Digital Healthcare Platform

Software Design Document

Yarin Ellawendy, Ahmed Elzahaby, Raphael Demian, Karim Abolghar, Ali Hammad, Omar Hany, Jacques Elia
Section 1

TABLE OF CONTENTS

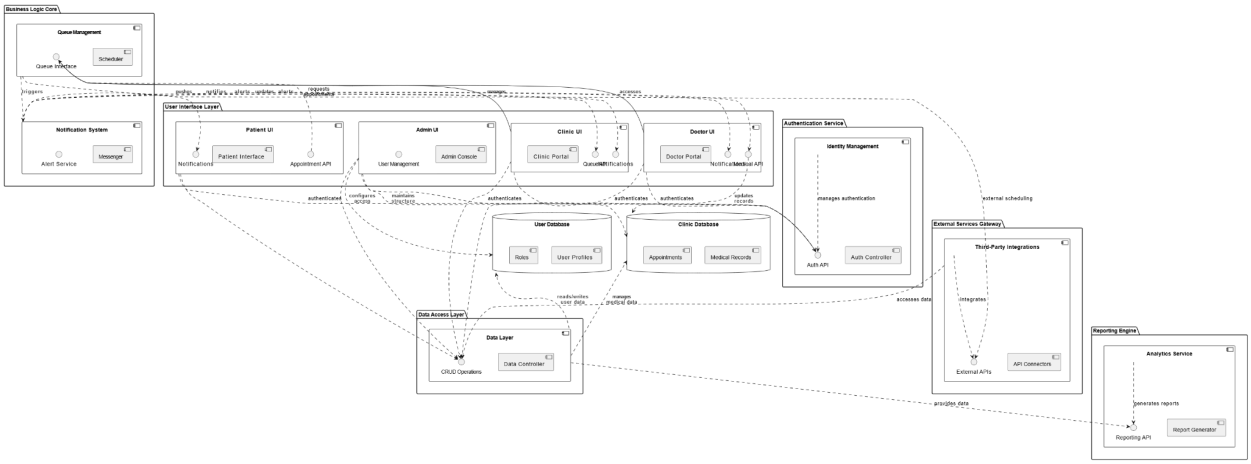
3. System Architecture

3.1 Architectural Design

The digital healthcare platform is designed using a layered architecture where all components perform one function and are separate to ensure separation, cohesion while retaining good interaction and enabling scalability and maintainability. The different components are the following:

1. **User Interface Layer**
    - Patient mobile application to always be accessible
        i. Signup and Login
        ii. Enter medical history
        iii. Browse clinics
        iv. Read and write reviews
        v. Book appointment
        vi. Follow queue
    - Doctor mobile application to perform simple tasks quickly
        i. Prescribe medication
        ii. Request Follow up
        iii. Follow queue
    - Clinic desktop application to perform complex tasks effectively
        i. Manage patient databases
        ii. Manage queue
        iii. Modify working hours
    - Admin desktop application to monitor effectively
        i. Verify and monitor clinics
        ii. View and monitor reviews and reports
        iii. Communication with users regarding reports
2. **Business Logic Core**
    -
    - Queue Management
    - Notification system
    - Handle core business processes

3. **Data Access Layer**
    - Data layer component, responsible for all database interactions
4. **External Services Gateway**
    - Integration of external APIs
        i. AI chatbot API
5. **Reporting Engine**
    - Data analytics
6. **Authentication Service**
    - Centralized user authentication
        i. User management API
        ii. Two-Factor Authentication



3.2 Decomposition Description

Primary Functions

1. **User Management**
    - Handles user registration, login, and authentication (including 2FA).
    - Manages patient profiles, including medical history and personal information.
    - Allows clinics and doctors to sign up and manage their accounts.
2. **Clinic Management**

- Enables clinics to modify working hours, manage queues, and send receipts.
- Provides functionality to check visit types (e.g., follow-ups or new consultations).
- Displays clinic information such as location, availability, ratings, and insurance compatibility.

