UNIVERSITE ABDELMALEK ESSAADI

Master IA et science de données

Module : Blockchain

Projet de Fin de Module :

DOSSIER MEDICAL AVEC BLOCKCHAIN

Réalisé PAR :

- Aymane MAHRI
- Adnane RIYADI
- Yasmina HACHHOUCH

Encadré PAR :

Prof. Ikram BEN ADBELOUAHAB

Année Universitaire : 2024/2025

Table des matières :

Executive Summary :

The Medical Record Management with Blockchain project addresses critical challenges in healthcare data management, such as ensuring security, privacy, and transparency in handling sensitive medical information. Traditional systems are often centralized and prone to vulnerabilities like data breaches, unauthorized access, and errors, which can compromise patient trust and safety. This initiative leverages the decentralized and immutable nature of blockchain technology to propose an innovative solution for securely managing medical records.

The core objective of this project is to develop a decentralized application (DApp) that utilizes the Ethereum blockchain to securely store and manage medical records. By implementing smart contracts written in Solidity, the system ensures data integrity, confidentiality, and traceability. A distributed storage system (IPFS) is integrated to handle large medical files efficiently, with only file hashes stored on the blockchain to guarantee their authenticity. For The project's frontend, developed in Python, it provides an intuitive interface for healthcare professionals and patients to interact with the blockchain seamlessly.

This approach transforms the healthcare sector by decentralizing data control, empowering patients to manage permissions, and creating an auditable history of record access. By combining blockchain's transparency with robust encryption techniques, the project prioritizes both accessibility and privacy. The result is a scalable, secure, and interoperable solution that aligns with the growing demand for digital transformation in medical data management.

Introduction

Medical data management is a critical aspect of modern healthcare, where the confidentiality, security, and accessibility of patient information are paramount. However, traditional systems often fall short, plagued by vulnerabilities such as data breaches, unauthorized access, and inefficiencies in sharing information across stakeholders. These shortcomings highlight the urgent need for innovative solutions to address the growing demands of the healthcare sector.

This project proposes an approach to tackle these challenges by harnessing the power of blockchain technology. Blockchain's decentralized and immutable architecture ensures data integrity while providing a transparent and secure framework for managing sensitive medical information. Unlike traditional systems, this solution eliminates single points of failure and reduces reliance on centralized control, making it more robust against cyber threats.

This project aims to develop a decentralized application (DApp) that enables secure and efficient handling of medical records. By combining blockchain with distributed file storage systems like IPFS and integrating smart contracts for access control and audit trails, we seek to offer a comprehensive solution that balances privacy with transparency. Furthermore, by employing user-friendly front-end technologies in custom tkinter Python, we ensure a seamless interaction experience for both patients and healthcare providers.

Through this initiative, we aspires to redefine how medical records are managed, fostering trust, security, and efficiency in an increasingly digital healthcare landscape.

Technology Stack :

1. Blockchain and Smart Contracts :

a. Ethereum:

Ethereum is a decentralized blockchain platform that establishes a peer-to-peer network that securely executes and verifies application code, called smart contracts. Smart contracts allow participants to transact with each other without a trusted central authority. Transaction records are immutable, verifiable, and securely distributed across the network, giving participants full ownership and visibility into transaction data. Transactions are sent from and received by user-created Ethereum accounts. A sender must sign transactions and spend Ether, Ethereum's native cryptocurrency, as a cost of processing transactions on the network.

b. Solidity:

Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs that govern the behavior of accounts within the Ethereum state.

Solidity is a curly-bracket language designed to target the Ethereum Virtual Machine (EVM). It is influenced by C++, Python, and JavaScript. Solidity is statically typed, supports inheritance, libraries, and complex user-defined types, among other features. With Solidity, we can create contracts for uses such as voting, crowdfunding, blind auctions, and multi-signature wallets.

