

Hospital management system

Digital Egypt Pioneers Initiative (DEPI)

Team Members:
1- عمر امجد عبدالعزيز
2- كريم محمد حلمي
3- كرلس هاني وديع
4- يوسف شرف
5- زياد ماهر

1. Project Planning & Management

1.1.a Purpose

The purpose of the Hospital Management System is to automate the registration, appointment booking, and management of patient and doctor records within a hospital. This proposal outlines the development of a comprehensive Hospital Management System (HMS), aimed at improving the process of scheduling doctor appointments. The system will allow patients to view the schedules of various doctors at the hospital and book an available slot online. Once a patient books a time slot, that slot will be reserved and unavailable for other patients, ensuring a smooth and efficient

The system will streamline the hospital's operations by providing a centralized platform for appointment scheduling, improving patient satisfaction and reducing manual errors

1.1.b Scope

The system facilitates:

- Patient registration and management.
- Appointment scheduling with doctors.
- Secure storage and retrieval of patient and doctor details.
- Role-based access control for administrators, doctors, and patients.

1.1.c Objectives

- **Streamline Appointment Scheduling:** Allow patients to access the hospital's website, view available time slots for different doctors, and book appointments in real-time.
- **Ensure Time Slot Availability:** Ensure that once a time slot is booked by a patient, it becomes unavailable for other patients, preventing double bookings.
- **Improve Patient Convenience:** Patients will be able to book, reschedule, or cancel appointments without having to call or visit the hospital in person.
- **Enhance Hospital Management Efficiency:** Reduce administrative workload by automating appointment scheduling and management.

1.2

- **System Uptime**

The percentage of time the system is operational and available to users without interruptions or downtime.

Why It's Important: Ensures that the system is reliable and always accessible, allowing patients to book appointments at any time.

How to Measure: Monitor the system's operational status and track any downtimes or outages over a period (e.g., monthly).

- **Response Time / Load Time**

The time it takes for the system to respond to a user's action, such as loading a doctor's schedule, confirming a booking, or updating appointment information.

Target: Response time should be under few seconds for each user interaction.

Why It's Important: Fast response times are essential to ensure a smooth user experience, especially when booking appointments during peak hours.

How to Measure: Monitor the time it takes for key pages or actions (e.g., booking, schedule view) to load on the platform

- **User Adoption Rate**

The percentage of patients who actively use the system to book appointments after registration.

Target: 70-80% adoption rate within the first 3 months after launch.

Why It's Important: High adoption rate indicates that the system is user-friendly and meets the needs of the patients, showing that it's an effective tool for appointment management.

How to Measure: Count the number of users who successfully book appointments through the system versus the total number of registered users.

2. Requirements Gathering

2.1 System Stakeholders

- **Admin:** Manages users, doctors, and appointments.
- **Doctor:** Views patient details and manages appointments.
- **Patient:** Registers, books, and requests and updates appointments.

2.2 Users' Objectives

- **Admin:** Maintain system security and data integrity.
- **Doctor:** Access patient history and manage schedules.
- **Patient:** Book and manage appointments.

