department shall have many researchers.

11-Medical Records

- 11.1 Medical records shall contain information about diagnoses, treatments, and medical history.
- 11.2 Medical Records can be accessed by many doctors.
- 11.3 Medical Records can be accessed by many nurses.
- 11.4 Medical Records can be accessed by many researchers.
- 11.5 Doctors and nurses shall be able to update patient medical records.

12-Billing information

- 12.1 Billing information shall be generated for patients.
- 12.2 Billing information shall be accessed by Staff.
- 12.3 Billing information shall contain details about services, costs, and payment status.

13-Insurance Information

- 13.1 Patients shall provide insurance information for billing and verification.
- 13.2 Insurance information and coverage shall be verified by Staff.
- 13.3 Insurance information shall include policy details, coverage, and contact information.

14-Lab Tests

- 14.1 Doctors shall have the ability to order lab tests for patients.
- 14.2 Lab tests shall be scheduled and conducted in the lab.
- 14.3 Lab results shall be linked to specific lab tests.
- 14.4 Doctors shall have access to lab test information.
- 14.5 Nurses shall have access to lab test information.
- 14.6 Researchers shall have access to lab test information.

15-Medication Inventory

- 15.1 Medication inventory shall track available medications, including stock levels and expiration dates.
- 15.2 Medication inventory shall be accessed by nurses.

16- AI Tools

- 16.1 AI tools shall be utilized by Doctors, Nurses, and Researchers.
- 16.2 AI tools shall be able to manage appointments.

17- Bank account

- 17.1 A bank account has an account number.
- 17.2 A bank account has a billing address.
- 17.3 A bank account processes patient billing information
- 17.4 A bank account has a debit card.
- 17.5 A bank account has a credit card.

Entity Description:

1. User (Strong):
 - User_ID: key, numeric
 - Username: composite, alphanumeric
 - Password: composite, alphanumeric
2. Registered User (Weak)
 - User_ID: key, numeric
 - Username: composite, alphanumeric
 - Password: composite, alphanumeric
 - Role: composite, alphanumeric
 - Weekly_Schedule: composite, text
3. Patient (Weak)
 - User_ID: key, numeric
 - Patient_ID: key, numeric
 - Username: composite, alphanumeric
 - Password: composite, alphanumeric
4. Doctor (Weak)
 - User_ID: key, numeric
 - Username: composite, alphanumeric
 - Password: composite, alphanumeric
 - Role: composite, alphanumeric
 - Weekly_Schedule: composite, text
5. Nurse (Weak)
 - User_ID: key, numeric
 - Username: composite, alphanumeric
 - Password: composite, alphanumeric
 - Role: composite, alphanumeric
 - Weekly_Schedule: composite, text
6. Researcher (Weak)
 - User_ID: key, numeric
 - Username: composite, alphanumeric
 - Password: composite, alphanumeric
 - Role: composite, alphanumeric
 - Weekly_Schedule: composite, text

7. Staff (Weak)
 - User_ID: key, numeric
 - Username: composite, alphanumeric
 - Password: composite, alphanumeric
 - Role: composite, alphanumeric
 - Weekly_Schedule: composite, text
8. System Administrator (Weak)
 - User_ID: key, numeric
 - Username: composite, alphanumeric
 - Password: composite, alphanumeric
9. Department (Strong)
 - Dept_ID: key, numeric
 - Dept_location: composite, alphanumeric
 - Dept_name: composite, alphanumeric
10. Department Database (Strong)
 - Session_ID: key, alphanumeric
 - Permissions: numeric
11. Hospital (Strong)
 - Hospital_ID: key, numeric
 - Name: composite, alphanumeric
 - Address: composite, alphanumeric
12. Management associative entity (department and hospital)(Strong)
 - Management_ID: key, numeric
 - Hospital_ID: key, numeric
 - Dept_ID: key, numeric
13. Appointments (Strong)
 - App_ID: key, numeric
 - Appointment_history: composite, alphanumeric
 - Patient_ID: key, numeric
14. Insurance (Strong)
 - Insurance_ID: key, numeric
 - Coverage: composite, alphanumeric
 - Contact_information: composite, alphanumeric
15. Insurance registration associative entity (Weak)
 - Reg_ID: key, numeric
 - Patient_insurance_information: composite, alphanumeric

- 16.ID verification (Strong)
 - Verification_ID: key, numeric
 - Is_verified: Boolean
 - Date: multivalue, timestamp
- 17.Account (Strong)
 - Acc_ID: key, numeric
 - Permissions: numeric
 - Name: composite, alphanumeric
- 18.Medical Records (Strong)
 - Record_ID: key, numeric
 - Diagnostic_reports: composite, alphanumeric
 - Treatment_plans: composite, alphanumeric
- 19.AI Tools (Strong)
 - Tool_ID: key, numeric
 - Description: alphanumeric
 - Tool_name: composite, alphanumeric
- 20.Medication inventory (Strong)
 - Prod_ID: key, numeric
 - Stock: numeric
 - Date: multivalue, timestamp
- 21.Lab (Strong)
 - Lab_ID: key, numeric
 - Location: composite, alphanumeric
 - Type: composite, alphanumeric
- 22.Lab test information associative entity (Weak)
 - Test_ID: key, numeric
 - Patient_ID: key, numeric
 - Result: composite, alphanumeric
- 23.Radiology test (Weak)
 - Test_ID: key, numeric
 - Date: multivalue, timestamp
 - Type: composite, alphanumeric
- 24.Blood test (Weak)
 - Test_ID: key, numeric
 - Result: composite, alphanumeric
 - Name: composite, alphanumeric