2. Data Storage :

a. IPFS :

The InterPlanetary File System (IPFS) is a protocol, hypermedia and file sharing peer-to-peer network for storing and sharing data in a distributed hash table. By using content addressing, IPFS uniquely identifies each file in a global namespace that connects IPFS hosts, creating a resilient system of file storage and sharing

3. Frontend Development :

Modern and customizable Python UI-library based on Tkinter. It provides additional UI elements compared to Tkinter and allows for extensive customization. CustomTkinter is designed to be easy to use and integrates seamlessly with Tkinter.

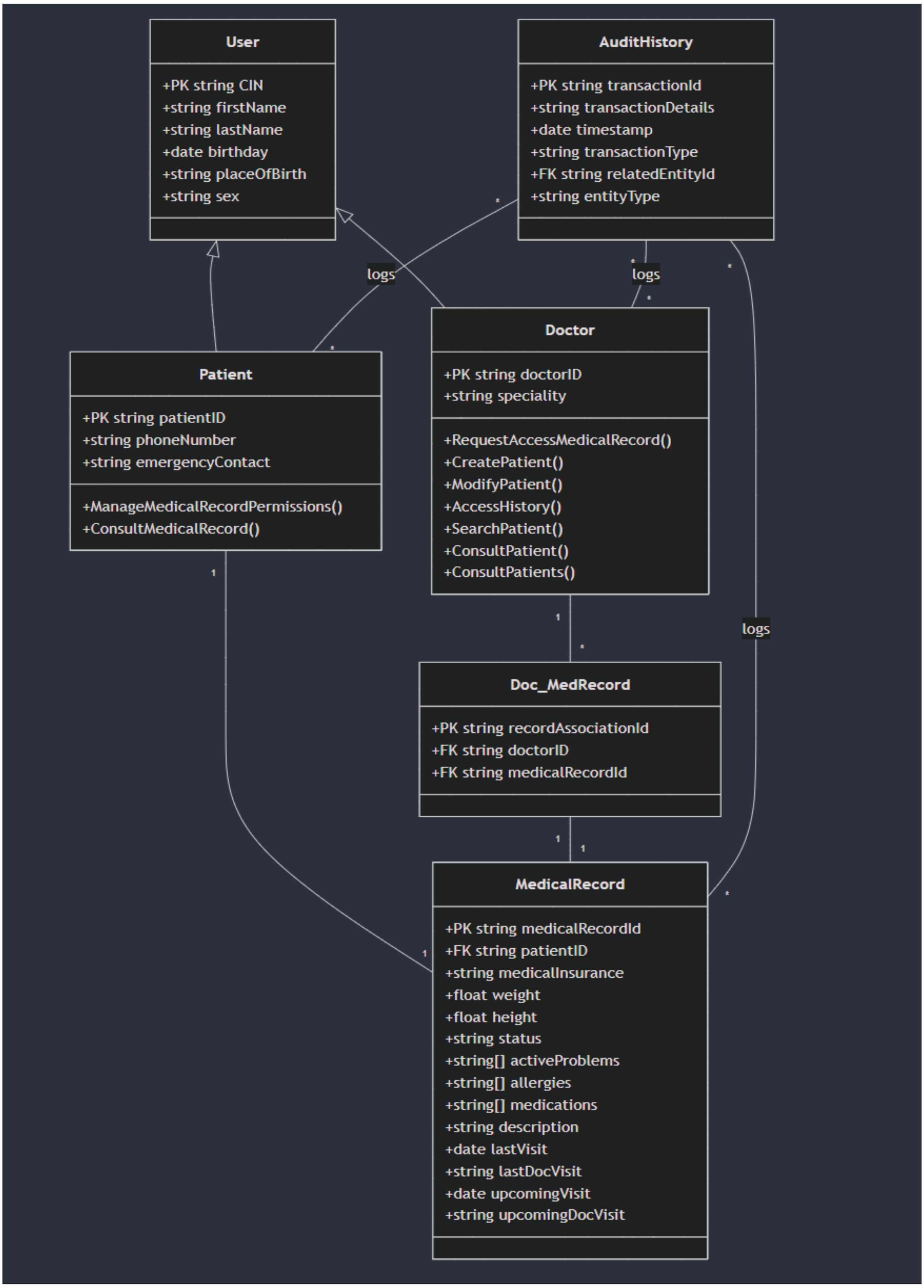
Design and Planning :

1. Technical Requirements:

The technical foundation of the system includes Ethereum blockchain, IPFS for distributed storage, and smart contract-based access control. Key functional areas such as patient registration, medical record management, access permissions, and auditing have been carefully planned to align with these requirements. The system also includes decentralized authentication using Ethereum wallets and an intuitive user interface for seamless interaction.