2.3 Functional Requirements

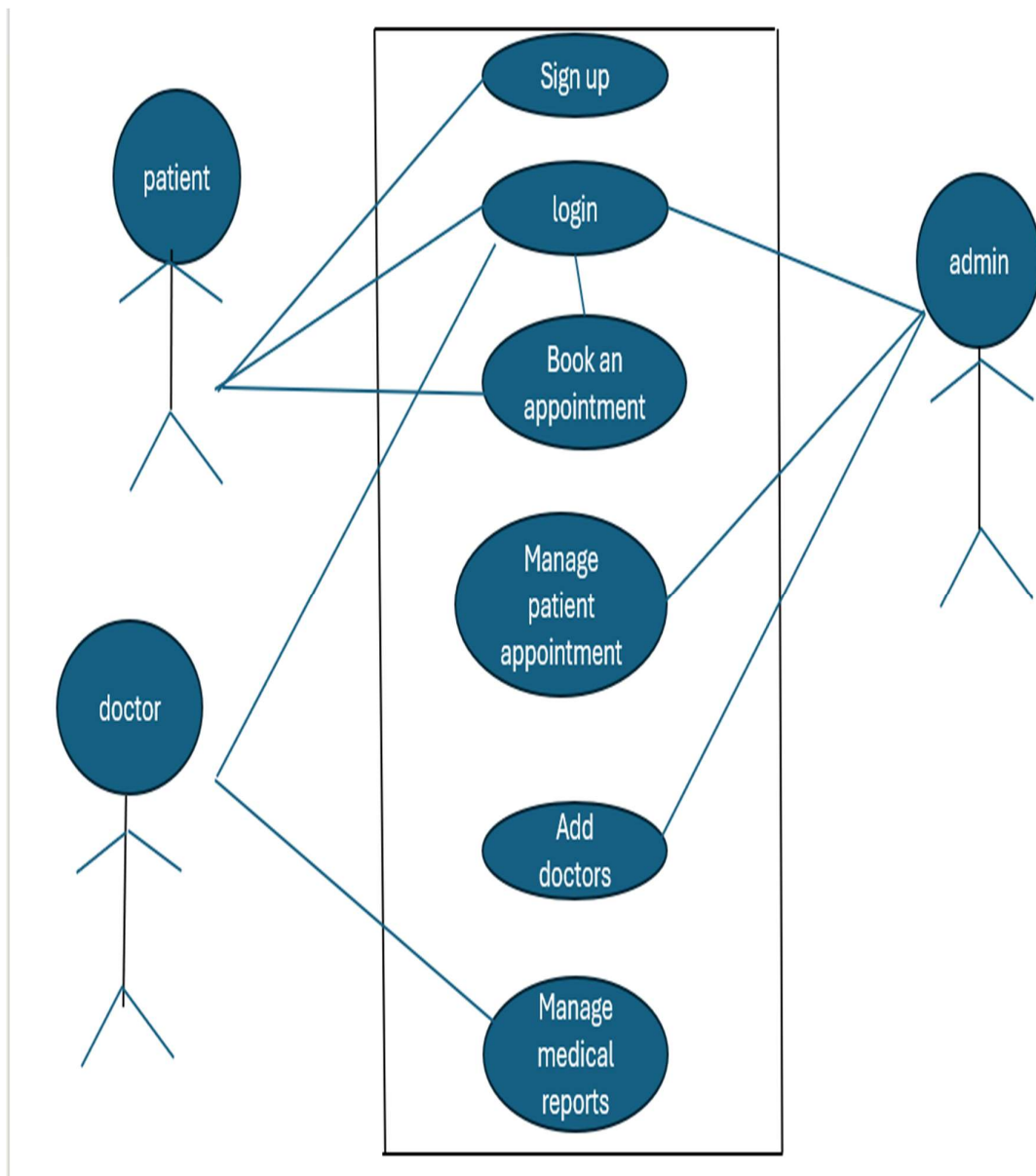
- Unique patient ID generation.
- Patient record storage and retrieval.
- Doctor profile management.
- Appointment scheduling system.

2.4 Non-Functional Requirements

- **Performance:** Response time < 1s for database queries.
- **Security:** Only authorized users can access system components.
- **Availability:** System should be available 24/7.

