Technical Design Document

Medical History Web App

Author: Ursu Vlad

Introduction

The Medical History (SanatOS) Web App is a web-based application designed to provide patients and healthcare providers with a secure and efficient platform for managing and accessing medical records. This document outlines the technical architecture, key components, and data flow of the application.

Architecture

Application Architecture

The app follows a three-tier architecture:

1. **Presentation Layer**: The user interface is built using HTML, CSS, and JavaScript, with a responsive design for mobile and desktop users. The frontend is developed using React.js, providing an interactive and intuitive user experience.
2. **Application Layer**: This layer comprises the core application logic. It is implemented in Java using the Spring Boot framework. Spring Security is used for user authentication and authorization.
3. **Data Layer**: Data is stored in a relational database using MySQL. Spring Data JPA is used to interact with the database.

Security

- User authentication is handled through Spring Security, which supports both username-password and two-factor authentication using Google Authenticator.
- Data transmission is secured with HTTPS to protect sensitive medical records.
- Patient and healthcare provider roles have different access levels, ensuring that sensitive data is protected.
- Encrypted data storage.

Components

User Management

- User registration and login.
- User profile management.
- 2FA setup and management using Google Authenticator.
- Role-based access control.

Patient Management

- Secure login for patients.
- Create and manage patient profiles.
- Record and update patient medical history.
- Upload and store medical documents.
- View patient history and records.

Healthcare Provider Dashboard

- Secure login for healthcare providers.
- Access to patients' medical records.
- View and update patient information.
- Messaging system for communication with patients.

Data Flow

1. User Registration:
    - Users register with a username and password.
    - They can optionally set up 2FA using Google Authenticator.
2. User Authentication:
    - Users log in with their credentials.
    - 2FA is enforced if configured.
3. Patient Management:
    - Patients create profiles and provide personal and medical information.
    - They can upload medical documents.
4. Healthcare Provider Dashboard:
    - Healthcare providers log in to access patient records.
    - They can view, update, and communicate with patients.
5. Data Storage:
    - User profiles, patient information, and medical records are securely stored in the MySQL database.

Conclusion

The Medical History App's technical design ensures robust security, user-friendly functionality, and efficient data management. It offers a responsive and intuitive user interface while protecting sensitive medical information. The three-tier architecture provides scalability, maintainability, and security for the app.