Project - High Level Design

“Healthcare Planning Assistant Agent”

“Datagami-Skill Based Course”

Institution Name: Medicaps University – Datagami Skill Based Course

Sr no	Student Name	Enrolment Number
1.	AAKANKSH THAKUR	EN23CS3T1002
2.	KRATI PORWAL	EN22CS301515
3.	KHUSHI JAISWAL	EN22CS301502
4.	KULDEEP SINGH SISODIYA	EN22CS301530
5.	KANAK VYAS	EN22CS301478

Group Name: Group 02D5

Project Number: AAI-02

Industry Mentor Name: Aashruti Shah

University Mentor Name:Prof. Maya Yadav Baniya

Academic Year:2026

"Table of Contents"

1. Introduction.	3
1.1. Scope of the document. 1.2. Intended Audience 1.3. System overview.	
2. System Design. 2.1. Application Design 2.2. Process Flow. 2.3. Information Flow. 2.4. Components Design 2.5. Key Design Considerations 2.6. API Catalogue.	4
3. Data Design. 3.1. Data Model 3.2. Data Access Mechanism 3.3. Data Retention Policies 3.4. Data Migration	11
4. Interfaces	15
5. State and Session Management	16
6. Caching	17
7. Non-Functional Requirements 7.1. Security Aspects 7.2. Performance Aspects 7.3. References	18

Introduction-

1.1 Scope of the document-

The high-level design concepts for an AI-driven healthcare assistant system. The system is engineered to recommend personalized treatments and hospital options, focusing on complex conditions like cancer and chronic diseases such as blood pressure management, Diabetes, Hypertension. The design addresses healthcare-specific requirements, patient data privacy, clinical accuracy, accessibility, scalability, and secure technical architecture for sensitive health information. The design incorporates healthcare-specific requirements to ensure clinical reliability, regulatory compliance, and ethical data handling. It emphasizes robust patient data privacy mechanisms, secure data storage, and encrypted communication channels to protect sensitive health information.

1.2 Intended Audience-

- **Healthcare Providers & Hospitals:** Physicians, specialists, and healthcare institutions utilize the system to streamline patient referrals, support clinical decision-making, and deliver accurate, validated, and evidence-based treatment recommendations. The platform enhances care coordination and optimizes resource utilization.
- **Patients:** The system primarily supports patients seeking reliable guidance for complex medical conditions, including cancer and chronic diseases such as blood pressure management, diabetes, and hypertension. It empowers users with personalized insights, improves treatment awareness, and facilitates informed healthcare decisions.

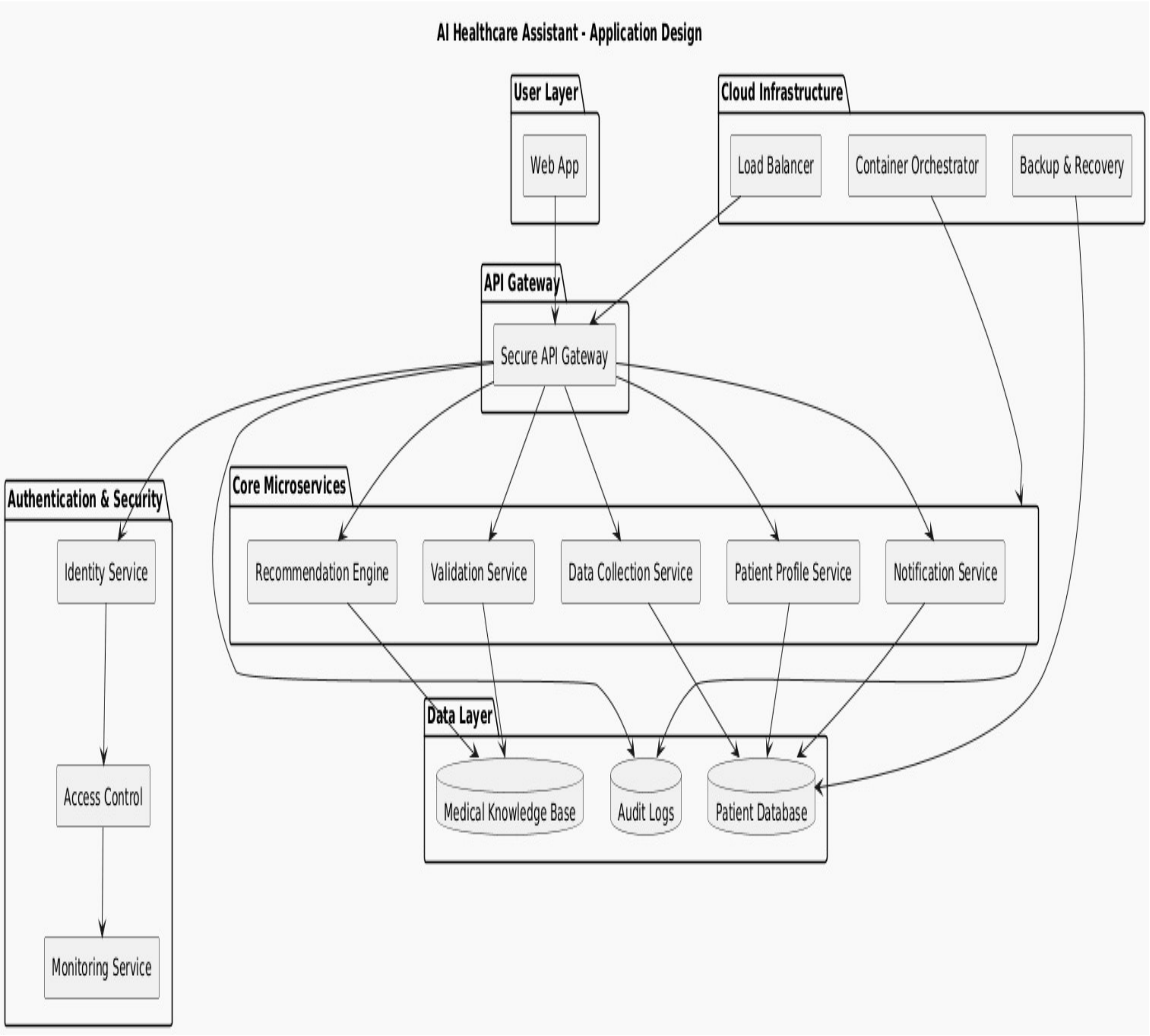
- **IT & Compliance Teams:** Technical and regulatory teams rely on the system for seamless integration with existing hospital infrastructure, data security management, and compliance with healthcare regulations and standards. They ensure the platform meets legal, ethical, and operational requirements.
- **AI/Software Developers:** Developers and system architects are responsible for designing, implementing, testing, and maintaining the AI models, backend infrastructure, and user interfaces. They continuously enhance system performance, scalability, and reliability to meet evolving healthcare demands.

