Using Blockchain in PHR (Personal Health Record) Systems

Implementation Report

Ryan Raffoul

Supervisor: Dr. Samet

COMP-4990 University of Windsor

Implementation of Personal Health Record Using Blockchain Web Application

In this Implementation Report I will discuss all major aspects of the Personal Health Record Platform Web Application. This includes the Client, Server/Database, Blockchain, Interaction of Components, Speed, Scalability, and Security.

Client

Purpose

The purpose of the Client is to give a user interface to control the main features of the application. There are 2 different management platforms for this application, the patient view and the healthcare professional view. The user interface for the patient view will include functionality for the patient to be able to create an account for the platform, create a PHR, view their own PHR, and control which healthcare professionals can view their PHR (using a Search Hierarchy). Healthcare professionals can create an account for the platform and search a database of patients to view PHRs they have access to. Patients find healthcare professionals using a search hierarchy and then they give access to the healthcare professional to view. The process of how this is done is discussed below. Some additional features are a patient can view all their past PHRs and be able to view the PHR in pdf format. The web3.js libraries will be used to connect to the Blockchain and get information from the smart contracts.

Patient View Functions

1. Create Platform Account using Personal Info
2. Login to the Platform
3. Create a PHR
4. View PHR using UI or pdf form

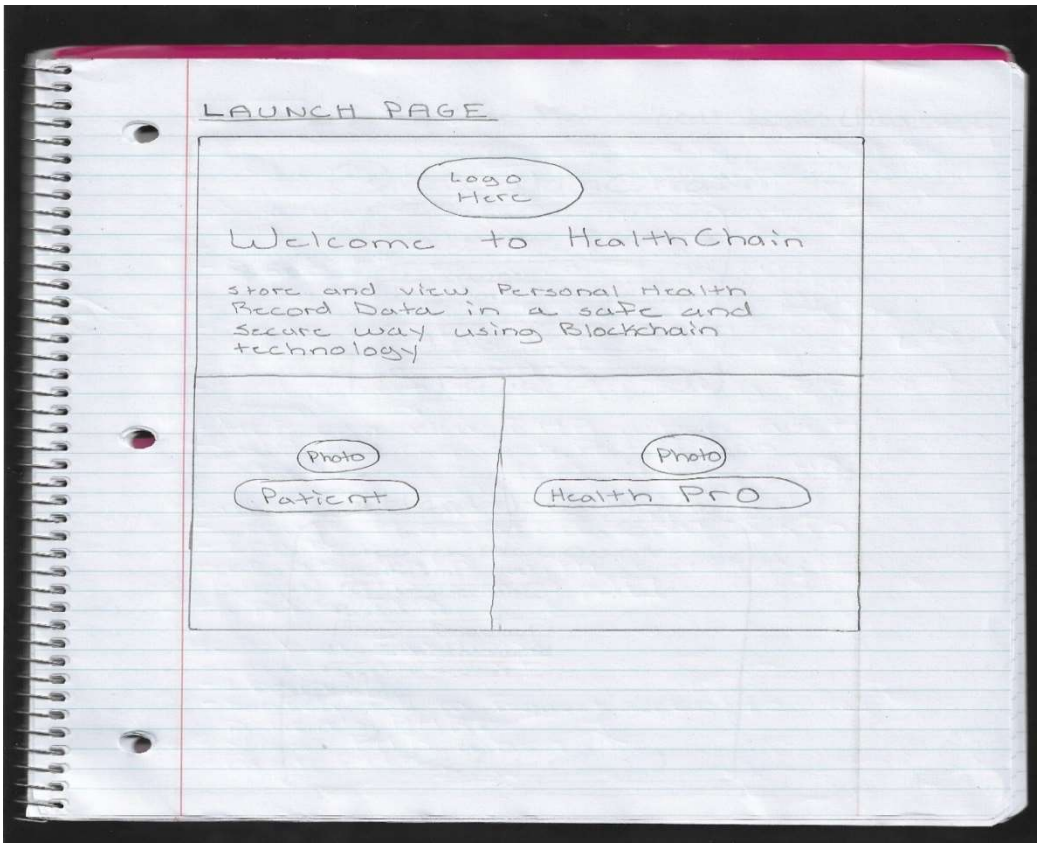
5. Give Access to a Healthcare Professional using a Search Hierarchy
6. Create a new updated PHR

Healthcare Professional View Functions

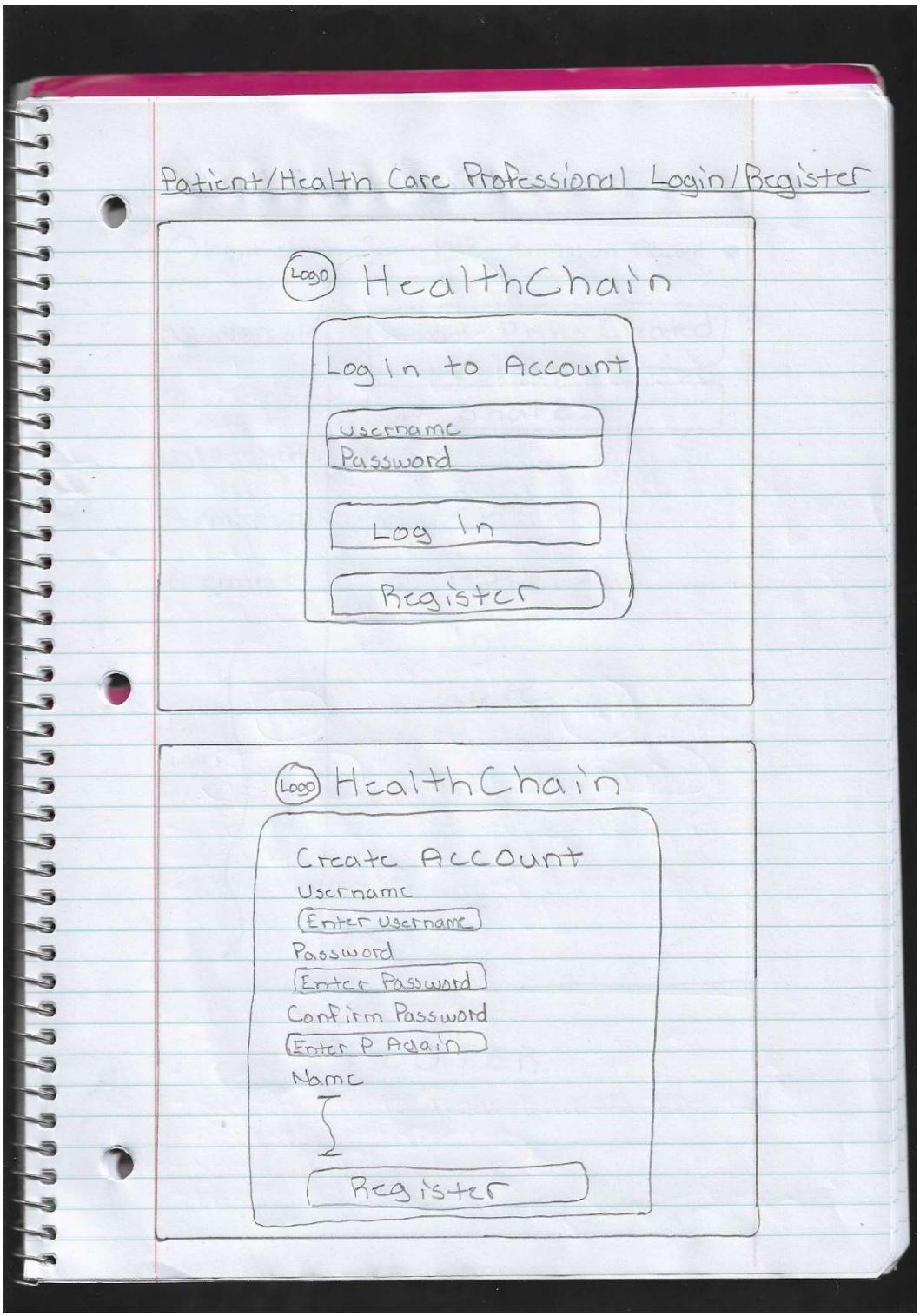
1. Create Platform Account using Personal Info (including specific profession)
2. Login to the Platform
3. Search Database for a Patient's PHR they have access to
4. View a Patients PHR in pdf format when given access