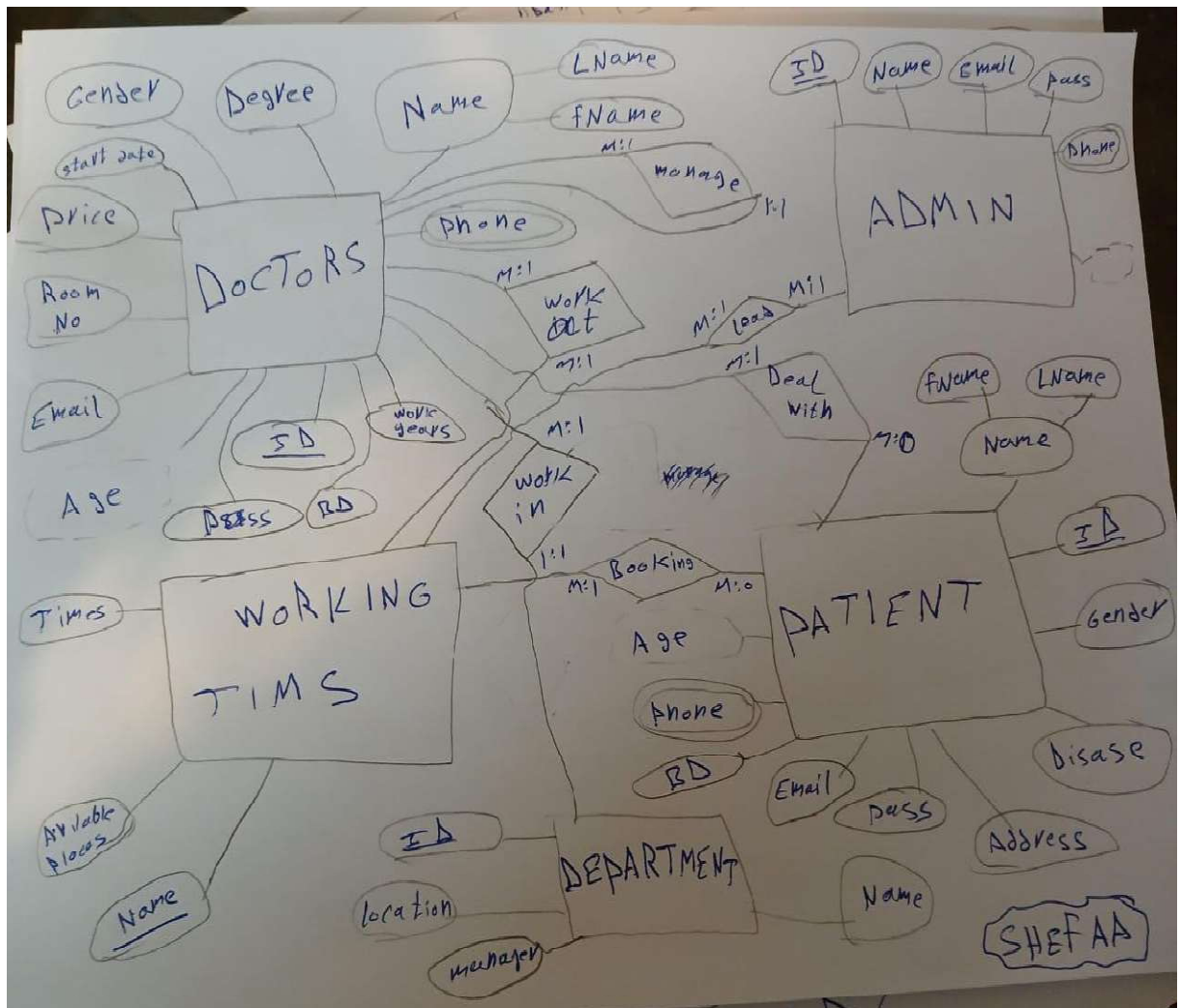
3. System Analysis & Design

3.1 use case diagram

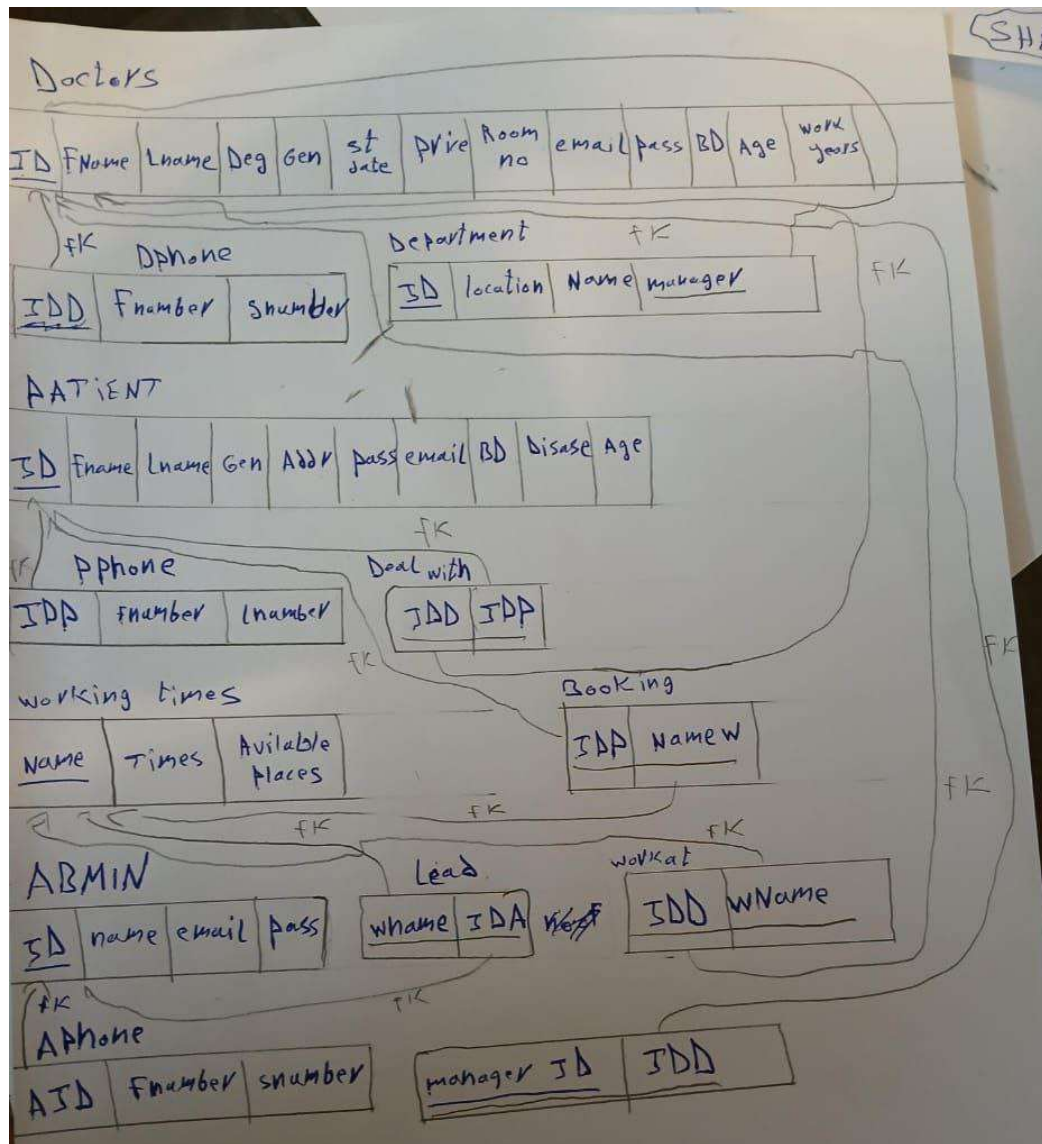


3.2 Database Design & Data Modeling

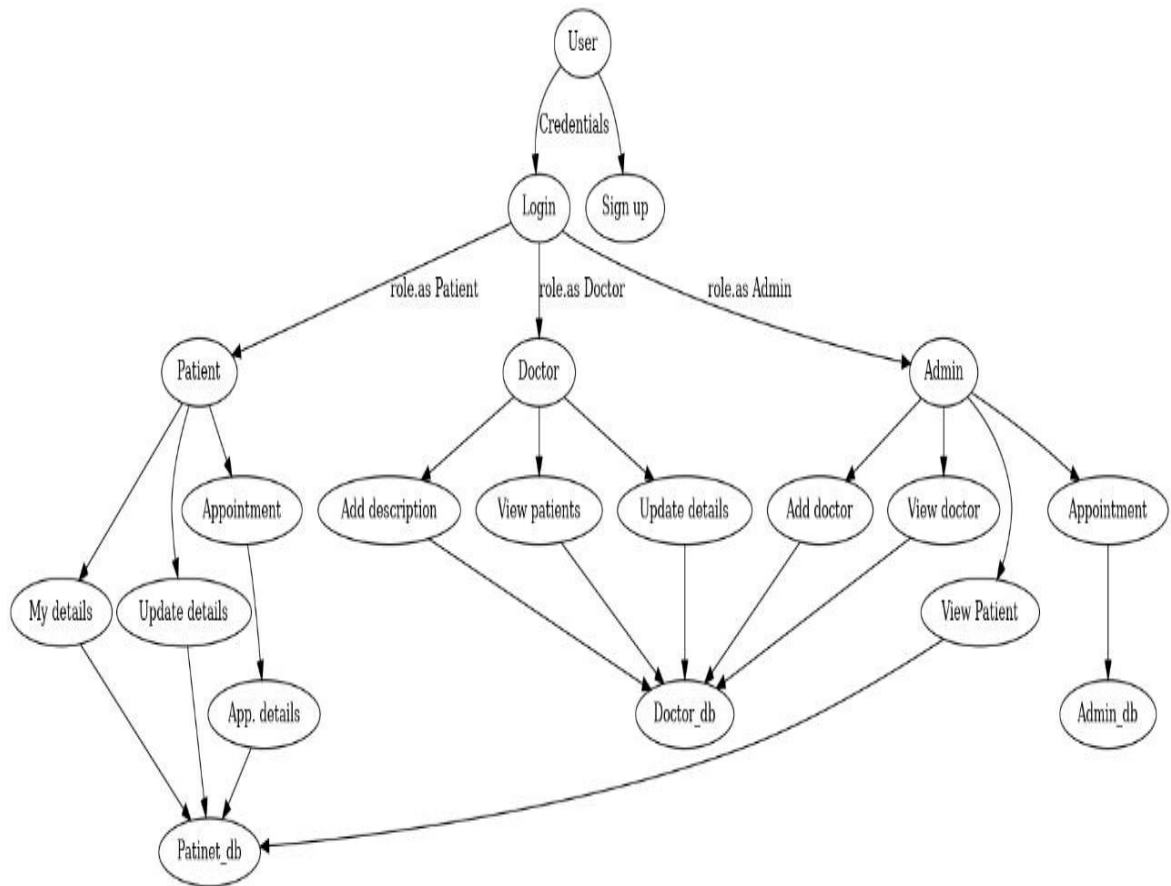