1.3 System Overview-

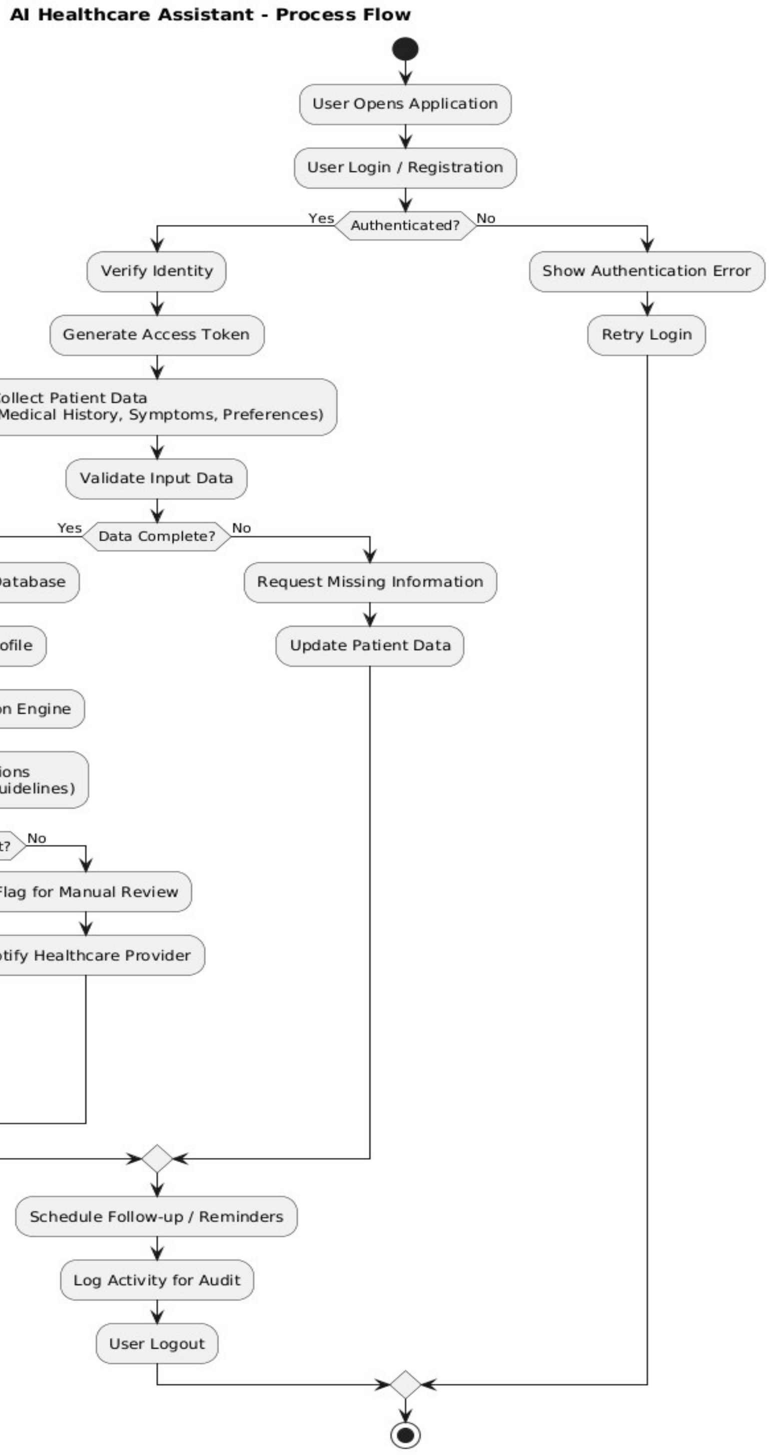
The AI assistant collects and validates patient data (medical history, disease stage, preferences, budget, location), interacts with users to clarify needs, and generates actionable, validated recommendations for treatments and hospital selection. The system is designed for integration with hospital systems and to support diverse patient populations, including those in rural or underserved areas. The advanced data analytics and medical knowledge models, the assistant generates personalized, evidence-based recommendations for treatment options and suitable healthcare facilities. These recommendations are validated against clinical guidelines, hospital capabilities, and patient-specific constraints to ensure safety, feasibility, and effectiveness.