Web Pages

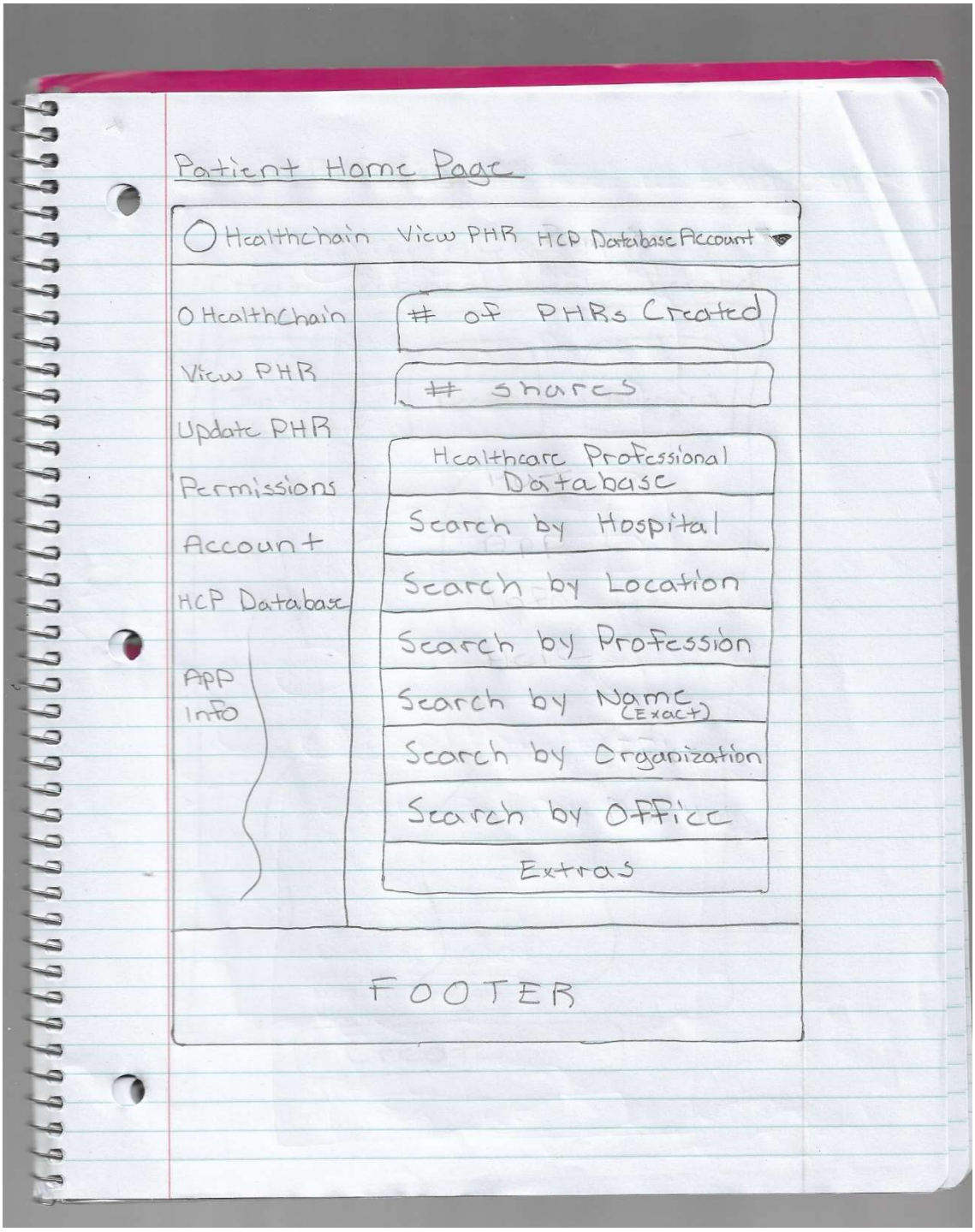
Launch - This is the welcome page to go to the patient or healthcare professional view.



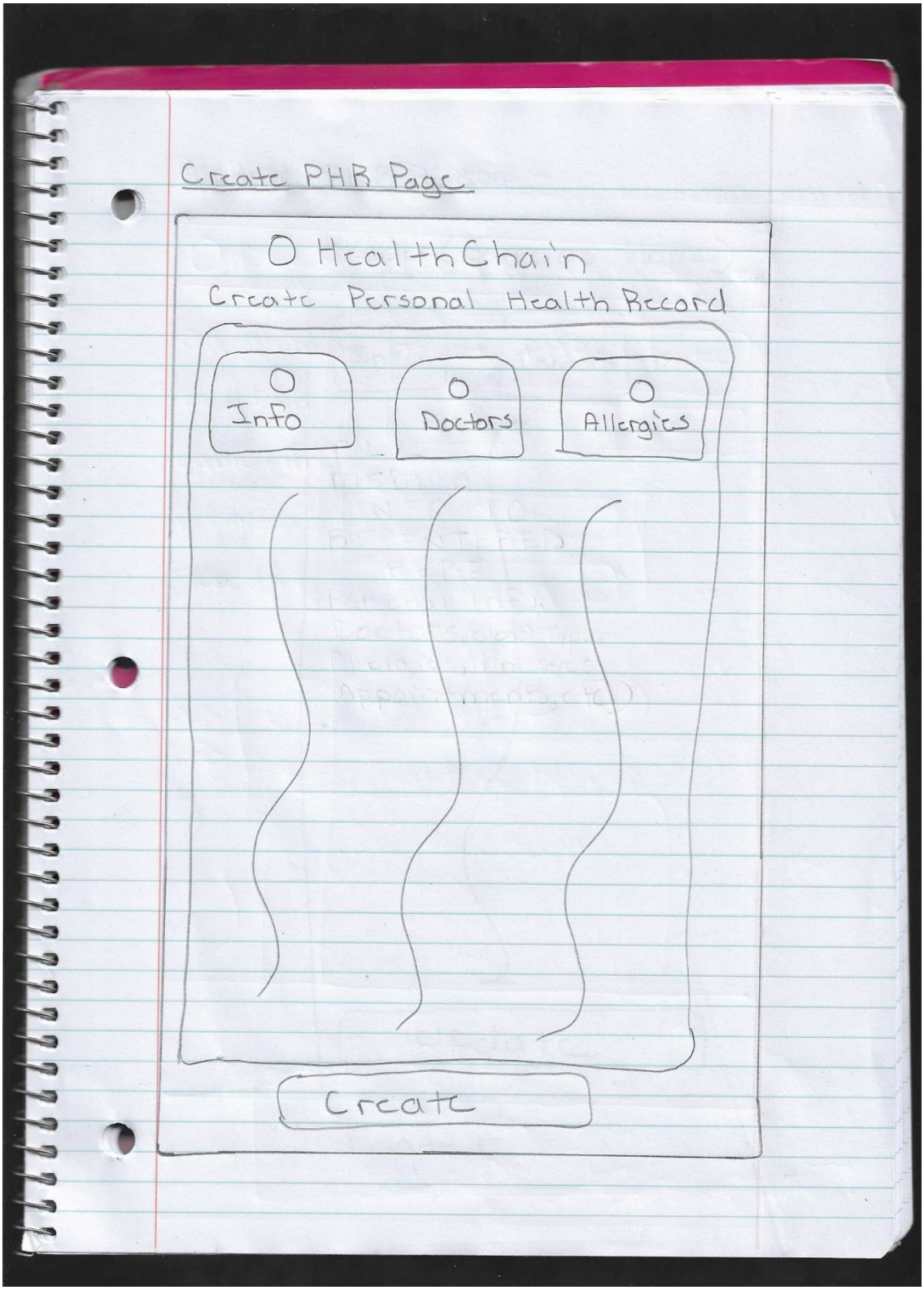
Patient/Healthcare Professional Login/Register - Login and create account for both views. 4 different pages.



