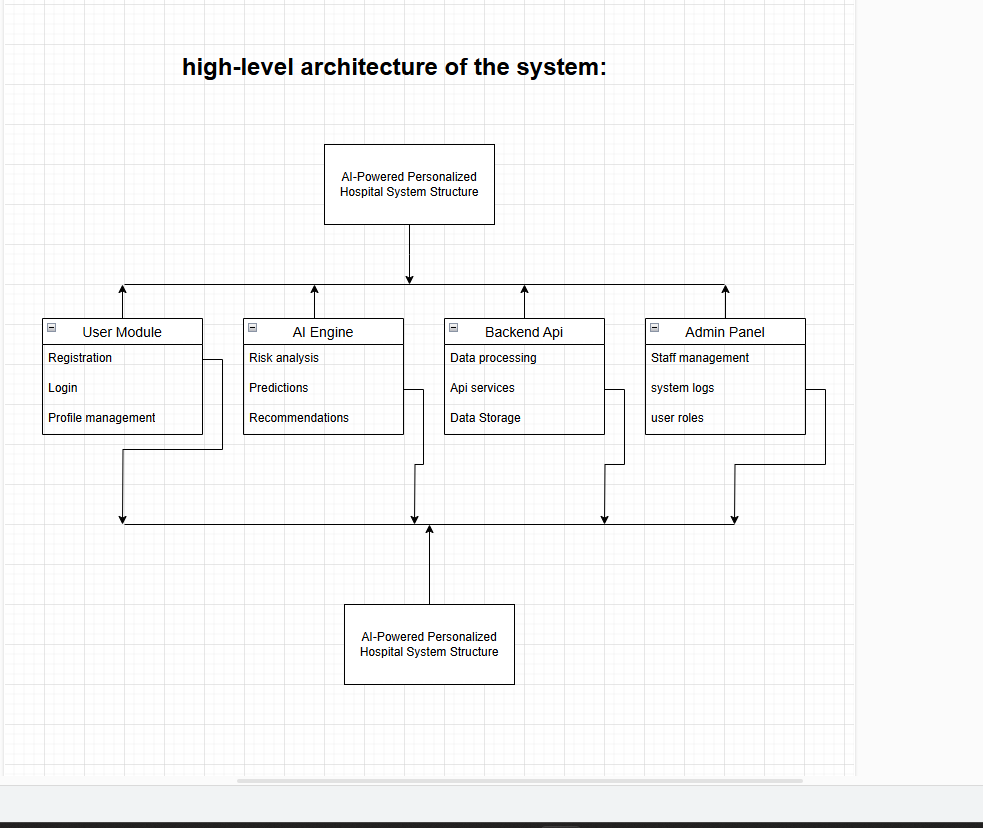
**Part 1**



**Subsystem Descriptions**

**User Module (Frontend - Next.js)**

Allow patients and healthcare providers to interact with the system.

Includes user authentication, profile management, and appointment scheduling.

**AI Engine**

Processes patient data for risk analysis and personalized healthcare recommendations.

Uses machine learning algorithms to predict potential health risks.

Provides insights and recommendations for patients and doctors.

**Backend API (Django)**

Serves as the communication layer between the frontend and the database.

Handles user authentication, appointment requests, and data processing.

Ensures security and scalability.

**Admin Panel**

Allows hospital administrators to manage users, staff, and system configurations.

Tracks system logs and security compliance.

**Database Layer (PostgreSQL/MongoDB)**

Stores patient records, medical history, and AI-generated recommendations.

Ensures data integrity and efficient retrieval for real-time operations.

**Data Flow Process**

The user logs in via the frontend (Next.js).

The backend API (Django) authenticates the user and retrieves necessary data.

The AI Engine processes health data and generates insights.

The system stores or updates data in the database layer.

The Admin Panel monitors system operations and manages users.

**Part 2**

**1. Functional Requirements**

Functional requirements define the specific behavior or functions of the system.

**1.1 User Management**

Users must be able to register and log in securely.