3. **Appointment Scheduling**
    - Allows patients to view available slots based on clinic schedules.
    - Facilitates booking, canceling, or rescheduling appointments.
    - Integrates live queue management for real-time updates on wait times.
4. **Medical History Management**
    - Centralizes patient medical records for easy access by both patients and healthcare providers.
    - Enables doctors to view patient history divided into specific fields (e.g., prescriptions, diagnoses).
    - Allows patients to upload medical history (scan or type) and view past appointments.
5. **Notification System**
    - Sends reminders for appointments, medication times, and queue updates.
    - Alerts doctors and secretaries when patients register or cancel appointments.
6. **Review and Report System**
    - Patients can write reviews about clinics/doctors based on their experience.
    - Doctors can view reviews related to their services.
    - Admins monitor reviews and reports for quality control.
7. **Security Measures**
    - Ensures encryption of sensitive data such as medical history and personal information.
    - Implements 2FA for secure user authentication.

Sub-Functions

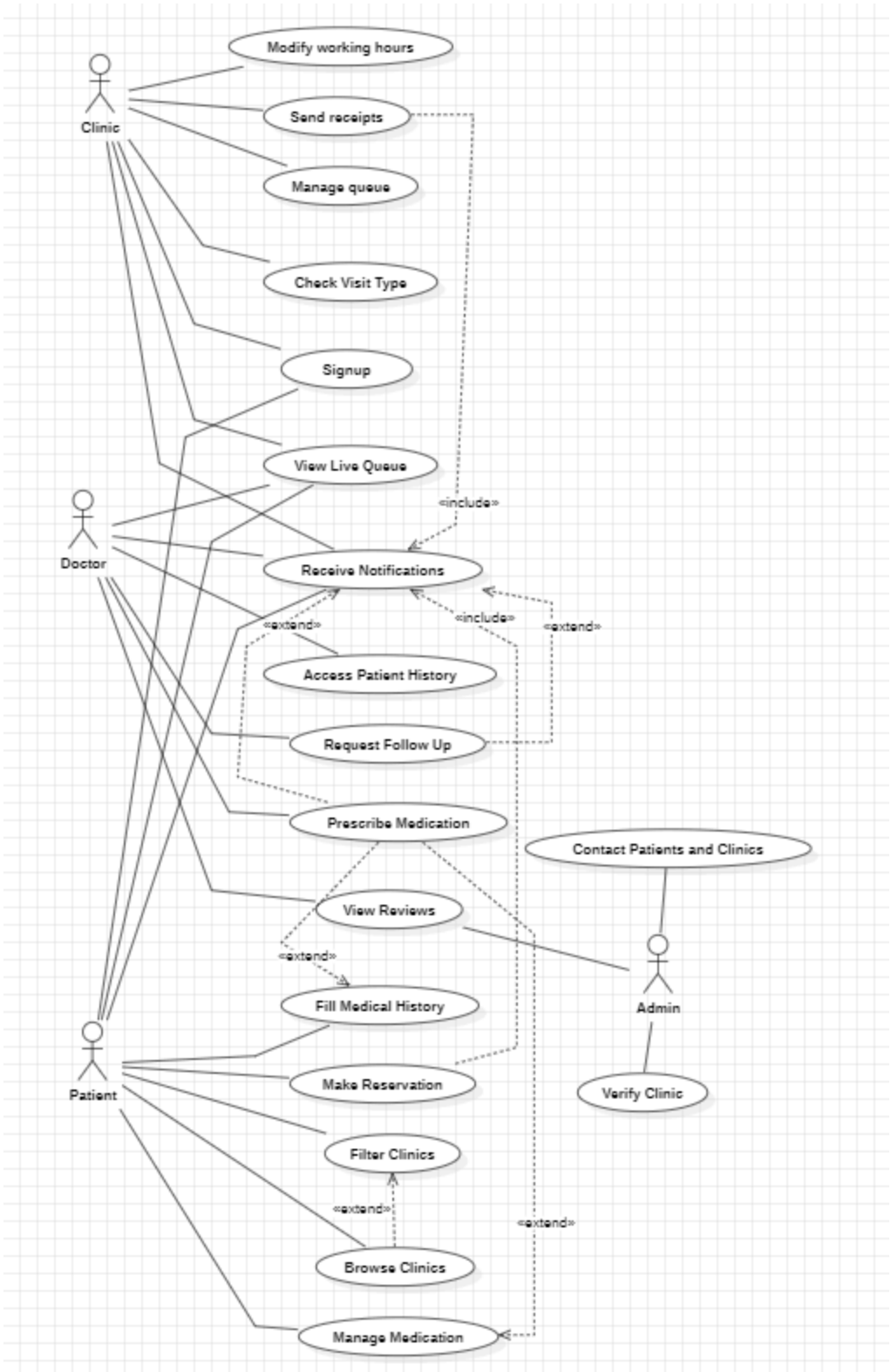
1. **User Management API**
    - Handles all user-related operations:
        - Patient sign-up/login
        - Doctor/clinic registration
        - Profile updates
    - Interacts with the database to retrieve user data securely.
2. **Queue Management API**
    - Manages real-time queue updates:
        - Adds/removes patients from the queue.
        - Displays live queue status for patients and clinics.
    - Connects with clinic schedules to ensure synchronization.
3. **Database Components**
    - User Database: Stores patient profiles, doctor accounts, admin details.
    - Clinic Database: Maintains clinic information like location, working hours, ratings.
    - Medical History Database: Holds encrypted records of patient medical histories.
    - Queue Database: Tracks real-time queue statuses per clinic.
4. **Functional Modules**
    - Smart Clinic Finder: Filters clinics by proximity, ratings, availability, insurance compatibility.
    - Live Queue System: Provides real-time updates on wait times for both patients and clinics.
    - Medication Reminder: Alerts patients about prescribed medication schedules.