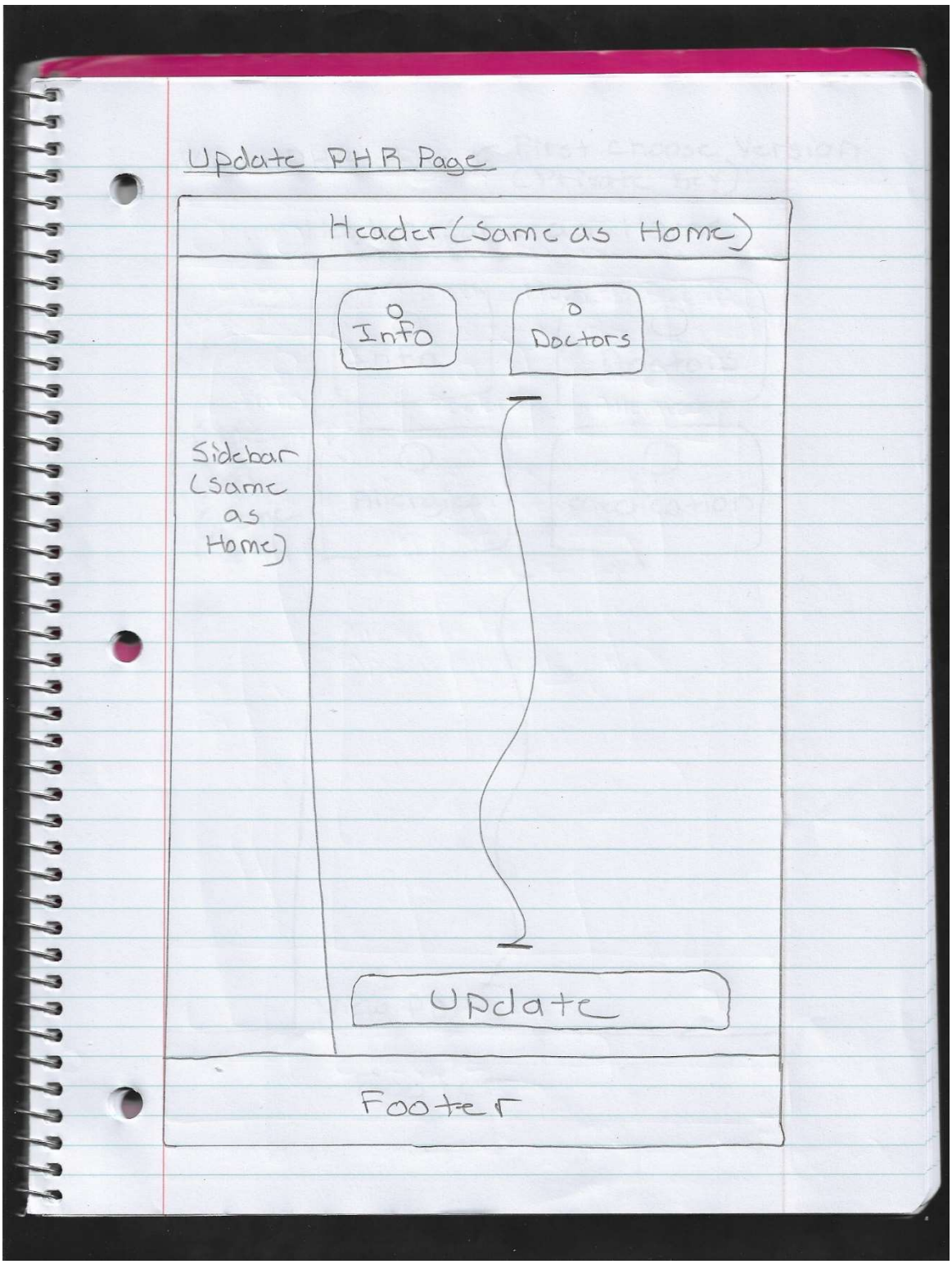
Patient Home - This is the home page for a patient to go to all main features. A search hierarchy is given here to find a healthcare professional. Pop-up windows will be used to follow each search hierarchy. Included some stats to fill the page.



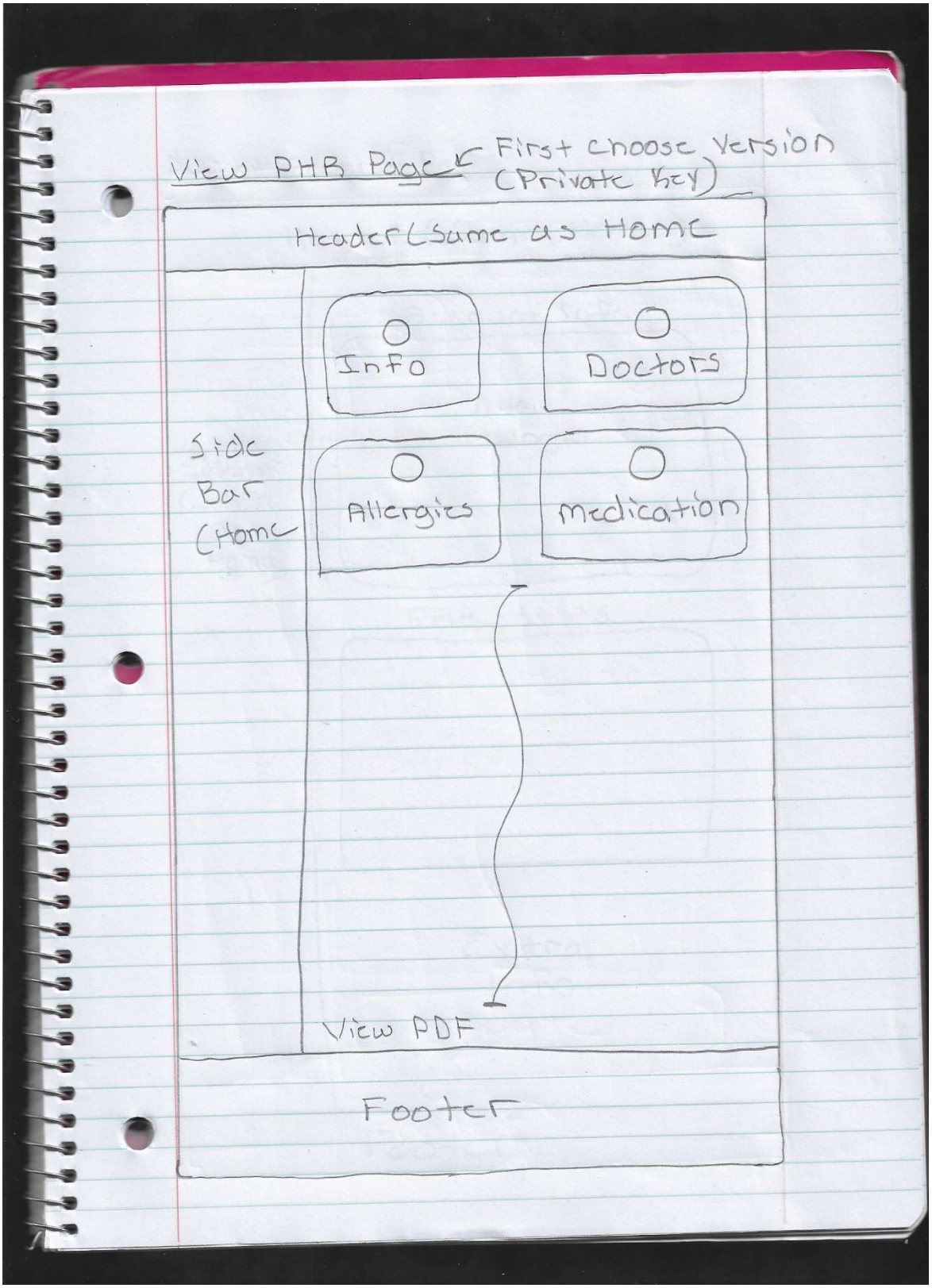
Create PHR - Patient creating a PHR. Patient can go to each component and enter info.