System Design-

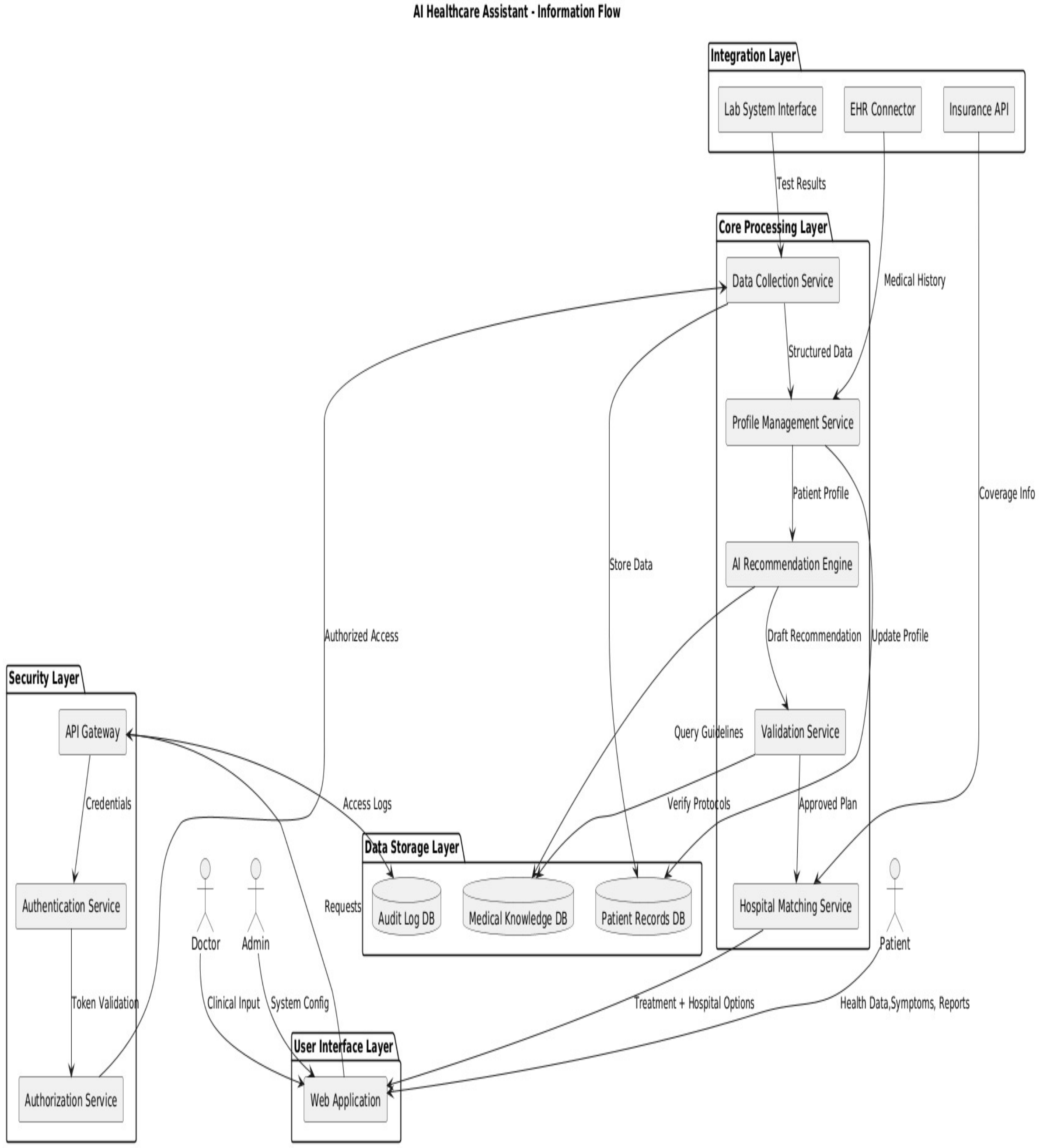
2.1 Application Design-



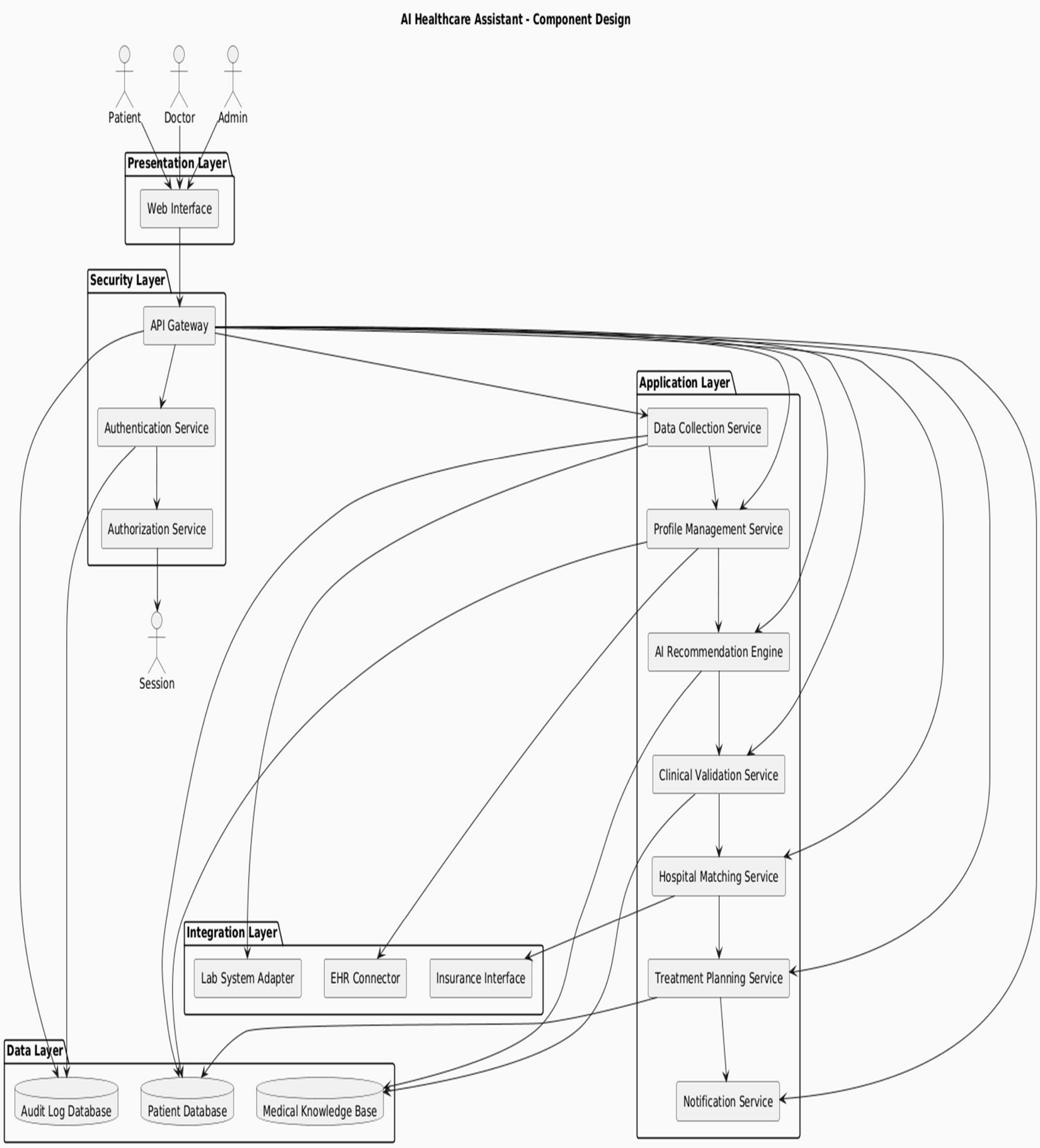
2.2. Process Flow-


AI Healthcare Assistant - Process Flow
User Opens Application
User Login / Registration
Authenticated?
Yes
No
Verify Identity
Show Authentication Error
Generate Access Token
Retry Login
Collect Patient Data
(Medical History, Symptoms, Preferences)
Validate Input Data
Data Complete?
Yes
No
Store Data in Secure Database
Request Missing Information
Analyze Patient Profile
Update Patient Data
Run AI Recommendation Engine
Validate Recommendations
(According to Clinical Guidelines)
Safe & Compliant?
Yes
No
Generate Treatment Plan
Flag for Manual Review
Match Hospital/Specialist
Notify Healthcare Provider
Prepare Personalized Report
Send Recommendations to User
Schedule Follow-up / Reminders
Log Activity for Audit
User Logout