2. Class Diagram:

The class diagram represents the static structure of the backend, outlining the system's key entities and their relationships.



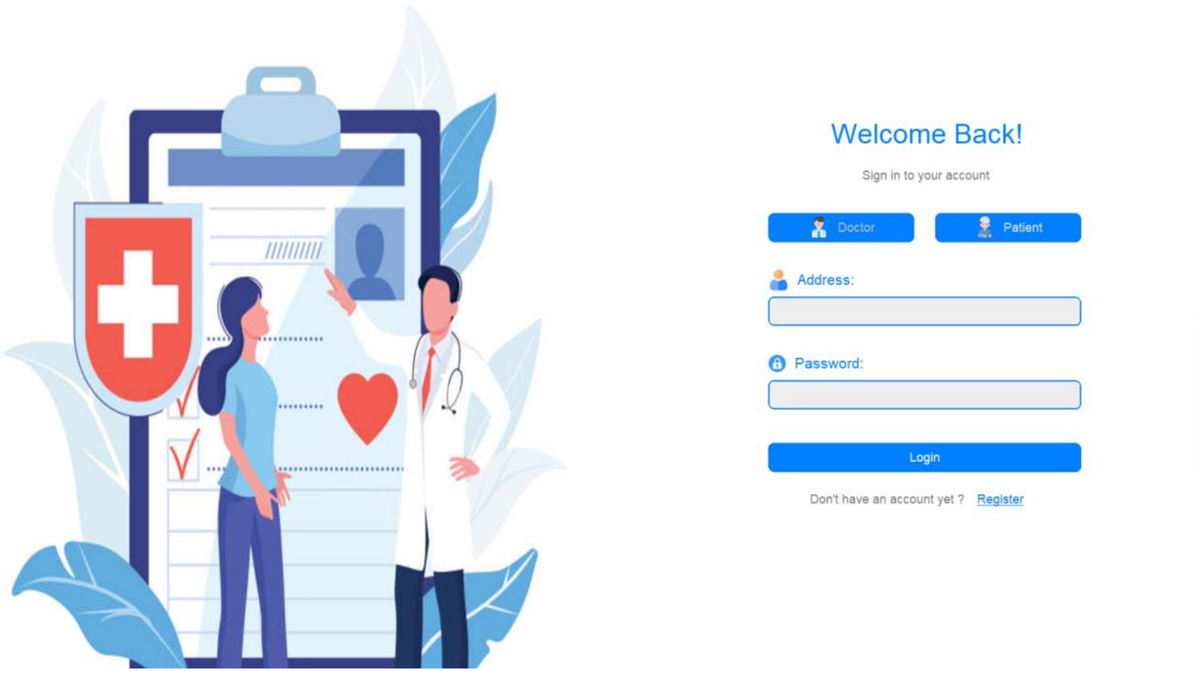
System Architecture :