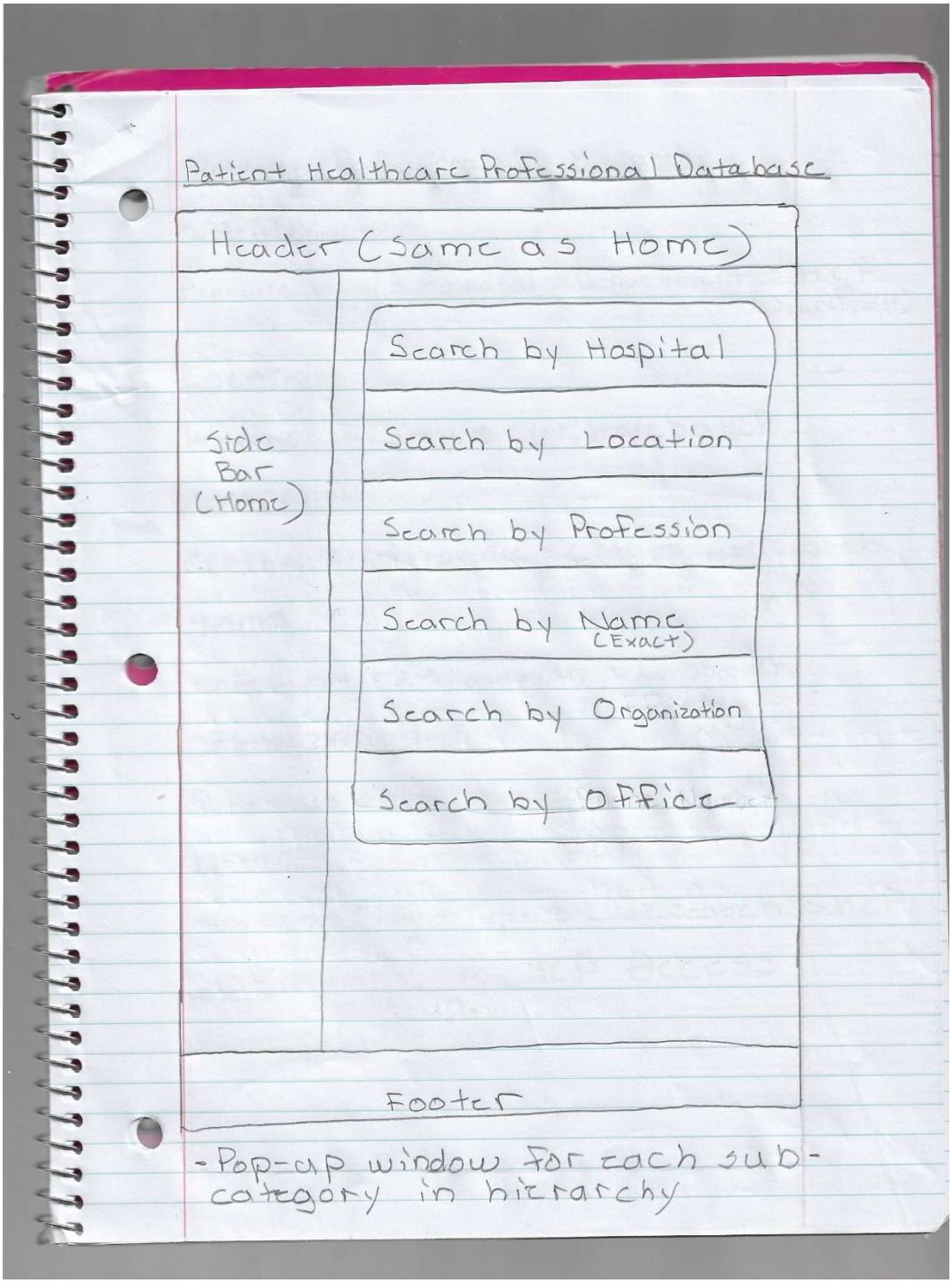
Update PHR - Patient updating their PHR. This creates a new PHR due to Blockchain immutability.



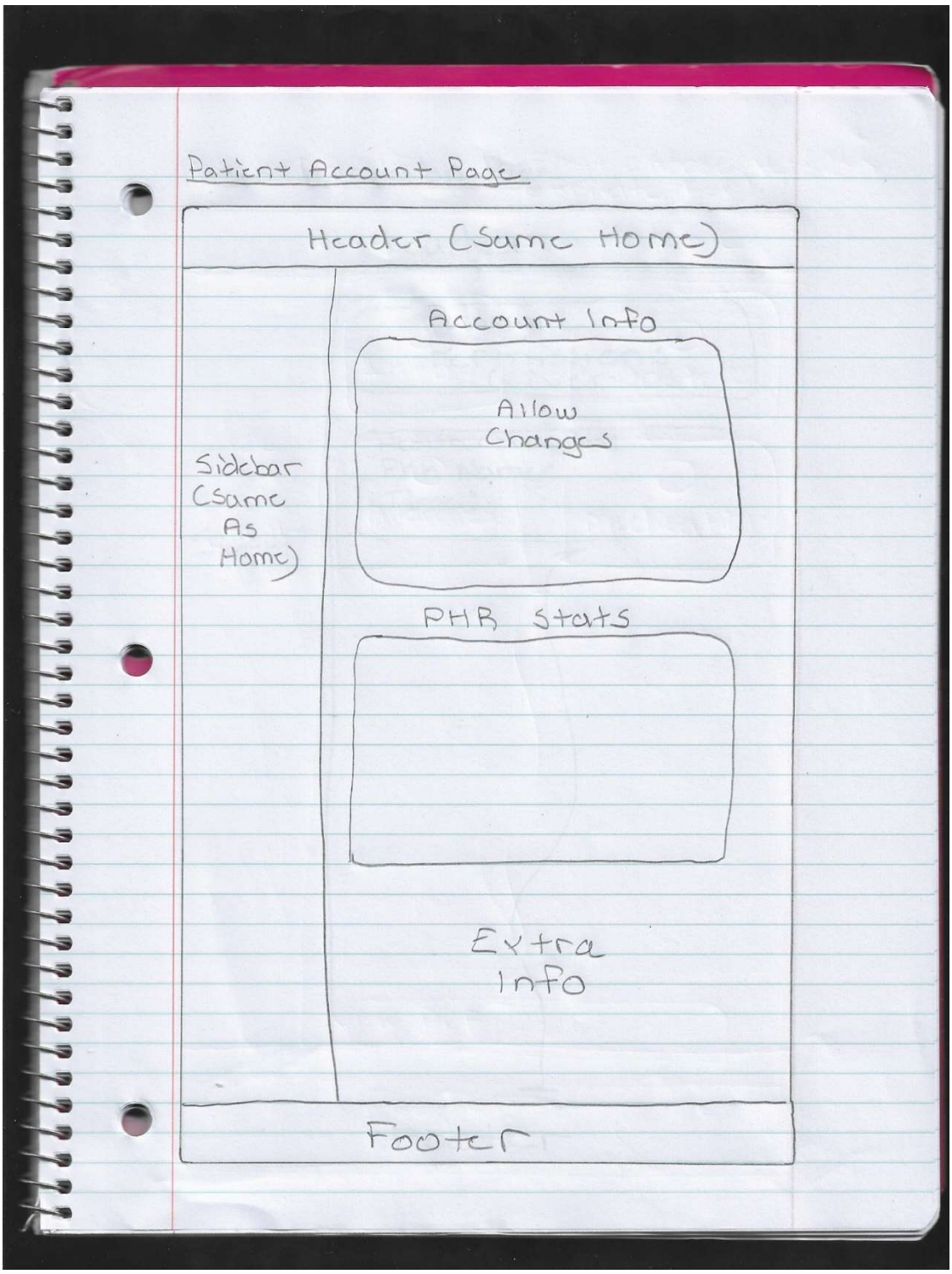
View PHR - Patient can view their PHR by selecting a component or viewing in a PDF version.