2.3 Information Flow-

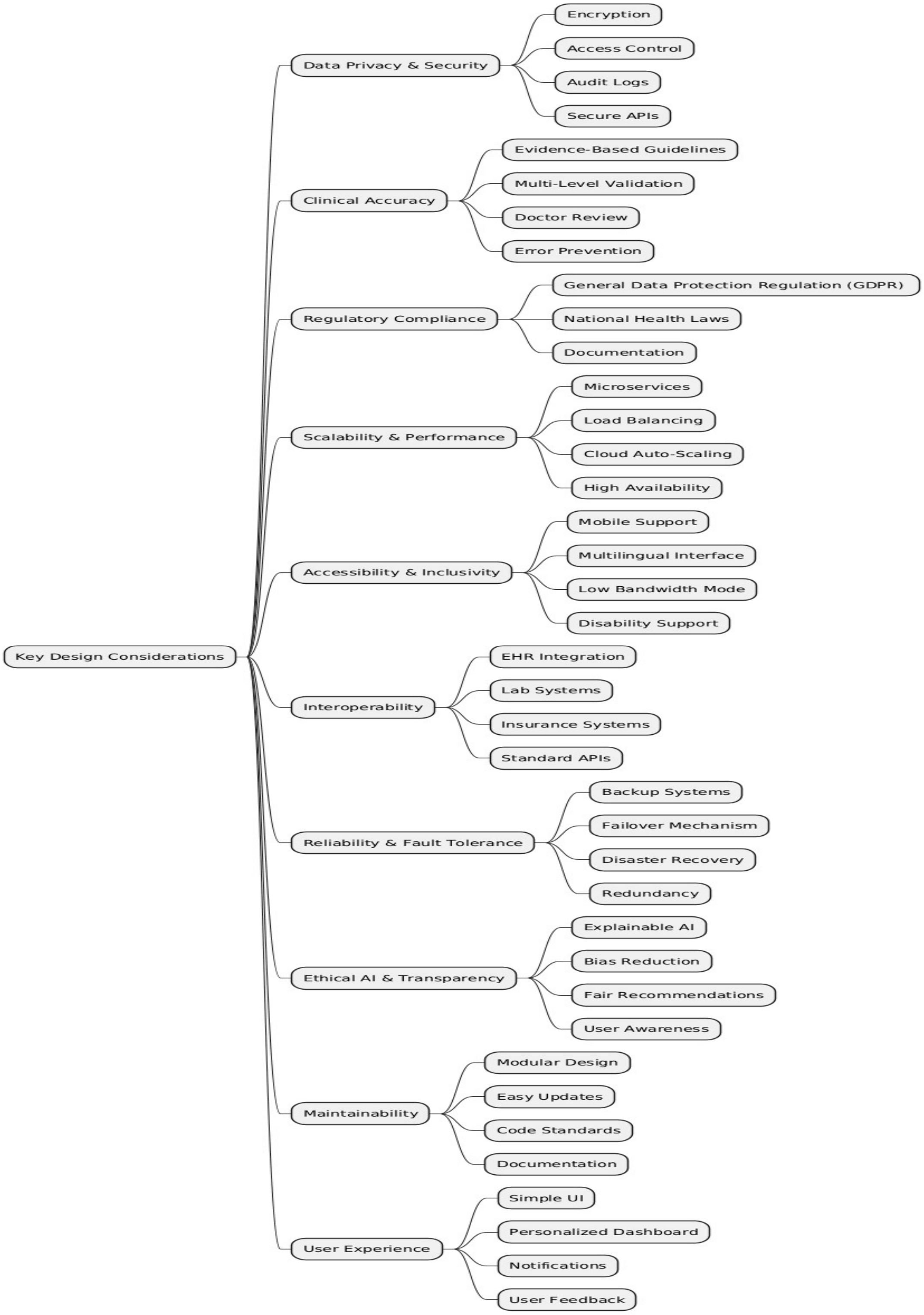


2.4 Components Design

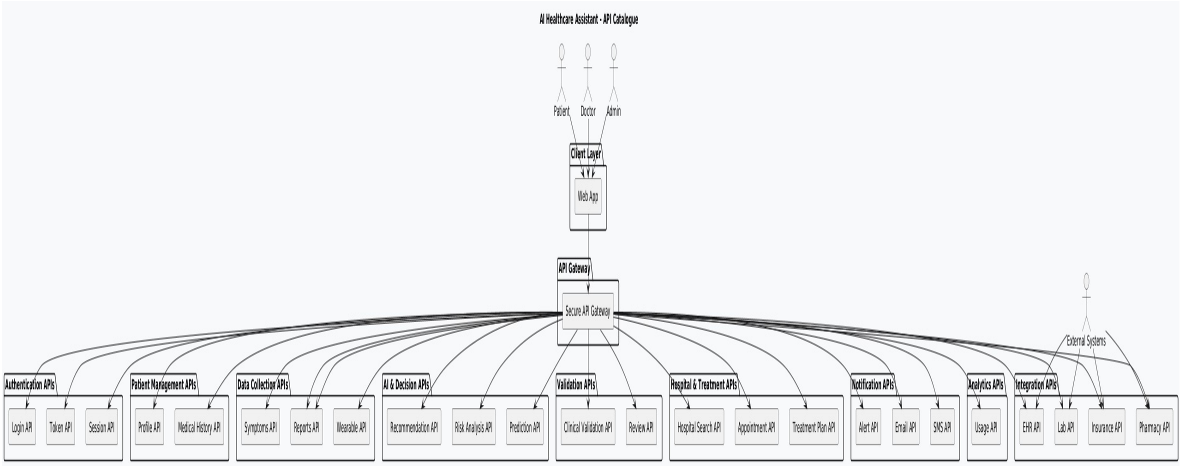


2.5 Key Design Considerations-

AI Healthcare Assistant - Key Design Considerations