3. Frontend :

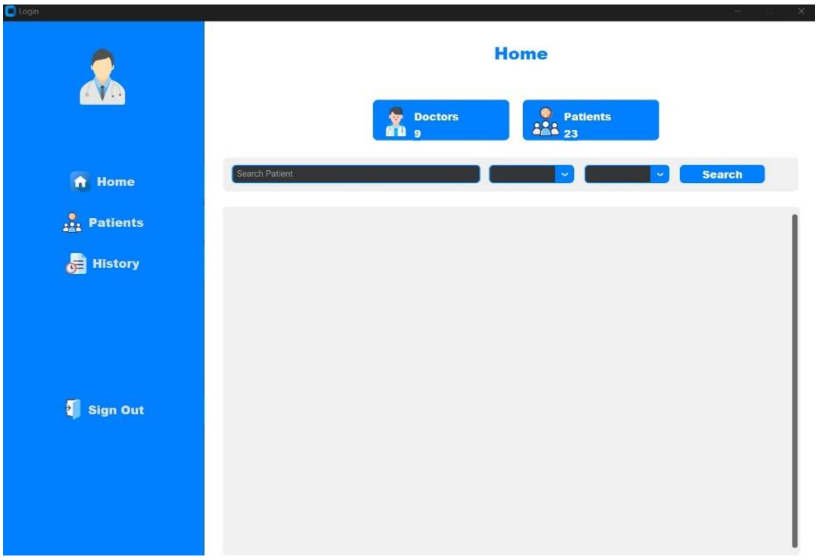
The frontend of the application serves as the primary interface for patients and medical professionals, designed to provide an intuitive and seamless experience for interacting with the blockchain-based medical record system. Developed using Python with CustomTkinter, the interface is built to be responsive, efficient, and user-friendly, catering to the unique needs of both user groups.

Doctor Interface

1. **Login and Registration**:
   Doctors access the system via a secure login page or, if they are new users, register through a dedicated registration page. Authentication is handled through Ethereum wallets or cryptographic keys, ensuring a decentralized and secure process.

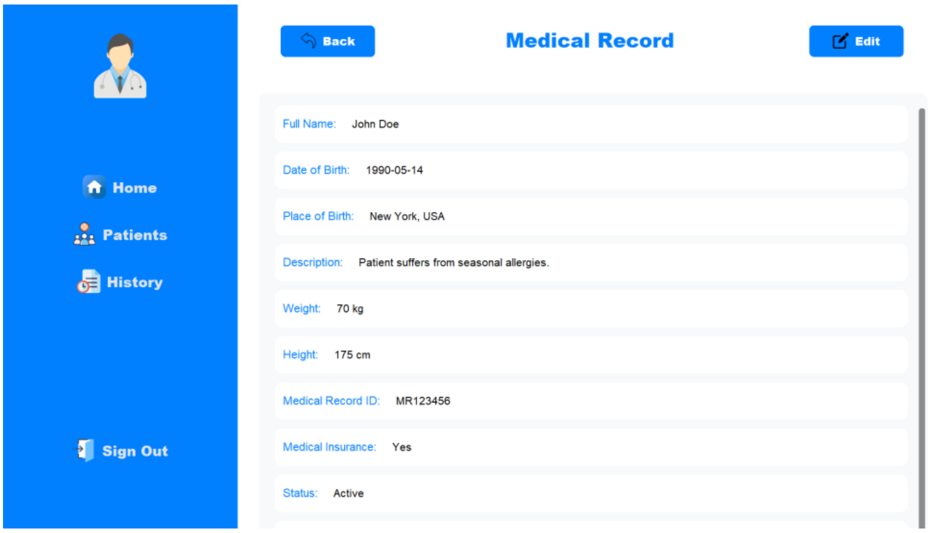


2. **Dashboard**:
   Once logged in, the doctor's dashboard provides the following key functionalities:
   - **Home**: A central hub that provides quick access to users through the search functionality

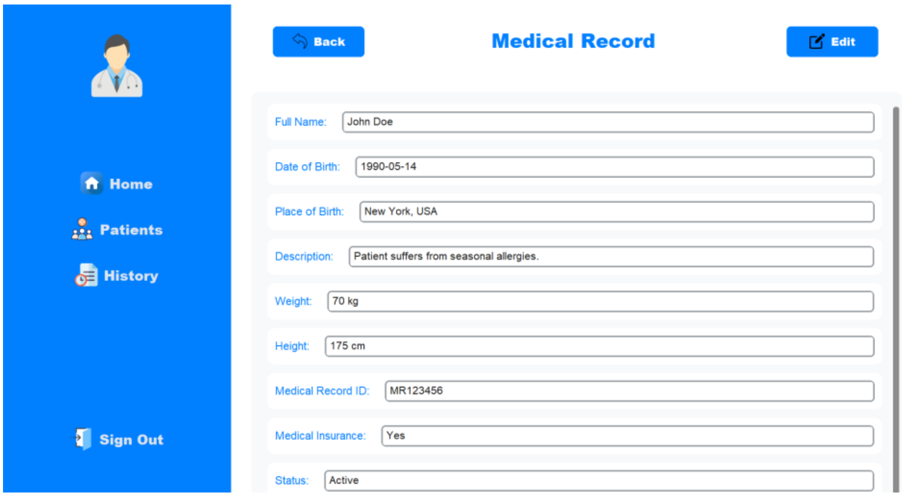


- **Patients**: A comprehensive list of registered patients, where doctors can view list of patients:

