**System Architecture Document**

**Project Title:** HealthHub

**Document Title:** System Architecture Document

**Version:** 1.0

**Date:** 08.08.2024

**1. Introduction**

**1.1 Purpose:** The purpose of this document is to describe the overall architecture of the HealthHub system, including its components, technology stack, data flow, and security measures.

**1.2 Scope:** This document covers the high-level system architecture, component architecture, technology stack, data flow, and security measures for HealthHub.

**1.3 Definitions, Acronyms, and Abbreviations:**

* SRS: Software Requirements Specification
* JWT: JSON Web Token
* RBAC: Role-Based Access Control
* HIPAA: Health Insurance Portability and Accountability Act

**1.4 References:**

* NestJS Documentation: https://docs.nestjs.com/
* Prisma Documentation: https://www.prisma.io/docs/
* React Documentation: https://reactjs.org/docs/getting-started.html
* React Native Documentation: https://reactnative.dev/docs/getting-started
* HIPAA Compliance: https://www.hhs.gov/hipaa/for-professionals/security/laws-regulations/index.html

**2. Overall System Architecture**

**2.1 High-Level System Architecture Diagram:**

* The diagram will visually represent the interaction between different components of the HealthHub system, including web and mobile clients, backend services, and third-party integrations.

**2.2 Description of the Architecture:** HealthHub is designed with a modular architecture to ensure scalability, maintainability, and security. The system comprises a web frontend, a mobile frontend, a backend API server, and a PostgreSQL database. The backend API server handles business logic, authentication, and communication with third-party services for notifications and video conferencing. The frontends interact with the backend API to provide a seamless user experience across devices.

**3. Component Architecture**

**3.1 Backend Architecture:** The backend is built using NestJS, a progressive Node.js framework. It uses PostgreSQL as the database and Prisma ORM for database management. The backend handles user authentication using JWT, manages patient records, schedules appointments, and facilitates telemedicine consultations. It also integrates with third-party services like WATI for WhatsApp notifications and Twilio for SMS notifications.

**3.2 Frontend Architecture:**

* **Web Frontend:** The web application is built using React and Next.js to provide a responsive and interactive user interface. It interacts with the backend API for data operations and real-time updates.
* **Mobile Frontend:** The mobile application is developed using React Native, ensuring a consistent user experience across iOS and Android devices. It communicates with the backend API for data synchronization and notifications.

**3.3 Database Design:** The database schema is designed using PostgreSQL, with Prisma ORM managing the database interactions. Key tables include Users, Patients, Doctors, Appointments, and MedicalRecords. Relationships between tables ensure data integrity and efficient querying.

**3.4 Integration Points:**

* **WATI:** Used for sending WhatsApp notifications to users.
* **Twilio:** Used for sending SMS notifications to users.
* **WebRTC:** Used for enabling video consultations between patients and doctors.

**4. Technology Stack**

**4.1 Backend Technologies:**

* **Framework:** NestJS
* **Database:** PostgreSQL
* **ORM:** Prisma
* **Authentication:** JWT

**4.2 Frontend Technologies:**

* **Web:** React, Next.js
* **Mobile:** React Native

**4.3 Database Technologies:**

* **Database:** PostgreSQL

**4.4 Other Tools and Services:**

* **Notifications:** WATI, Twilio
* **Video Conferencing:** WebRTC
* **Hosting and Deployment:** AWS

**5. Data Flow**

**5.1 Data Flow Diagram:**

* The diagram will illustrate how data flows between the web/mobile clients, backend services, and third-party integrations.

**5.2 Description of Data Flow:**

1. **User Interaction:** Users interact with the web or mobile frontend.
2. **API Requests:** Frontends send API requests to the backend server.
3. **Database Operations:** Backend processes requests, performs necessary database operations using Prisma, and retrieves or stores data in PostgreSQL.
4. **Notifications:** For actions requiring notifications, the backend communicates with WATI and Twilio to send messages to users.
5. **Telemedicine:** For video consultations, the backend establishes a WebRTC connection between patients and doctors.
6. **API Responses:** Backend sends responses back to the frontend, which updates the user interface accordingly.

**6. Security**

**6.1 Security Measures:**

* **Data Transmission:** All data transmitted between clients and the server will be encrypted using SSL/TLS.
* **Data Storage:** Sensitive data will be encrypted in the database.
* **Authentication:** JWT will be used for secure authentication and session management.
* **Authorization:** Role-Based Access Control (RBAC) will be implemented to ensure users can only access authorized resources.

**6.2 Compliance Requirements:**

* The system will comply with HIPAA regulations to ensure the privacy and security of patient health information.