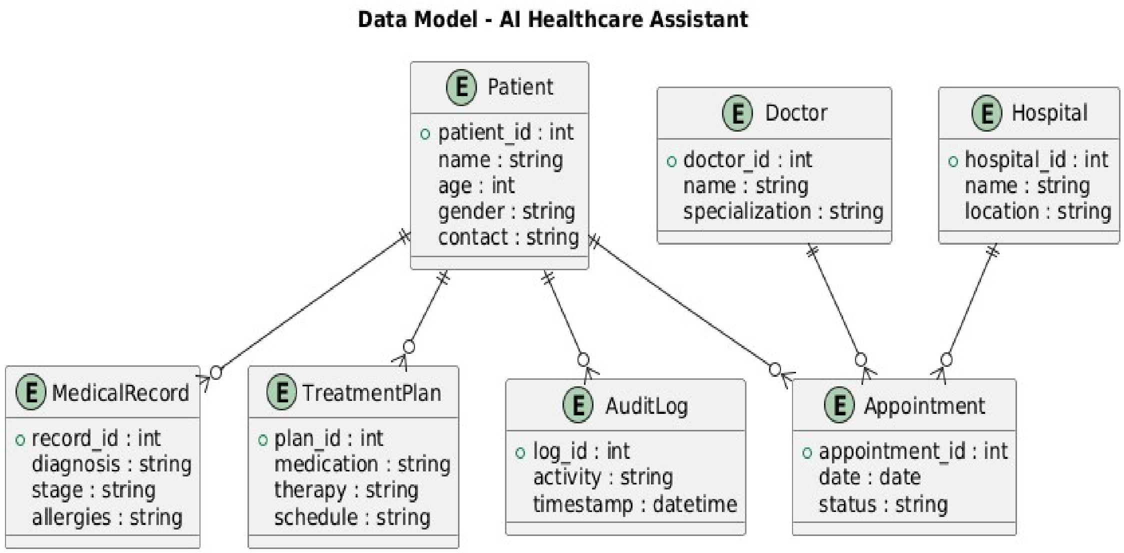


2.6 API Catalogue-

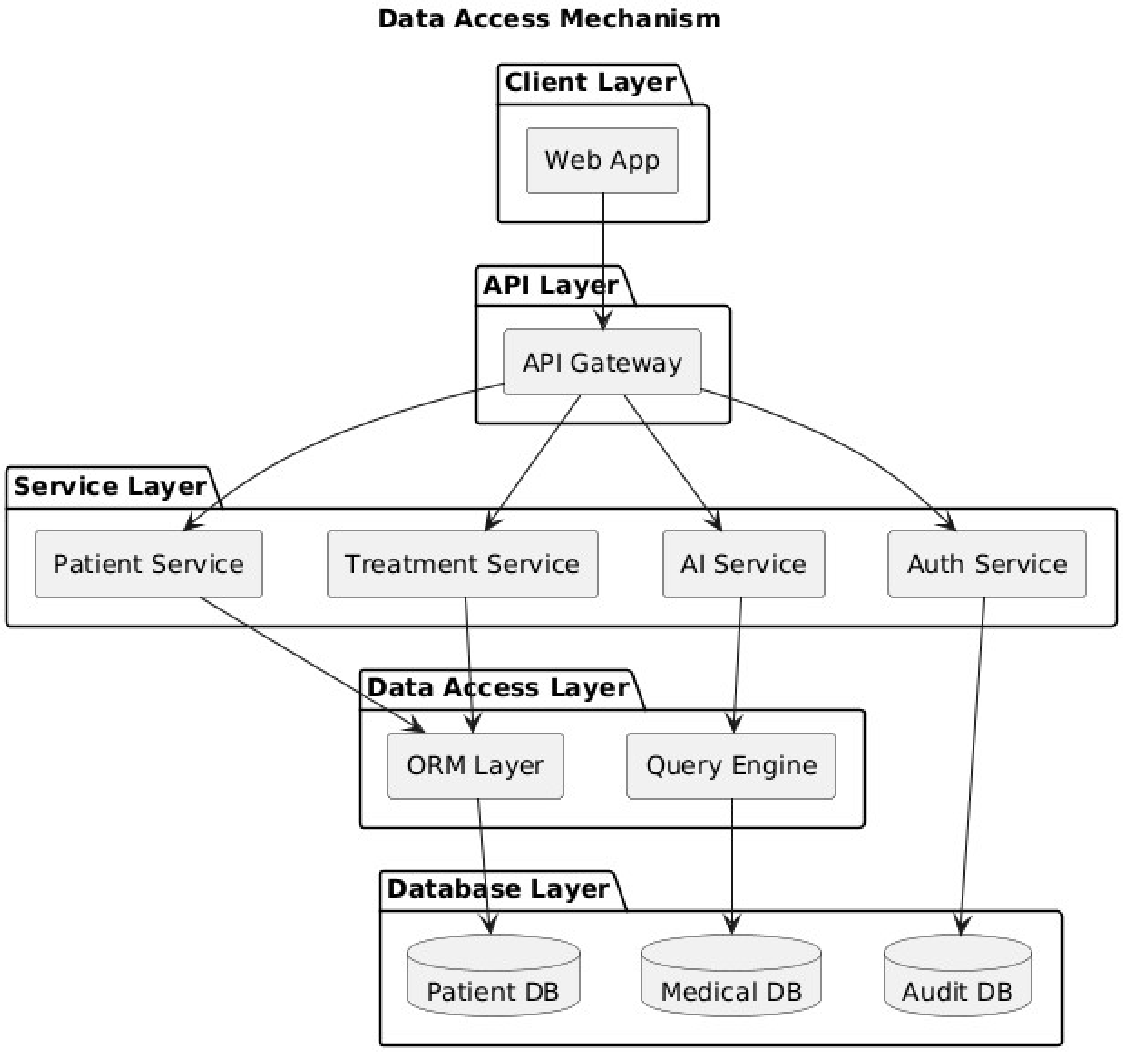


3 Data Design-

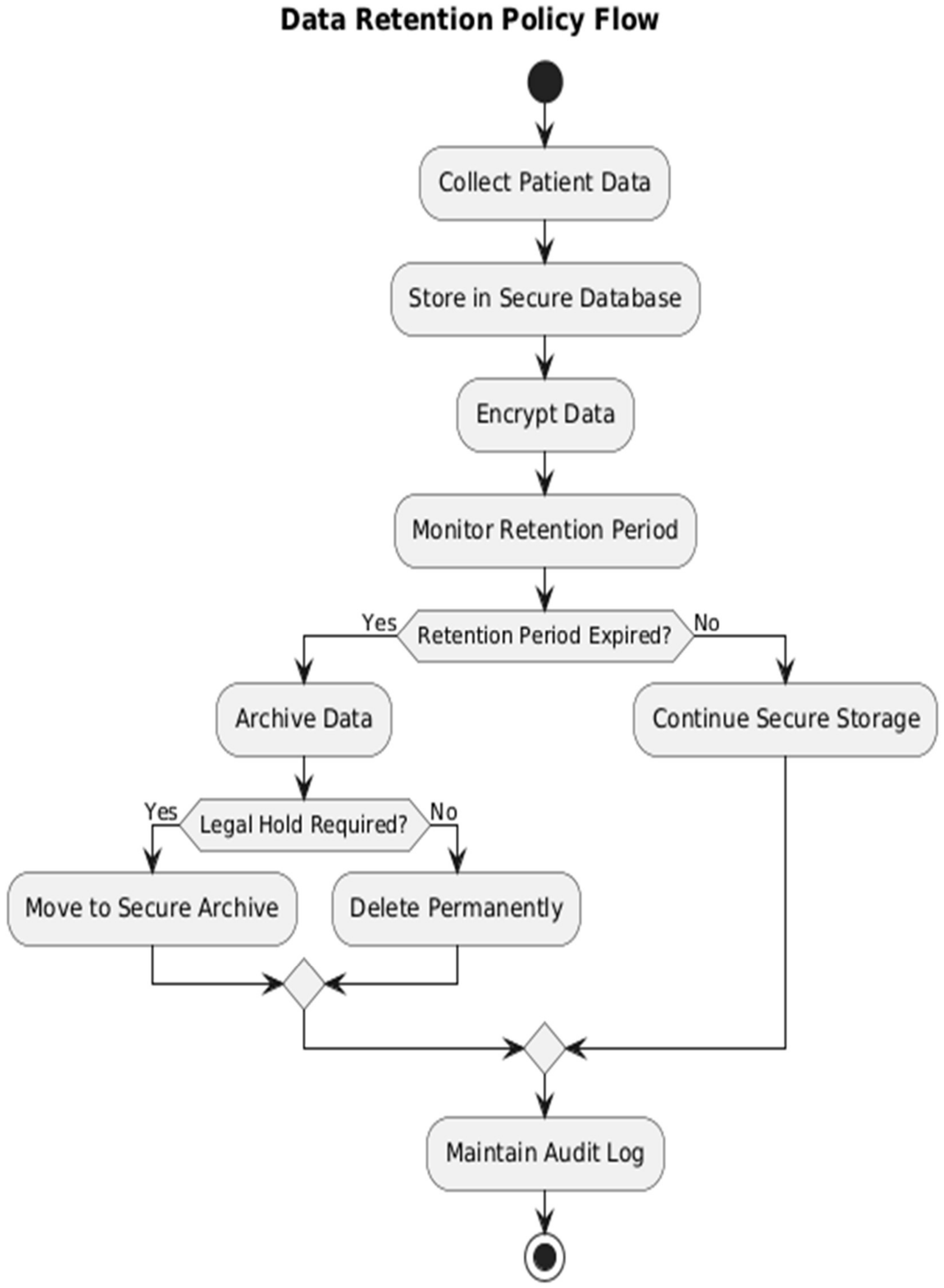
3.1 Data Model