For Registration:

Patient Registration - Required Fields

* Personal Info: Full Name, Email (OTP), Phone, DOB, Gender, Password
* Medical Info: Blood Group, Allergies, Conditions, Medications, Family History, BMI
* Emergency Contact: Name, Phone, Relationship
* System Info: Auto-generated Patient ID, 2FA (optional)

Doctor Registration - Required Fields

* Personal Info: Full Name, Email (OTP), Phone, DOB, Gender, Password
* Professional Info: Medical License (Admin Approval), Specialization, Experience, Hospital/Clinic Name, Availability, Consultation Fee
* System Info: Auto-generated Doctor ID, 2FA (optional)

Admin Registration - Required Fields

* Personal Info: Full Name, Email (OTP), Phone, Password
* Role Info: Admin Type (Super Admin / Hospital Admin)
* System Info: Auto-generated Admin ID, 2FA (mandatory)

For Login:

Login Fields for Patients:

* Email (Registered email)
* Password (Encrypted)
* Two-Factor Authentication (2FA) Code (if enabled)
* Patient ID (Auto-generated at registration, used internally but not required for login)

For doctor and admins:

* Email (Registered email)
* Password (Encrypted)
* Two-Factor Authentication (2FA) Code (if enabled)

Profile management should allow users to update personal details, medical history, and preferences. Allergies must be recorded before suggesting a diet plan.

Role-based access control (patients, doctors, administrators).

**1.2 AI-Powered Health Risk Analysis**

The system should analyze patient health data to predict potential health risks.

AI algorithms should provide recommendations for lifestyle changes and preventive care.

**1.3 Personalized Recommendations**

AI-based diet recommendations are generated while considering user allergies.

Exercise and lifestyle suggestions based on user activity and goals.

**1.4 Appointment Scheduling**

Patients should be able to schedule, reschedule, or cancel appointments.

Doctors should receive real-time notifications of new bookings.

**1.5 Real-Time Alerts and Notifications**

The system should send health alerts and medication reminders.

Doctors should be alerted in case of critical health risk predictions.

**1.6 Electronic Health Records (EHR) Management**

Secure storage and retrieval of medical records.

Authorized personnel can update and access records.

**1.7 Payment and Billing**

Integration with third-party payment gateways for secure transactions.

Patients should be able to view and pay bills online.

**1.8 Admin Panel for Hospital Management**

System administrators can manage users, doctors, and reports.

Access to system logs and security compliance monitoring.

**1.9 Data Management and Retention**

Inactive user data is automatically cleared from the database after 90 days of inactivity.

**Security Features**

Two-factor authentication (2FA) ensures secure account access.

**2. Non-Functional Requirements**

These requirements define system performance, security, usability, and other quality attributes.

**2.1 Performance Requirements**

The system should support simultaneous access by at least 500 users without degradation.

Response time for API calls should be less than 2 seconds.

**2.2 Security Requirements**

All patient data must be encrypted at rest and in transit (AES-256).

Role-based access control to prevent unauthorized access.

System should comply with HIPAA and GDPR regulations.

**2.3 Usability Requirements**

The system should have an intuitive user interface for both patients and medical staff.

Mobile-responsive design for accessibility on all devices.

**2.4 Reliability and Availability**

The system must have 99.9% uptime and support failover mechanisms.

Automatic data backup and disaster recovery procedures in place.

**2.5 Scalability**

Architecture should allow easy scaling to support future growth in users and features.

Cloud-based deployment to handle increased loads dynamically.

**3. Interfaces**

**3.1 User Interface (UI)**

Web-based frontend (Next.js) with a dashboard for patients and doctors.

Mobile-friendly design for accessing health records and booking appointments.

**3.2 Software Interfaces**

Backend API (Django) will expose endpoints for authentication, appointments, and AI insights.

Integration with third-party EHR systems and payment gateways.

**3.3 Hardware Interfaces**

The system should support integration with wearable health devices (e.g., smartwatches, fitness trackers).

Compatibility with hospital equipment (e.g., digital thermometers, ECG monitors).

**3.4 Communication Interfaces**

Secure email and SMS notifications for appointment reminders and health alerts.

API-based integration with hospital management systems for seamless data exchange.