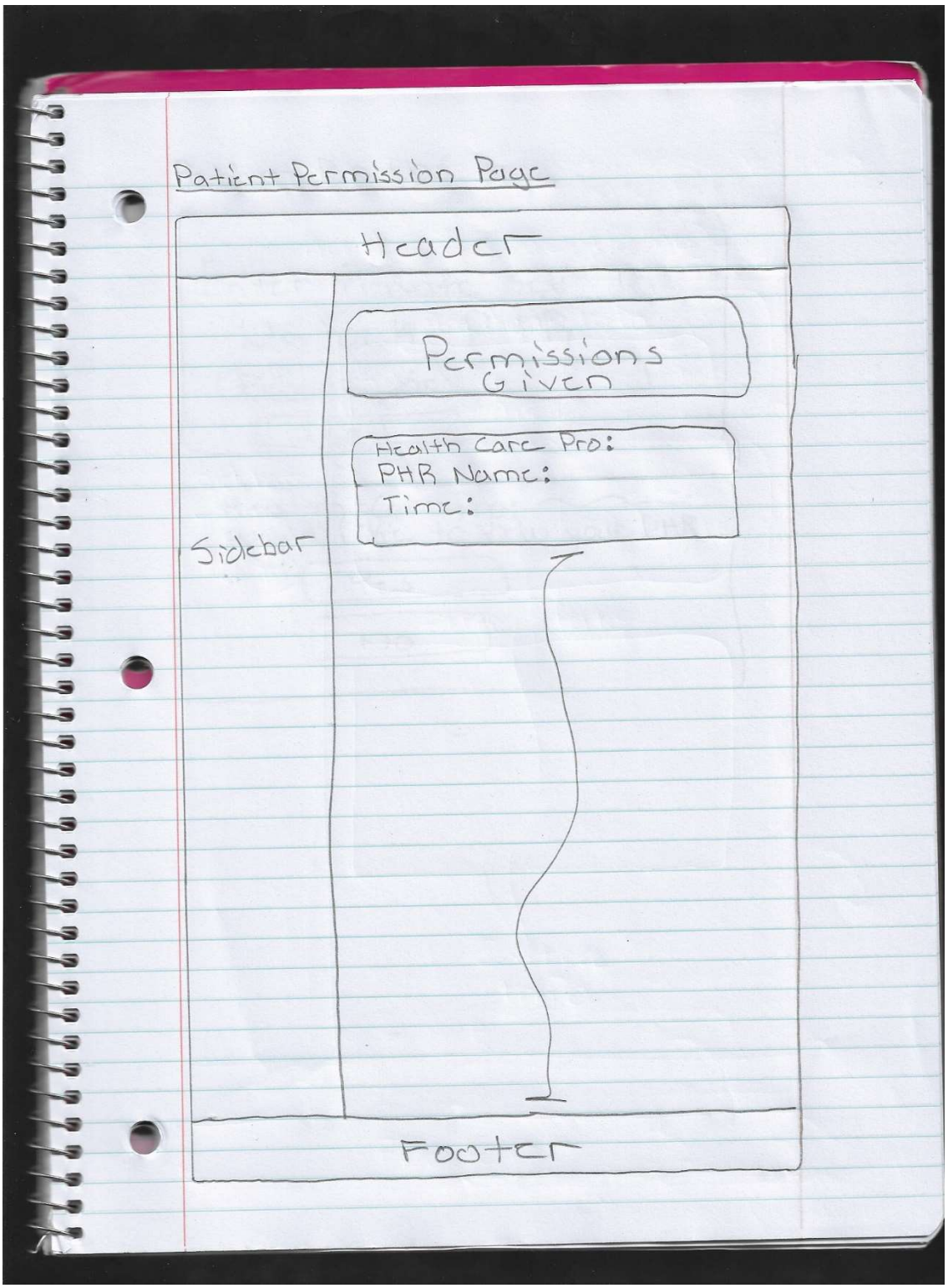
Patient Healthcare Professional Database - Patient can use the search hierarchy to find healthcare professionals and give them access. Hierarchy will be implemented using pop-up windows. List and Search functionality will be presented to find a Healthcare Professional.



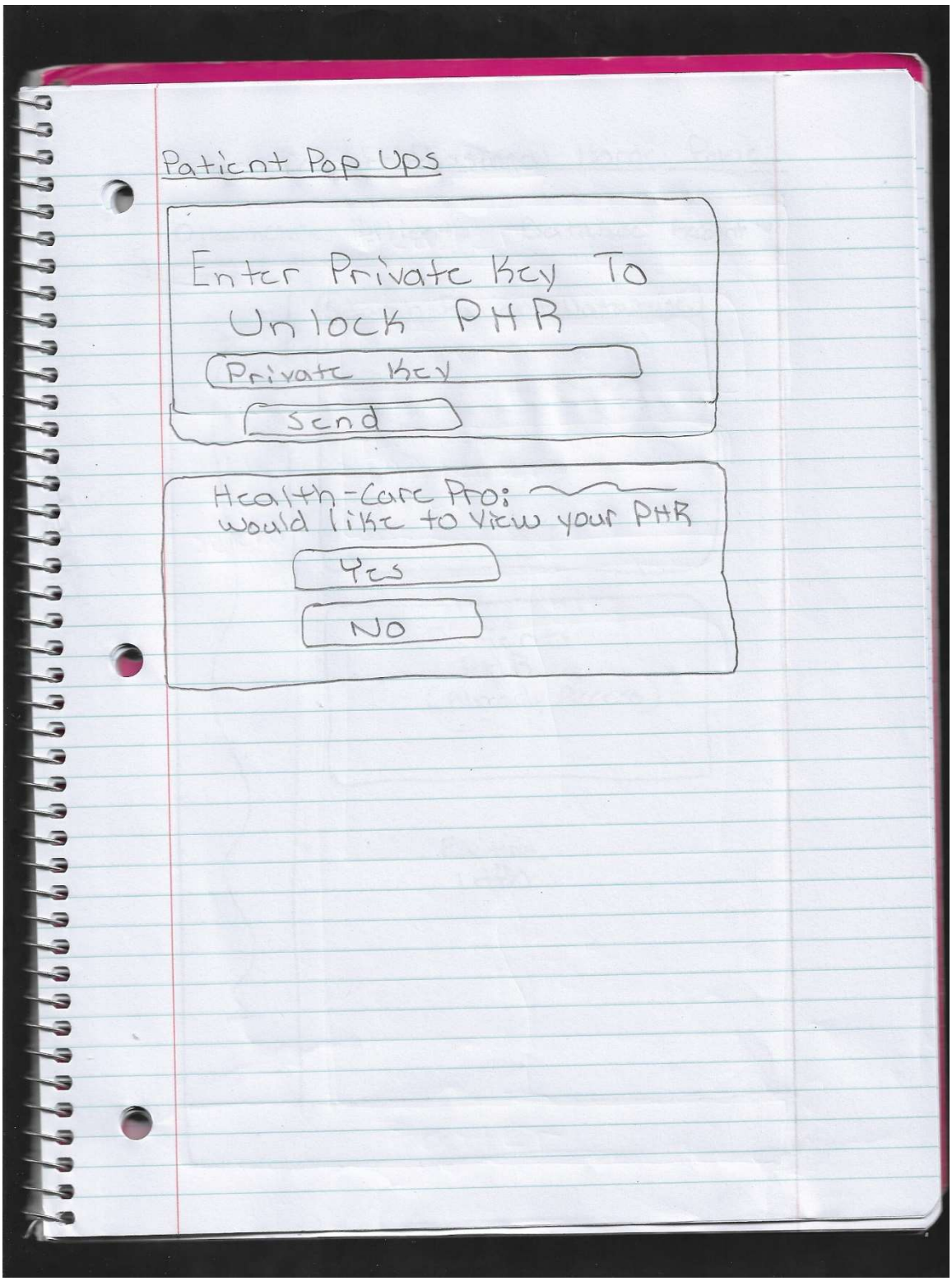
Patient Account - Patient can view/update all personal info stored in a database and see their PHR stats.