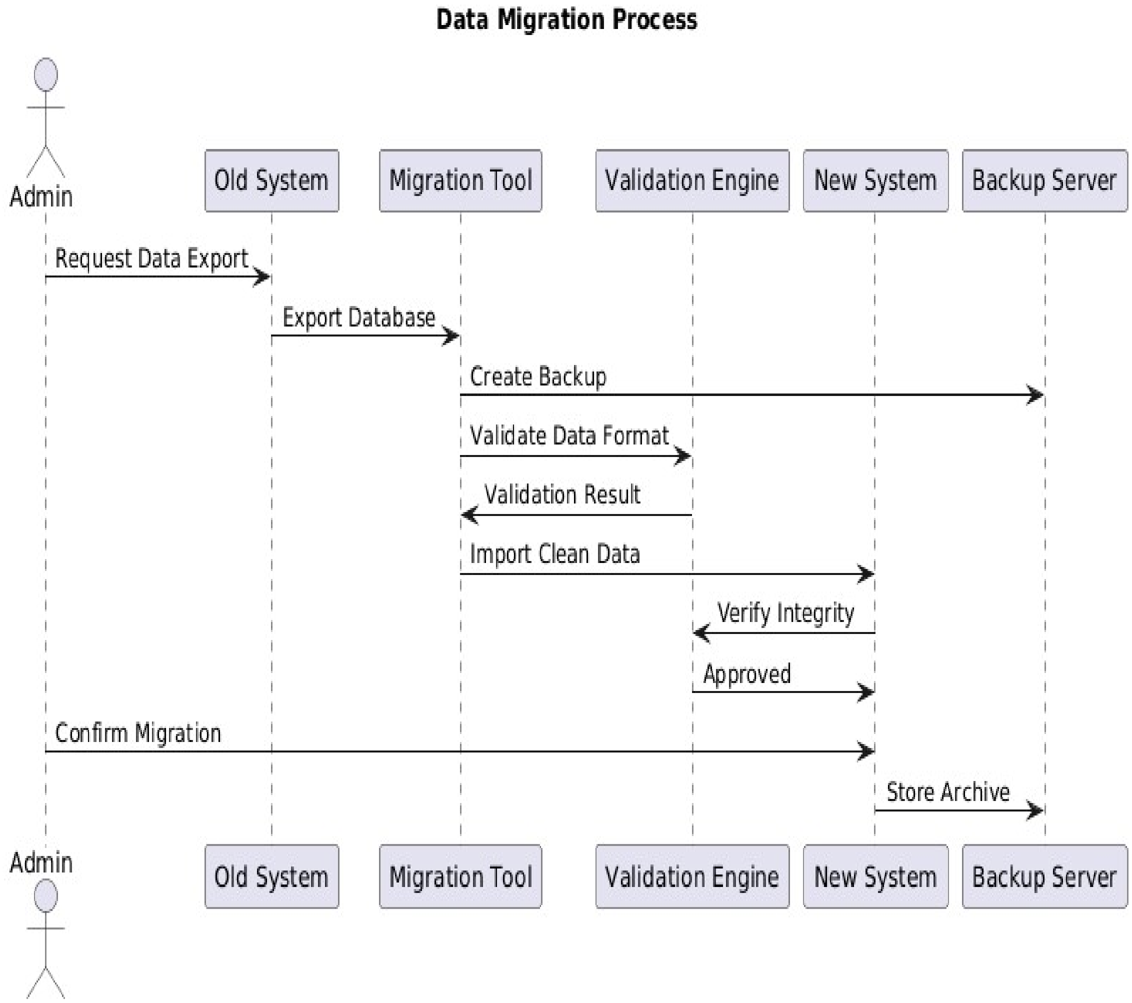
3.2 Data Access Mechanism



3.3 Data Retention Policies



3.4 Data Migration



4. Interfaces

The AI-driven healthcare assistant provides well-defined and secure interfaces that enable seamless interaction between users, system components, and external platforms. These interfaces ensure efficient data exchange, usability, and system interoperability.