25. Bank Account (Strong)

- Acc_number: key, numeric
- Billing_address: composite, alphanumeric
- Acc_holder_name: composite, alphanumeric

26. Billing information (Strong)

- Bill_ID: key, numeric
- Payment_method: composite, alphanumeric
- Status: Boolean

27. System configuration settings (Strong)

- Setting_ID: key, numeric
- Setting_name: composite, alphanumeric
- Password: composite, alphanumeric

28. Account permissions associative entity (Weak)

- Permission_ID: key, numeric
- Type: composite, alphanumeric
- User_role_ID: key, numeric

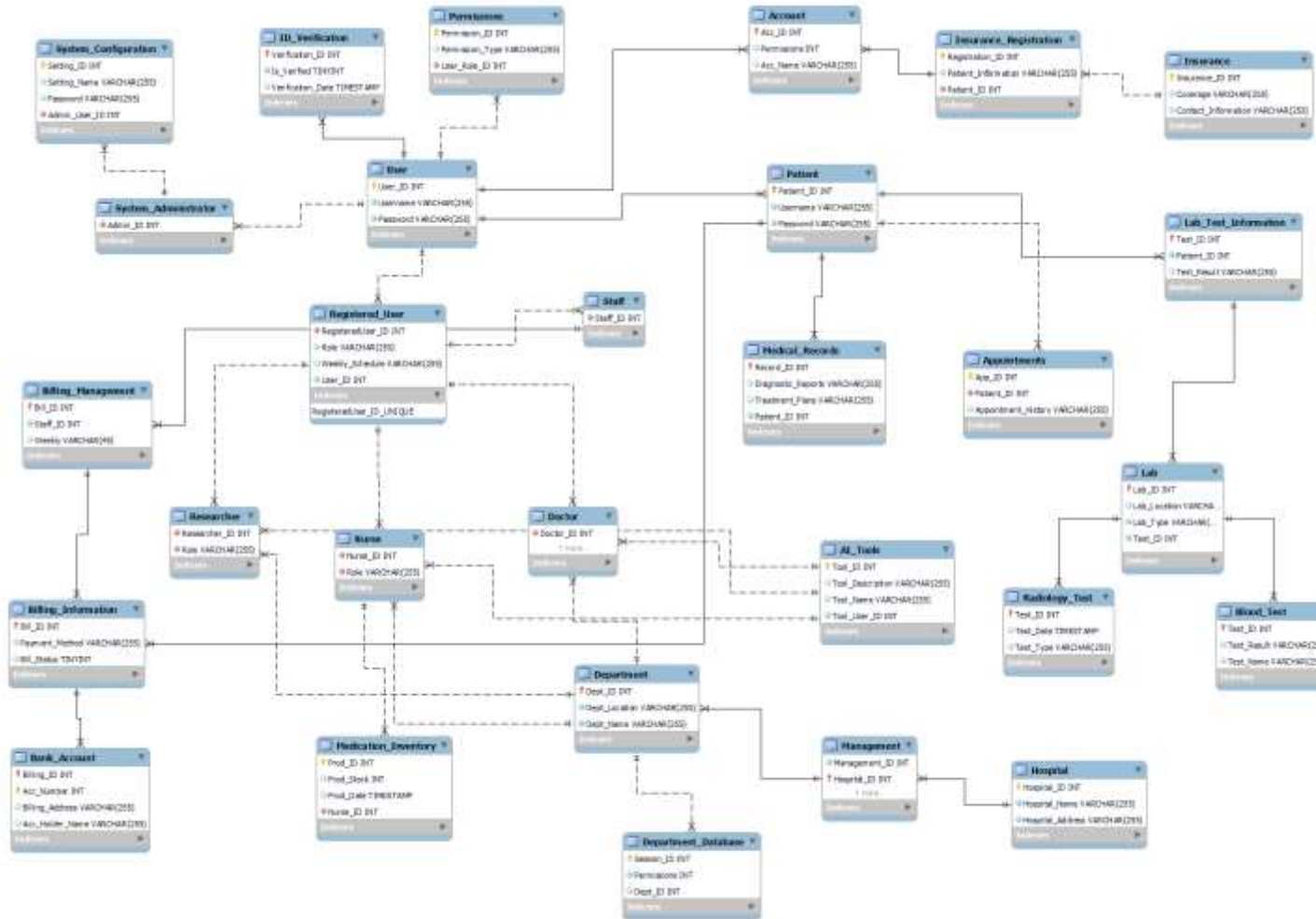
TABLE	FK	ON DELETE	ON UPDATE	COMMENT
Registered_User	FK_Reg_User_ID	CASCADE	CASCADE	If a user is deleted, then the account from that user must be deleted as well
Patient	FK_Patient_User_ID	CASCADE	CASCADE	If a user is deleted, then the patient record should be deleted as well
System_Administrator	FK_Admin_User	CASCADE	CASCADE	If a user is deleted, then the admin record should be deleted as well
Management	FK_Hospital	CASCADE	CASCADE	If a hospital is deleted, then related management records should be deleted as well
Department	FK_Management	CASCADE	CASCADE	If a department is deleted, then related management records should be deleted as well
Researcher	FK_ResearcherID_RegUser	CASCADE	CASCADE	If a user is deleted, then the researcher record should be deleted as well
Doctor	FK_DoctorID_RegUser	CASCADE	CASCADE	If a user is deleted, then the doctor record