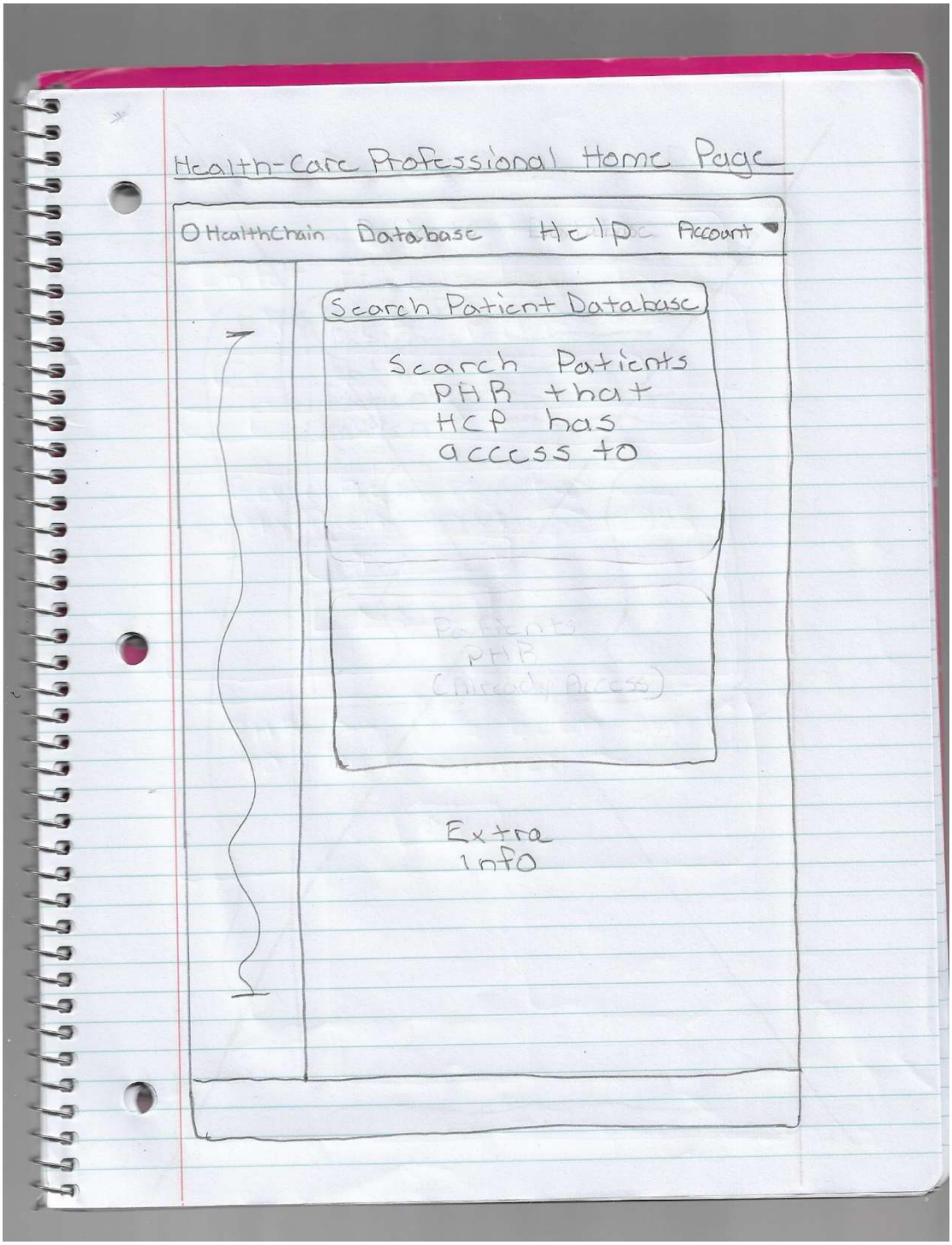
Patient Permission - Patient can see which healthcare professionals have permission to see their PHR.



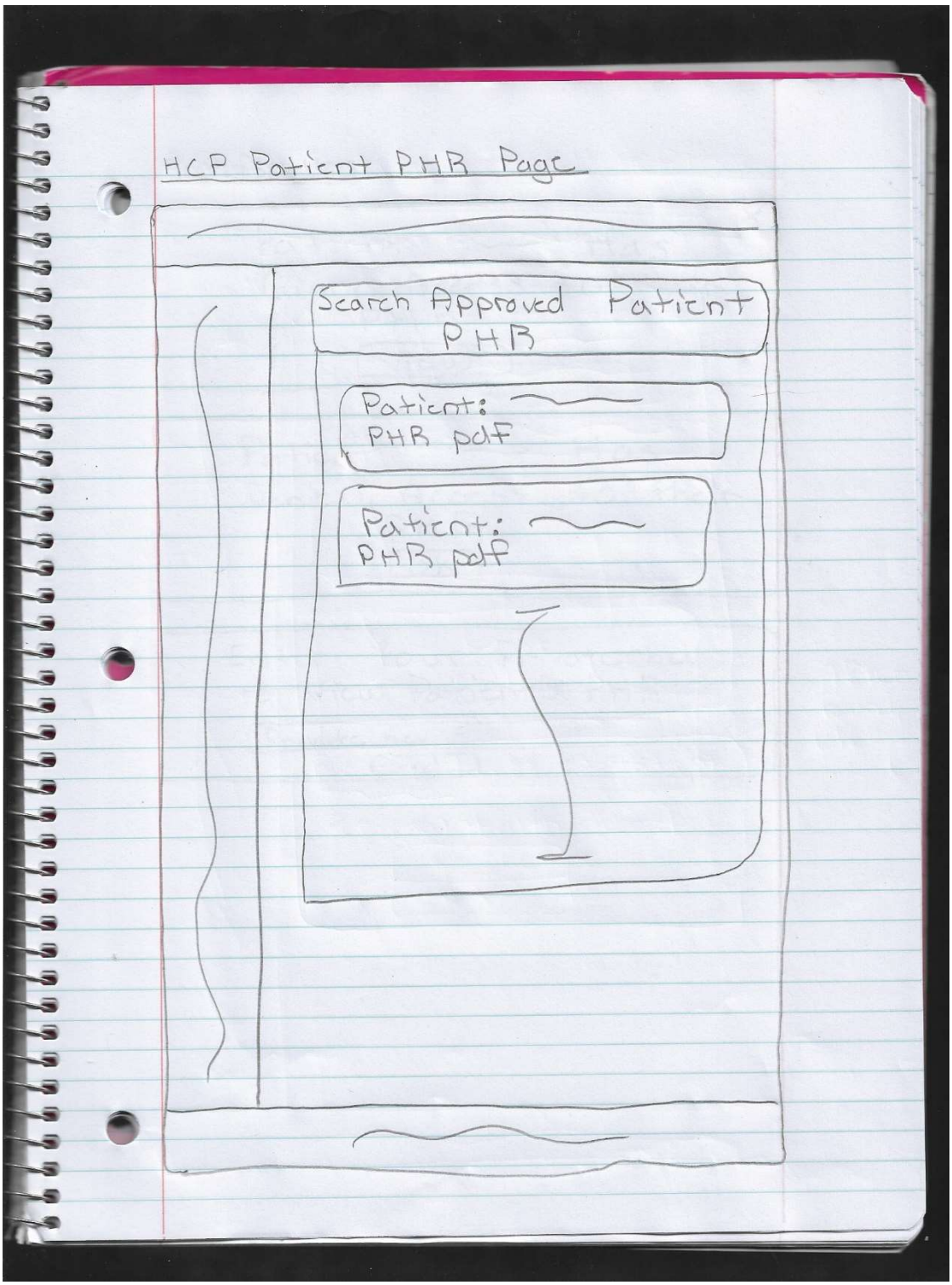
Patient Pop-Ups - Patient can enter private key to decrypt PHR and patient can give healthcare professional access.