4.1 User Interface (UI)

The User Interface serves as the primary interaction point between users and the system. It is designed to be intuitive, responsive, and accessible across multiple devices.

- Supports web and mobile platforms
- Provides secure login and registration
- Enables patients to submit medical data and view recommendations
- Offers administrators tools for system monitoring and management
- Includes multilingual support and accessibility features

4.2 System Interface

The System Interface manages communication between internal components and microservices within the application.

- Facilitates interaction between frontend and backend services
- Uses secure RESTful APIs for data exchange
- Ensures authentication and authorization for every request
- Supports service-to-service communication
- Enables load balancing and fault tolerance

4.3 API Interface

The API Interface defines how external and internal applications interact with system services.

- Provides RESTful endpoints for all major functionalities
- Supports JSON-based request and response formats
- Implements token-based authentication mechanisms
- Enforces role-based access control

5. State and Session Management

5.1 User Session Management

The system maintains secure user sessions to track authenticated interactions between users and system services.

- Each user session is initiated upon successful authentication.
- Secure session tokens are issued and stored in encrypted form.
- Sessions have predefined expiration times to prevent unauthorized access.
- Automatic logout is triggered after periods of inactivity.
- Multi-device session management is supported with controlled concurrency.

5.2 Application State Management

Application state refers to the current status of user interactions and system processes.

- Maintains temporary data such as form inputs, ongoing consultations, and draft records.
- Supports state synchronization across devices and sessions.
- Ensures continuity during network interruptions or system restarts.
- Separates transient session data from persistent patient records.

5.3 Distributed Session Handling

To support scalability and high availability, session management is implemented in a distributed manner.

- Session data is stored in centralized or distributed caches.
- Load balancers route requests without session dependency.
- Supports stateless service architecture.
- Enables seamless failover and fault tolerance.

5.4 Security in Session Management

Strong security measures are applied to protect session integrity and prevent misuse.

- Enforces secure cookie handling and HTTPS communication.
- Implements token rotation and renewal mechanisms.
- Detects and blocks session hijacking and replay attacks.
- Logs session activities for auditing and compliance.

5.5 Session Lifecycle Management

The system manages the complete lifecycle of user sessions.

1. Session Creation – Initiated after authentication
2. Session Validation – Verified on every request
3. Session Renewal – Extended upon active usage
4. Session Termination – Closed on logout or timeout
5. Session Cleanup – Automatically removed after expiration

5.6 Fault Tolerance and Recovery

Session continuity is maintained even during partial system failures.

- Backup session storage ensures data recovery.
- Automatic session revalidation after service restarts.
- Graceful degradation during network failures.
- Minimal disruption to active users.

6. Caching

Caching is implemented to enhance system performance, reduce response latency, and minimize repeated access to backend services and databases. The caching mechanism stores frequently accessed and computationally expensive data in high-speed memory layers.

- Patient session data, authentication tokens, and user preferences are temporarily cached to improve responsiveness.

- Medical guidelines, hospital directories, and treatment protocols are cached to reduce repeated database queries.
- AI inference results for similar cases are selectively cached to optimize processing time.
- Distributed in-memory caching systems are used to support scalability and high availability.
- Cache expiration and invalidation policies ensure data consistency and accuracy.
- Sensitive information is encrypted within cache storage to maintain security compliance.

7. Non-Functional Requirements

Non-functional requirements define the quality attributes and operational constraints of the healthcare assistant system. These requirements ensure that the system remains secure, reliable, efficient, and user-friendly.

7.1 Security Aspects

Security is a primary concern due to the sensitive nature of healthcare data. The system incorporates multi-layered security mechanisms to protect user information and system resources.

- End-to-end encryption for data in transit and at rest
- Strong authentication with multi-factor support
- Role-based and attribute-based access control
- Regular security audits and vulnerability assessments
- Intrusion detection and real-time threat monitoring
- Secure key management and certificate handling
- Compliance with healthcare data protection regulations
- Audit trails for all critical operations

7.2 Performance Aspects

The system is designed to deliver fast, consistent, and reliable performance under varying workloads.

- Average response time below defined thresholds
- Support for concurrent users without degradation
- Auto-scaling based on workload demand
- Load balancing across microservices
- Optimized database indexing and query execution
- Asynchronous processing for long-running tasks
- Efficient memory and resource utilization
- Continuous performance monitoring and tuning

These performance requirements ensure smooth system operation and positive user experience.

8. References

The following references provide foundational knowledge and technical guidance for the design and implementation of the AI-driven healthcare assistant system:

1. World Health Organization (WHO). Digital Health Guidelines.
2. National Institute of Standards and Technology (NIST). Cybersecurity Framework.
3. Health Level Seven International (HL7). FHIR Standards Documentation.
4. Cloud Security Alliance (CSA). Cloud Security Best Practices.
5. Research articles on AI-based Clinical Decision Support Systems.
6. Official documentation of cloud service providers and security frameworks.