A BlockChain Project: Decentralized Prescription Validator

Nikil Shyamsundar, Hrishabh Ayush, Pradhi Pakkerakari

Vision

Our project is about revolutionizing the healthcare system by leveraging blockchain technology to redefine prescription validation, empowering patients like never before. Traditionally, the power hierarchy among patients, pharmacists, and physicians leaves little room for patient autonomy, creating bottlenecks and inefficiencies. Our solution flips this paradigm by introducing a decentralized, voter-based system where pharmacists collectively validate prescriptions, ensuring transparency and fairness.

With our system, patients are no longer tied to specific pharmacies or dependent on individual pharmacists. Instead, prescriptions can be approved in mere seconds through a secure, blockchain-enabled "black box" process. This eliminates unnecessary delays and opens the door to seamless online ordering from any pharmacy, fueling the growth of e-commerce in the pharmaceutical industry.

Imagine a future where patients can access their medications from a global network of pharmacies, free from bureaucratic roadblocks, while maintaining safety and trust. Our blockchain application doesn’t just change the process—it redefines the very fabric of patient empowerment in healthcare.

At the heart of our solution is a simple yet powerful blockchain structure that stores hashed prescriptions. Once recorded on the blockchain, each prescription becomes immutable, ensuring its integrity and preventing tampering. The consensus mechanism we employ—Proof of Authority (PoA)—ensures that only authorized doctors and pharmacies can add or validate blocks on the network. Pharmacists can then use Zero-Knowledge Proofs (ZKPs) to verify these prescriptions without exposing sensitive patient information. This not only secures the authenticity of prescriptions but also eliminates the influence of the doctor-pharmacist relationship, promoting fairness and transparency across the healthcare system.

Summary of Progress

Overall, we were able to successfully complete the project with a little less than 1300 lines of code, with 56% of our code consisting of test cases. The project workflow consists of 8 compilation units, a total of 16 ml/mli files, a driver program, and a folder consisting of 2 csv files. In order to incorporate a blockchain algorithm, we had blockchain.json help us with I/O testing of our workflow to ensure surety in the confidence of our algorithm. Here is a full-length breakdown of our project structure

/bin:

- **main.ml:** The main entry point of the application, where the execution logic of the blockchain-based prescription validation system is defined.

/data:

- **accounts.csv:** Contains information about users (patients, pharmacists, doctors) participating in the system
- **blockchain.json:** Stores the blockchain state, including blocks and transactions, allowing the system to maintain a history of validated prescriptions.
- **tasks.csv:** Holds tasks or operations related to the blockchain, such as transactions or prescription validations.

/lib:

- **account.ml** and **account.mli**: Define the `Account` module, which might handle user-related functionalities like registration, authentication, and profile management.
- **authenticator.ml** and **authenticator.mli**: Implement authentication mechanisms, ensuring secure access to the system for patients, pharmacists, and doctors.
- **block.ml** and **block.mli**: Define the structure and operations of individual blocks within the blockchain, including their creation, validation, and linking to the previous block.
- **blockchain.ml** and **blockchain.mli**: Handle the entire blockchain logic, managing blocks, transactions, and ensuring the integrity of the blockchain.
- **doctor.ml** and **doctor.mli**: Manage doctor-specific functionalities, possibly including prescription approvals and doctor-patient associations.

- **patient.ml** and **patient.mli**: Handle patient-specific functionalities such as requesting and managing prescriptions.
- **pharmacist.ml** and **pharmacist.mli**: Manage pharmacist-specific operations, such as verifying and validating prescriptions in the system.
- **task.ml** and **task.mli**: Define the operations or tasks associated with prescriptions within the blockchain.

Activity Breakdown

Pradhi - implemented the patient module, block and blockchain aspect of the program and integrated the testing for the files block, blockchain, and pharmacist.

Hrishabh –implemented the patient module, and did the Unit testing of the Account, Patient, Pharmacist, Task, Doctor

Nikil – implemented authenticator, account, task module and integrated the overall program to function properly with unit testing of authenticator, and account. Implemented the driver program to integrate all the functionalities.

Productivity Analysis

We were pretty productive, and worked according to the deadlines. The integration part took a great deal of time, as dealing with the new nuances of blockchain was an intricate process, and we had already had tasks in our csv file, so we had to retrace and add every voting task on the chain. Integrating these backend files to function well on the user's frontend took some time and it took some time in figuring out the graceful handling of errors. Finally, we could have handled TTD (test-driven dev) more productively, as there were several type errors and underlying root bugs that took us much longer than expected. Either way, we were able to successfully overcome these and bring our project to fruition.

Acknowledgement: this progress report format is based on Prof. Walker White's CS 4152 (Advanced Games) progress report.