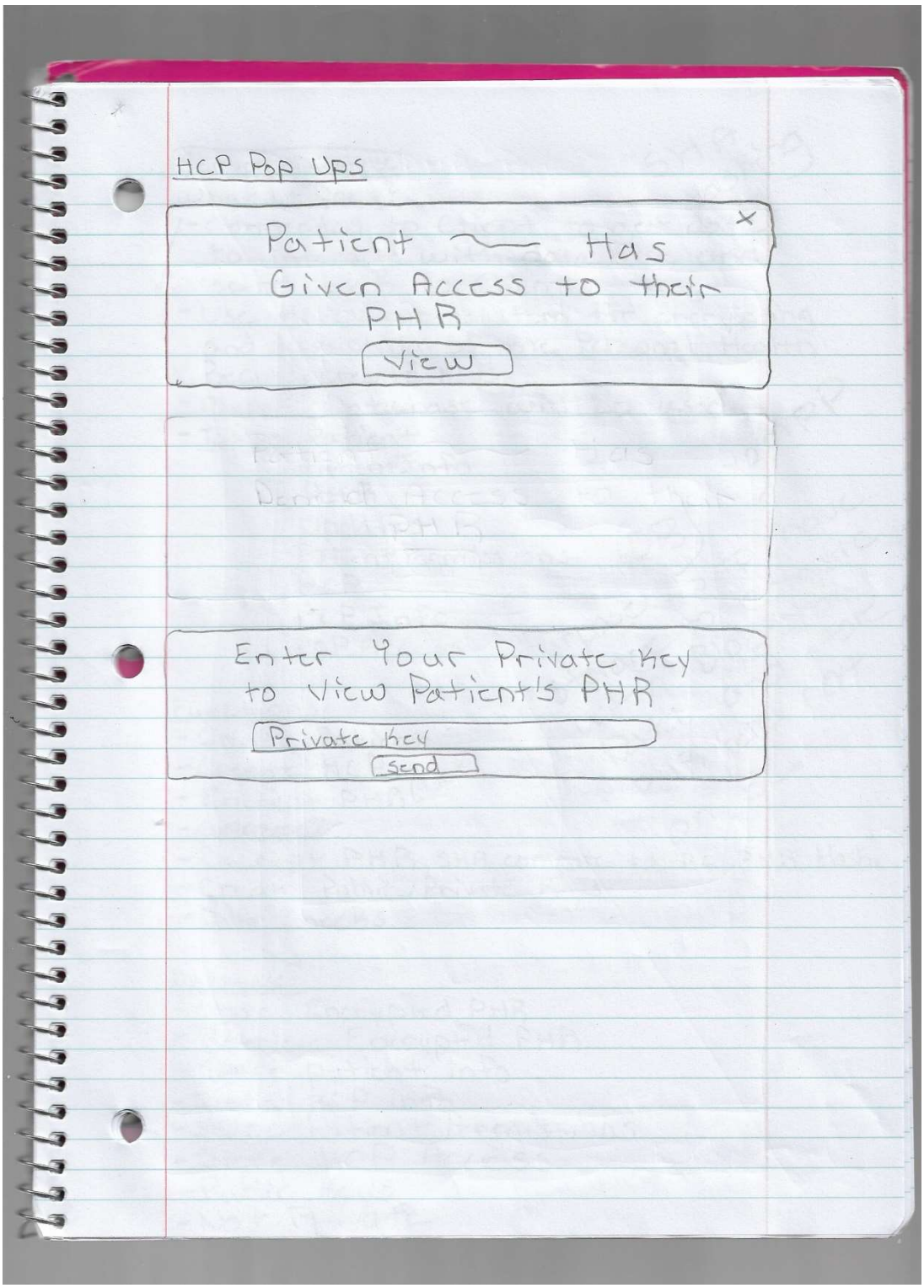
Healthcare Professional Home - This is the home page for a healthcare professional to go to all main features. Find PHR for patient's PHRs they have access to. List and Search for them.



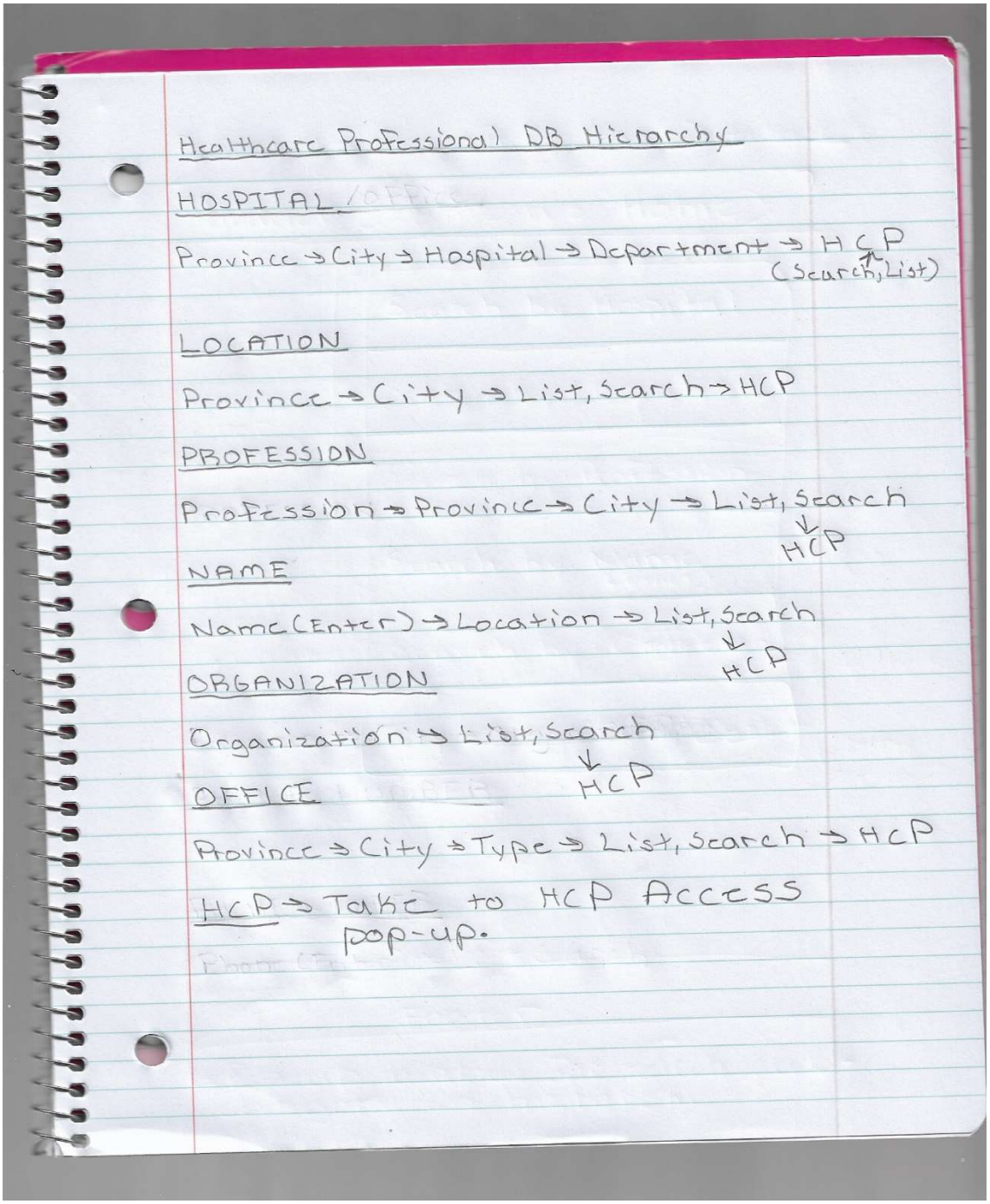
Healthcare Professional PHR - This is the page where healthcare professionals can search for PHRs that they have been given access to and then view these in pdf format.



Healthcare Professional Pop-Ups – Healthcare professional can see when a patient has given access and a healthcare professional can enter their private key to unlock a PHR to view.



Healthcare Professional Search Hierarchy (Patient View) - Patients have different options to find a healthcare professional to give access to their PHR. Implementation of sub-categories will be done using pop up windows. List and Search functionality given.



Programming Languages and Software

Programming Languages and Software used for the client will be:

- HTML
- CSS
- JS
- Web3.js libraries for interacting with Blockchain
- Metamask
- JSON
- XML
- Various APIs

Server and Database

Purpose

The purpose of the Server is to connect to the client to get data and then interact with the database for various functions. The Server will use the RSA cryptography technique for encrypting and decrypting the PHR and SHA hashing will be used to hash the PHR. The Server will also create random Private and Public Keys for each user of the platform. Keys will be created at random by an API when a user registers for the platform. The Server used will be XAMPP. It allows for easy connecting and is free to use for development.