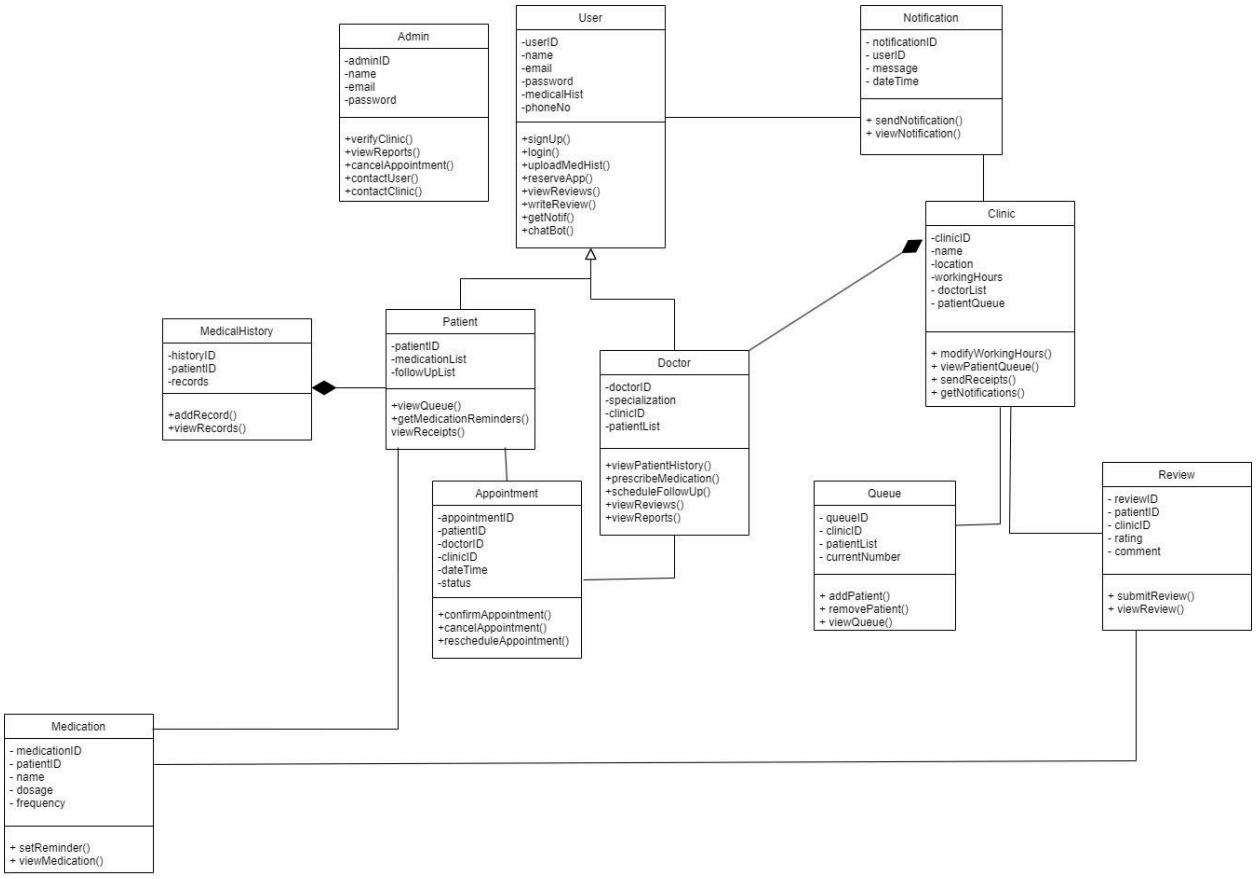
Relationships Between Components

1. **User Interface Components:**
    - The Patient App interacts with the User Management API for sign-up/login functionalities.
    - The Doctor Interface accesses patient history via the Medical History Database.

- The Secretary Interface manages queues via the Queue Management API.
- The Admin Interface oversees user activities and verifies clinics through the User Management API.

2. **API Interactions:**
    - The User Management API communicates with the User Database to authenticate users securely using 2FA.
    - The Queue Management API updates queue statuses in the Queue Database based on real-time patient registrations or cancellations.
3. **Data Flow:**
    - When a patient books an appointment via the Patient App:
        - The system checks availability in the Clinic Database.
        - Updates the Appointment Database with booking details.
        - Sends notifications through the Notification System.
4. **Security Measures:**
    - All sensitive data (e.g., medical history) is encrypted before storage in databases.
    - 2FA ensures secure access to user accounts across platforms.


Modify working hours
Clinic
Send receipts
Manage queue
Check Visit Type
Signup
View Live Queue
«include»
Doctor
Receive Notifications
«extend»
«include»
«extend»
Access Patient History
Request Follow Up
Prescribe Medication
Contact Patients and Clinics
View Reviews
«extend»
Admin
Fill Medical History
Patient
Make Reservation
Verify Clinic
Filter Clinics
«extend»
«extend»
Browse Clinics
Manage Medication



3.3 Design Rationale

All decisions are based on our commitment to building an accessible, scalable, and secure system which will satisfy needs both today and tomorrow. Medical data being of sensitive nature, regulatory compliance with requirements is highest on our priority list. This influences data encryption, access control, and audit trails. In terms of performance and scalability, we have utilized a modularity-focused service-oriented architecture (SOA). With the system divided into standalone microservices, we can scale individually as and when required.