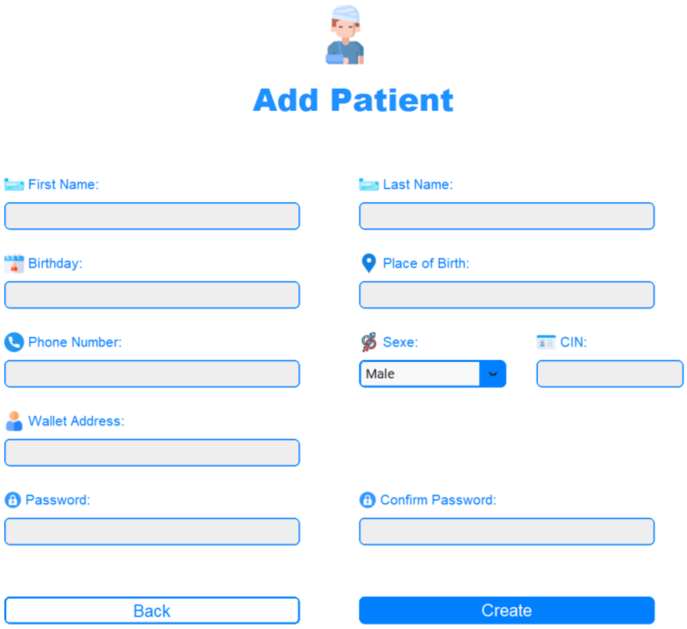
  - **Patient Page**: Displays a patient's medical history and associated records.



- **Edit Page**: Enables updates to a patient's medical record, with all changes securely stored and tracked on the blockchain.



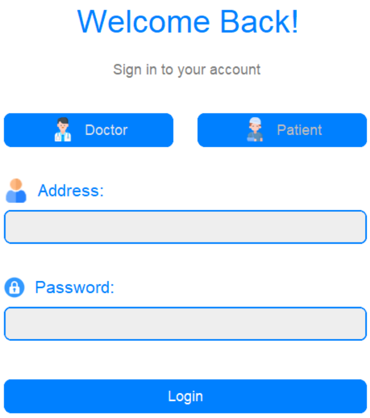
- **Add Patient**: register a new user to the blockchain and add its own medical record.



- **History**: Displays an immutable log of all changes made to medical records, providing full transparency and accountability.

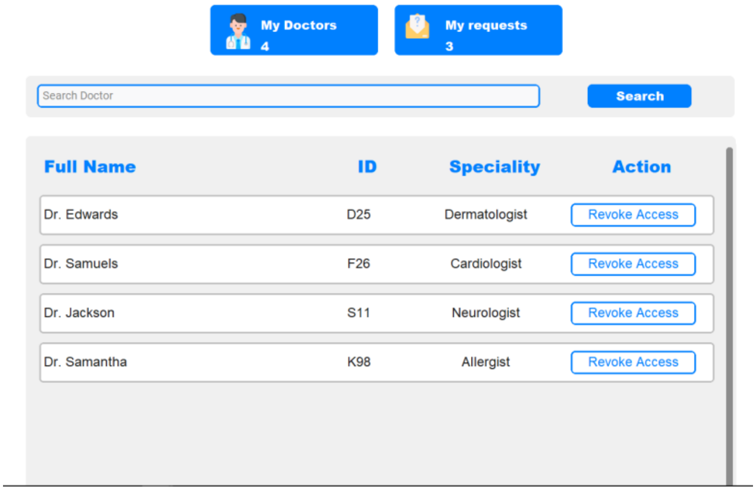
Patient Interface

1. **Login and Access**:
   Patients log in securely using their cryptographic credentials, ensuring only authorized individuals can access sensitive medical information.