Functions

The main functions of the Server are:

- Create Patient Account

- Create Healthcare Professional Account
- Use RSA cryptography for encryption and decryption
- SHA Hashing for the PHR
- Public and Private Keys creation
- Data Checks

Database

A MySQL Database will be used for this Personal Health Record Management Platform. The database will be responsible for storing the encrypted PHR, retrieving encrypted PHR, store patient account info, store healthcare professional account info, store search hierarchy, store permissions and accesses, store public keys, and do not store private keys (users have these in a safe spot, no one knows them). The main database tables will be Patient, Patient Info, PHR, Patient Permission, Healthcare Professional, Healthcare Professional Info, Healthcare Professional Access. Some more tables will be added later when needed.

Programming Languages and Software

Programming Languages and Software used for the Server/Database will be:

- PHP
- MySQL
- XAMPP
- Helpers
- APIs for cryptography and hashing

Blockchain

Purpose

The purpose of the Blockchain is to store the hashes of patients PHRs and to be able to store patients and healthcare professionals authorization of PHR data sharing. This will be done using the functionality of Smart Contracts which will be deployed on the Blockchain.

Ethereum

An Ethereum Blockchain will be created for this application. This will be a local Blockchain that will be created on my machine with a few authority nodes. The Blockchain will provide no downtime, fraud, control, or interference from a third party. All basic Blockchain properties will be followed.

Properties

A Permission Blockchain will be created. A Permission Blockchain is a Blockchain where nodes/accounts need permission to be able to join the network. This makes sense for a PHR platform where authorization and knowing who is on the network is key. A permission Blockchain is also fast and requires no crypto-currency. A Proof of Authority Consensus model will be used where blocks are validated by verified and approved accounts or nodes. Proof of Authority makes sense for this application since it is fast, secure, and easily scalable since blocks can be added quickly. All other properties will follow basic Blockchain properties.

Smart Contracts

There will be 2 main Smart Contracts for this application, the Create PHR Contract and the Healthcare Professional Authorization Contract. For the Create PHR Contract, mappings will be

used to store the PHR Hash in a safe and secure way. The data will include a patient's public key, PHR hash, PHR DB index, and a timestamp. Getters and Setters will be used to retrieve and set transactions. For the Healthcare Professional Authorization Contract, mappings will be used to store the PHR hash in a safe and secure way. The data will include a Healthcare Professional's public key, a patient's public key, PHR hash, patient name, and timestamp. Getters and Setters will be used to retrieve and set transactions. Mappings to mappings will be utilized to ensure patients can update a PHR, but a new version will be created due to the immutability property.

Programming Languages and Software

Programming Languages and Software used for the Blockchain will be:

- Solidity for Smart Contracts
- JS web3.js for interaction with the Blockchain
- Geth (Go Ethereum)
- Visual Studio Code
- Google Chrome
- Metamask
- Truffle

Interaction

Client ⟷ Server ⟷ Database

Client ⟷ Smart Contract ⟷ Blockchain

Create Account

1. User creates account with personal info. (Add to Database using Client/Server)
2. User is given a public key stored in database and a private key not stored but they need to store in a safe place. These are created randomly using an API. (Client/Server, Database)
3. If user is a patient, send to Create PHR Page. If user is a healthcare professional, send to healthcare professional home page. (Client)

Patient Creates Personal Health Record

1. User creates a PHR. (Client/Server)
2. Create a PHR Hash using SHA hashing. (Server)
3. Encrypt the PHR using patient's public key and store in Database. (Database)
4. Using Smart Contract mappings and structs, store public key, PHR SHA hash, DB index, and timestamp on the Blockchain. (Blockchain)
5. If successful, send a message to patient saying created successfully (Client)

   The above process is used when updating a PHR also.

Patient Views Personal Health Record

1. Use Patient's public key to get encrypted PHR. (Database)
2. Use Patient's public key to get PHR SHA hash, DB index, and timestamp from the Smart Contract on the Blockchain. (Blockchain)

3. Decrypt the PHR using the patient's private key (they enter, do not store). (Client)
4. Use SHA hashing to hash the PHR. (Server Database)
5. Compare the SHA hash of the PHR and the PHR SHA hash on the Blockchain. Also compare the indices. (Server, Blockchain)
6. If all equal, the patient can now view their PHR. (Client)

Patient Sharing Personal Health Record with Healthcare Professional

1. Patient finds a Healthcare Professional using Search Hierarchy and gives them permission to view their PHR. (Client)
2. Use Patient's public key to get encrypted PHR. (Database)
3. Use Patient's public key to get PHR SHA hash, DB index, and timestamp from the Smart Contract on the Blockchain. (Blockchain)
4. Decrypt the PHR using Patient's private key (they enter, do not store). (Client)
5. Use SHA hashing to hash the PHR. (Server Database)
6. Compare the SHA hash of the PHR and the PHR SHA hash on the Blockchain. Also compare the indices. (Server, Blockchain)
7. If all equal, continue else error message. (Client)
8. Encrypt the PHR using healthcare professional's public key and store in Database. (Database)
9. Using Smart Contract mappings and structs, store healthcare professional's public key, patient's public key, PHR SHA hash, patient name, DB index, and timestamp on the Blockchain. (Blockchain)
10. Send Patient notification stating PHR has been shared with healthcare professional.
11. Send Healthcare Professional notification stating they have access to Patient's PHR.

Healthcare Professional Views Personal Health Record

1. Use Healthcare Professional's public key to get encrypted PHR. (Database)
2. Use Healthcare Professional's public key to get PHR SHA hash, DB index, patient name, and timestamp from the Smart Contract on the Blockchain. (Blockchain)
3. Decrypt the PHR using the Healthcare Professional's private key (they enter, do not store). (Client)
4. Use SHA hashing to hash the PHR. (Server Database)
5. Compare the SHA hash of the PHR and the PHR SHA hash on the Blockchain. Also compare the indices. (Server, Blockchain)
6. If all equal, the healthcare professional can now view Patient's PHR. (Client)

Speed

The Blockchain will be fast due to it being a Permission Proof of Authority Blockchain with a Smart Contract. The permission property allows for fast transactions since only known accounts can join the network and no crypto currency exchange is needed. The Proof of Authority property allows for fast transactions since the algorithm for proof of authority increases the speed at which authority nodes can validate blocks/transactions. Smart Contracts will allow for fast retrieval of data as the mapping property allows for O (1) retrieval.

Scalability

The Blockchain will be relatively scalable due to the Proof of Authority consensus model. This is because only a few nodes are authority nodes, and they can valid blocks/transactions much quicker than a permission-less blockchain can with the proof of work consensus model. The Smart Contracts can also be scalable due to the mapping and struct properties. The last reason

this solution can be scalable is because only the PHR hash will be stored, not the entire PHR data. This will save space as a PHR can have large amounts of data.

Security

The Permission Proof of Authority Blockchain allows for enhanced security as opposed to a Permission-less Blockchain since nodes/accounts need permission to be able to join the network and create transactions. SHA hashing will also provide security and ensure the PHR cannot be tampered with at all. RSA cryptography will be used for encryption and decryption to provide the highest amount of security. RSA cryptography works by encrypting a PHR using only a patient's public key and then the only way to decrypt the PHR is by using the patient's private key, which is not stored anywhere in the platform, only the patient has it. So even if an attack causes the encrypted PHR to be stolen, there is no way to decrypt the PHR.

Conclusion

Discussed above was each of the main components of the Personal Health Record Platform and how each one of the components will be able to interact with each other to provide the desired functionality.

References

Yaga, D. , Mell, P. , Roby, N. and Scarfone, K. (2018), Blockchain Technology Overview, NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/NIST.IR.8202

Roehrs, Alex & André da Costa, Cristiano & Righi, Rodrigo & Silva, Valter & Goldim, Jose & Schmidt, Douglas. (2019). Analyzing the Performance of a Blockchain-based Personal Health Record Implementation. Journal of Biomedical Informatics. 92. 103140. 10.1016/j.jbi.2019.103140.

Park YR, Lee E, Na W, Park S, Lee Y, Lee J
Is Blockchain Technology Suitable for Managing Personal Health Records? Mixed-Methods Study to Test Feasibility
J Med Internet Res 2019;21(2):e12533