**Part 3**

**Project Implementation Plan: Three Development Phases**

**Phase 1: Core System Development (MVP - Minimum Viable Product)**

Objective: Establish the fundamental functionalities required for the system to function at a basic level.

Priority: High criticality features that are essential for the system's operation.

**Requirements Implemented in Phase 1:**

User Authentication & Profile Management

Secure registration and login with role-based access control (patients, doctors, admins).

Profile management for storing user details (medical history, preferences).

**Health Risk Analysis**

AI-powered risk assessment based on user-provided health data.

Integration with basic health indicators (BMI, blood pressure).

**Appointment Scheduling**

Calendar-based booking system with time-slot availability for doctors.

Automatic notifications for confirmed, rescheduled, or canceled appointments.

**Basic User Interface & Database Setup**

Web-based UI (Next.js) for user interactions.

Backend setup (Django) with PostgreSQL database for storing medical records.

**Deliverables for Phase 1:**

Functional user authentication and profile management.

Initial AI-driven health risk assessment model.

Working appointment scheduling system.

Deployed MVP with essential UI/UX components.

**Phase 2: Advanced Features & AI Integration**

Objective: Enhance the system by integrating AI-driven recommendations and improving usability.

Priority: Medium criticality features that improve personalization and automation.

**Requirements Implemented in Phase 2:**

**Personalized Health Recommendations**

AI-driven health insights based on patient history and risk factors.

Customized lifestyle and medication suggestions.

**Real-time Alerts & Notifications**

Alerts for high-risk patients requiring urgent consultation.

Reminders for scheduled appointments and medication intake.

**Doctor & Patient Communication Module**

Secure chat functionality for doctor-patient interactions.

Video consultation integration for telehealth services.

**System Performance Optimization**

Backend performance tuning for faster data processing.

Enhanced security measures (data encryption, access control).

**Deliverables for Phase 2:**

AI-powered personalized recommendations fully functional.

Real-time notifications and alerts integrated.

Secure communication module between doctors and patients.

Improved system performance and security enhancements.

**Phase 3: Scalability & External System Integration**

Objective: Expand the system to support larger-scale operations and interoperability with other healthcare platforms.

Priority: Low criticality features that improve scalability, external integration, and reporting.

**Requirements Implemented in Phase 3:**

**Integration with Wearable Devices & EHR Systems**

Synchronization with wearable health trackers (Fitbit, Apple Health).

API integration with existing Electronic Health Records (EHR) systems.

**Advanced Analytics & Reporting**

AI-based trend analysis for predicting disease progression.

Customizable reports for hospital administrators and healthcare providers.

**System Load Testing & Scalability Improvements**

Stress testing to handle concurrent users efficiently.

Deployment of cloud-based infrastructure for scalability.

**Deliverables for Phase 3:**

Seamless integration with wearable devices and EHR systems.

AI-driven predictive analytics and custom reporting.

Fully functional mobile app for enhanced accessibility.

Optimized system performance for high scalability.

**Part 4**

|  |  |  |  |
| --- | --- | --- | --- |
| Member Name | Contribution Description | Overall Contribution (%) | Note (if applicable) |
| Rohit Ibrahimpatnam | Led the overall system design, structured the phased development plan, and outlined key system components. | 14% | Ensured the report aligned with system objectives. |
| Anusha Pujari | Coordinated report structure, project scope definition, and risk management. | 14% | Assisted in organizing the implementation timeline. |
| Harshini Sai Sangadi | Developed system architecture, backend framework, and API integrations. | 14% | Project Lead: Provided insights into backend infrastructure and security. |
| Ram Gopal Anne | Authored AI-driven health recommendations, machine learning integration, and dataset analysis. | 14% | Focused on AI functionality and data modeling. |
| Rehamath Shaik | Wrote UI/UX design specifications, frontend requirements, and user interface functionality. | 14% | Led frontend design documentation and usability considerations. |
| Sai Murali Kilaru | Handled database design, cloud deployment, scalability strategies, and performance optimizations. | 14% | Focused on cloud infrastructure and system efficiency. |
| Vinay Kumar Parvathini | Contributed to testing strategies, quality assurance, and security testing documentation. | 14% | Led QA and ensured system robustness. |