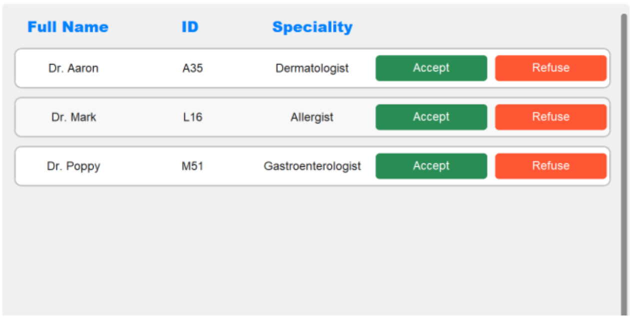
2. **Dashboard**:
The patient dashboard is designed with simplicity and ease of use in mind, offering the following features:

    - **Home**: Provides a personalized overview of recent activities and requests.



    - **Medical Records**: Displays the patient's medical history, stored securely in IPFS and accessible only with appropriate permissions.

- **Requests**: Allows patients to manage data access requests from medical professionals, enabling or revoking permissions in real time.



- **History**: Provides a detailed log of all interactions with the patient's medical data, including access events and updates, ensuring full transparency.

This interface design ensures that both doctors and patients can interact with the system in a secure, efficient, and user-centric manner.

4. Backend:

The backend implements sophisticated access control mechanisms and maintains a transparent record of all interactions within the system.

Smart Contracts for Data Management

The backend relies on Solidity-based smart contracts to manage key operations such as patient registration, doctor permissions, and audit logging. The `PatientContract` handles patient information, allowing healthcare providers to register patients and manage their medical records securely. Each patient's data is linked to a unique blockchain address, ensuring identification without centralized storage.

The `DoctorContract` focuses on managing doctors' registrations and permissions. It allows doctors to request access to patient records and, with appropriate permissions, create or update those records. Access control is enforced rigorously, ensuring that no unauthorized individual can view or modify sensitive data.

Transparency and accountability are core features of the system, achieved through the `AuditContract`. This contract logs all actions performed on medical records, including patient registrations, record modifications, and access approvals. Each entry in the audit log is immutable and timestamped, providing a trustworthy record of system activity.

Integration with IPFS

To handle large files such as medical documents, the system integrates with the InterPlanetary File System (IPFS). Files are encrypted and uploaded to IPFS, with only their content hashes stored on the blockchain. This approach reduces storage costs while maintaining the immutability and integrity of the data.

Access to files stored on IPFS is controlled through smart contracts. Patients grant or revoke permissions for specific doctors to view their files, ensuring that privacy is maintained. When a doctor retrieves a file, the backend verifies their permissions and provides the necessary decryption key, ensuring that only authorized users can access sensitive information.