3.2.a ERD



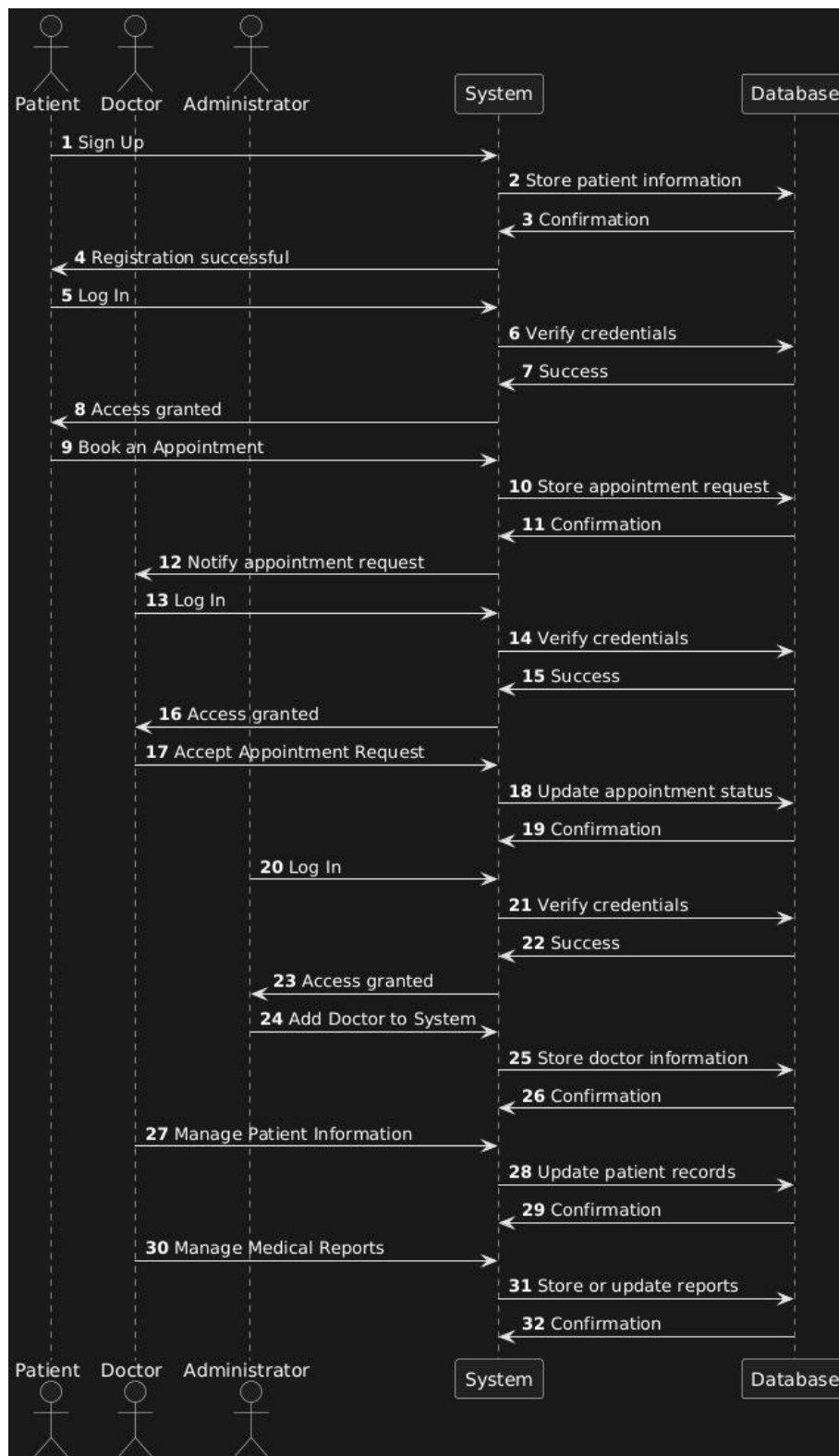
3.2.b Mapping



3.3 DFD (Data Flow Diagram)



3.4 Sequence Diagrams



4. System Deployment & Integration

The system follows a client-server model with a web-based interface for access. It includes:

- Front-end: Java script & HTML & CSS.
- Back-end: C# & Entity framework core & ASP .net core.
- Database: SQL SERVER.