				should be deleted as well
Nurse	FK_NurseID_RegUser	CASCADE	CASCADE	If a user is deleted, then the nurse record should be deleted as well
Appointments	FK_App_ID	CASCADE	CASCADE	If a patient is deleted, then related appointments should be deleted as well
Staff	FK_Staff_User_ID	CASCADE	CASCADE	If a user is deleted, then the staff record should be deleted as well
Lab_Test_Information	FK_Patient_Test_ID	NO ACTION	CASCADE	When a patient is deleted, the lab test information should not be deleted, but the references are updated
Lab	FK_Test_Info	NO ACTION	CASCADE	When lab test information is deleted, related lab records should be updated
Radiology_Test	FK_Lab_RT	NO ACTION	CASCADE	When lab records are deleted, related radiology test records should be updated
Blood_Test	FK_Lab_BT	NO ACTION	CASCADE	When lab records are deleted, related blood test records

				should be updated
Medical_Records	FK_Record_ID	NO ACTION	CASCADE	When a patient is deleted, the medical records should not be deleted, but the references are updated
Insurance_Registration	FK_Insurance	CASCADE	CASCADE	If a user is deleted, then the insurance registration should be deleted as well
Account	FK_Acc_ID	CASCADE	CASCADE	If a user is deleted, then the account should be deleted as well
Department_Database	FK_Dept_ID	NO ACTION	CASCADE	When a department is deleted, the department database should not be deleted, but the references are updated
ID_Verification	FK_Verification_ID	CASCADE	CASCADE	If a user is deleted, then the ID verification record should be deleted as well
Medication_Inventory	FK_Nurse_Access	NO ACTION	CASCADE	When a nurse is deleted, the medication inventory should not be deleted, but the references are updated
Billing_Management	FK_Managed_By_Staff	CASCADE	CASCADE	If a staff member is

				deleted, then the related billing management record should be deleted as well
Billing_Information	FK_Bill_ID	CASCADE	CASCADE	If a patient is deleted, then the related billing information should be deleted as well
Bank_Account	FK_Billing_Info	CASCADE	CASCADE	If billing information is deleted, the related bank account records should be deleted as well
System_Configuration	FK_Config_By_Admin	NO ACTION	NO ACTION	When an admin is deleted, the system configuration should not be deleted, and no action is taken on update
Permissions	FK_Perm_ID	CASCADE	CASCADE	If a user is deleted, then the related permissions should be deleted as well

PS: Most of the cardinalities are incorrect, I tried to fix it for days but I couldn't.



Non-functional requirements:

1-Performance

- 1.1 The database system shall be able to handle high volume of concurrent users.
- 1.2 The database system shall handle many queries in a short time.
- 1.3 The database system shall have fast response time in data upload.
- 1.4 The database system shall have fast response time in data retrieval.

2-Security

- 2.1 The database system shall have role-based authorization requirements.
- 2.2 The database system shall only allow access to registered users.
- 2.3 The database system shall authenticate user login by username and password.
- 2.4 The database shall encrypt user data during transmission.

3-Usability

- 3.1 The database system shall be easy to navigate.
- 3.2 The database system shall be accessible to those with audio or visual imparities.

4-Maintainability

- 4.1 The database system shall be easy to maintain.
- 4.2 The database system shall be easy to update.
- 4.3 The database system shall be written in readable and understandable code.

5-Scalability

- 5.1 The database system shall scale to handle increased load.
- 5.2 The database system shall scale horizontally if necessary.
- 5.3 The database system shall scale vertically if necessary.
- 5.4 The database system shall have a mechanism for data archiving to manage historical data.

6-Storage

- 6.1 The database system shall handle large amounts of data.
- 6.2 The database system shall frequently back-up the data.
- 6.3 The database system shall have a recovery storage.
- 6.4 The database system shall compress data for optimizing storage space.
- 6.5 The database shall encrypt sensitive user information.