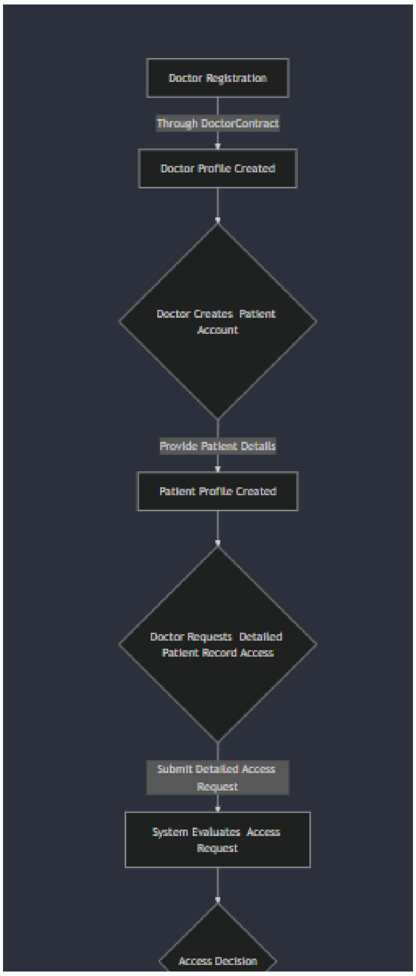
Medical Record Management

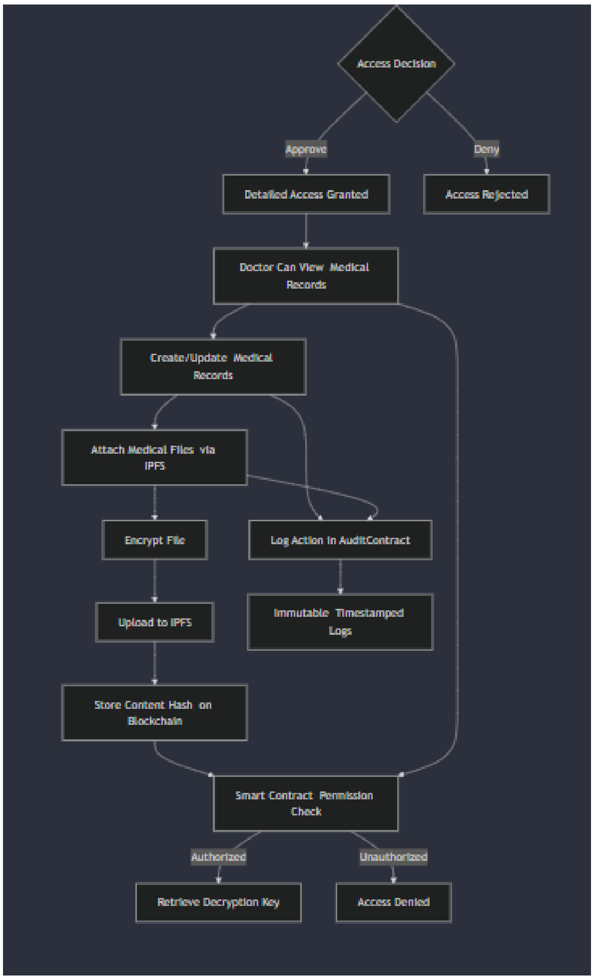
Using the `DoctorContract`, doctors can create detailed medical records that include information about a patient's medical history, medications, allergies, and upcoming appointments. These records are linked to unique identifiers and stored securely, with changes logged in the audit system. Additionally, files such as medical reports can be attached to records via IPFS, ensuring that all relevant data is accessible in one place.

Patients play an active role in managing their medical information. Through the `PatientContract`, they can review access requests from doctors and approve or deny them. This decentralized approach empowers patients with full control over their data and ensures compliance with privacy requirements.

Workflow of the Backend

The workflow begins with the registration of patients and doctors through the `PatientContract` and `DoctorContract`. Doctors can request access to specific patient records, which patients can approve or deny. Once access is granted, doctors can view, update, or attach files to medical records, with all changes logged in the audit system. Sensitive files are stored in IPFS, and only authorized users can retrieve them through permissions enforced by smart contracts.





Conclusion:

This project demonstrates a transformative approach to addressing the persistent challenges in healthcare data management. By leveraging blockchain technology's decentralized and immutable nature, combined with distributed storage via IPFS, the system ensures security, privacy, and transparency in handling sensitive medical information.

The project achieves a significant step toward empowering patients with control over their medical data while providing healthcare professionals with secure and efficient tools for managing records. The implementation of smart contracts for patient registration, access control, and audit logging offers a robust framework that eliminates reliance on centralized systems and reduces vulnerabilities. Additionally, the seamless integration of an intuitive user interface with backend technologies ensures usability without compromising security.

This system not only addresses current issues in medical data management but also paves the way for scalable, interoperable solutions in the future. With further refinements, such as expanding support for additional blockchain platforms or optimizing the system for larger-scale deployments, this project can serve as a foundational model for decentralized healthcare systems globally.

In conclusion, the project successfully fulfills its objectives of creating a secure, transparent, and decentralized solution for medical record management, demonstrating the potential of blockchain to revolutionize the healthcare industry.