Smooth data communication is required for healthcare systems. Adherence to standards like HL7 and FHIR allows our platform to get easily integrated into other systems like Electronic Health Records (EHRs) and third-party tools. Security is weaved into every layer of the platform. End-to-end

encryption, role-based access controls, and secure APIs are used by us so that patient information does not get leaked and accessed by the wrong persons.

The system has a user-centric design methodology. Simple front-ends, multi-device compatibility, and extendable dashboards enable clinicians and patients to operate it seamlessly.

Clean module-based upgradable architecture can be made long-term maintenance-easy. The solution also fosters technology heterogeneity where a service uses an appropriate stack of technologies.

Cloud hosting of the platform supports scalability, high availability, and disaster recovery at reduced investment cost in high-cost on-premises infrastructure.

A well-normalized database schema removes data inconsistency, reduces redundancy, and promotes query performance optimization. Especially dealing with massive amounts of sensitive patient health information. We have data encryption during transit and in storage, secure access controls, and periodic security audits. They protect patients' data and permit compliance with law.

The UI is clean and simple. By including real end-users in the process of design, we are ensuring the platform is solving real problems of the clinicians, administrators, and patients in real life.

We make the interface responsive on devices (desktops, smartphones, tablets) such that the experience is the same regardless of the device used.

Security controls have to be suitable, but at times they also make the issue worse and worsen it for the users. Our design tries to strike a balance between the two by including security in a way that the end user disruptions are minimized to a bare minimum.

Although a microservices architecture assists in maintainability and scalability, it comes with the threats of increased deployment and service-to-service interaction complexity. We mitigate these threats through end-to-end monitoring, automated testing, and intelligent tooling for orchestration. Our design style is a purposeful balance of innovation, reliability, safety, and usability.