

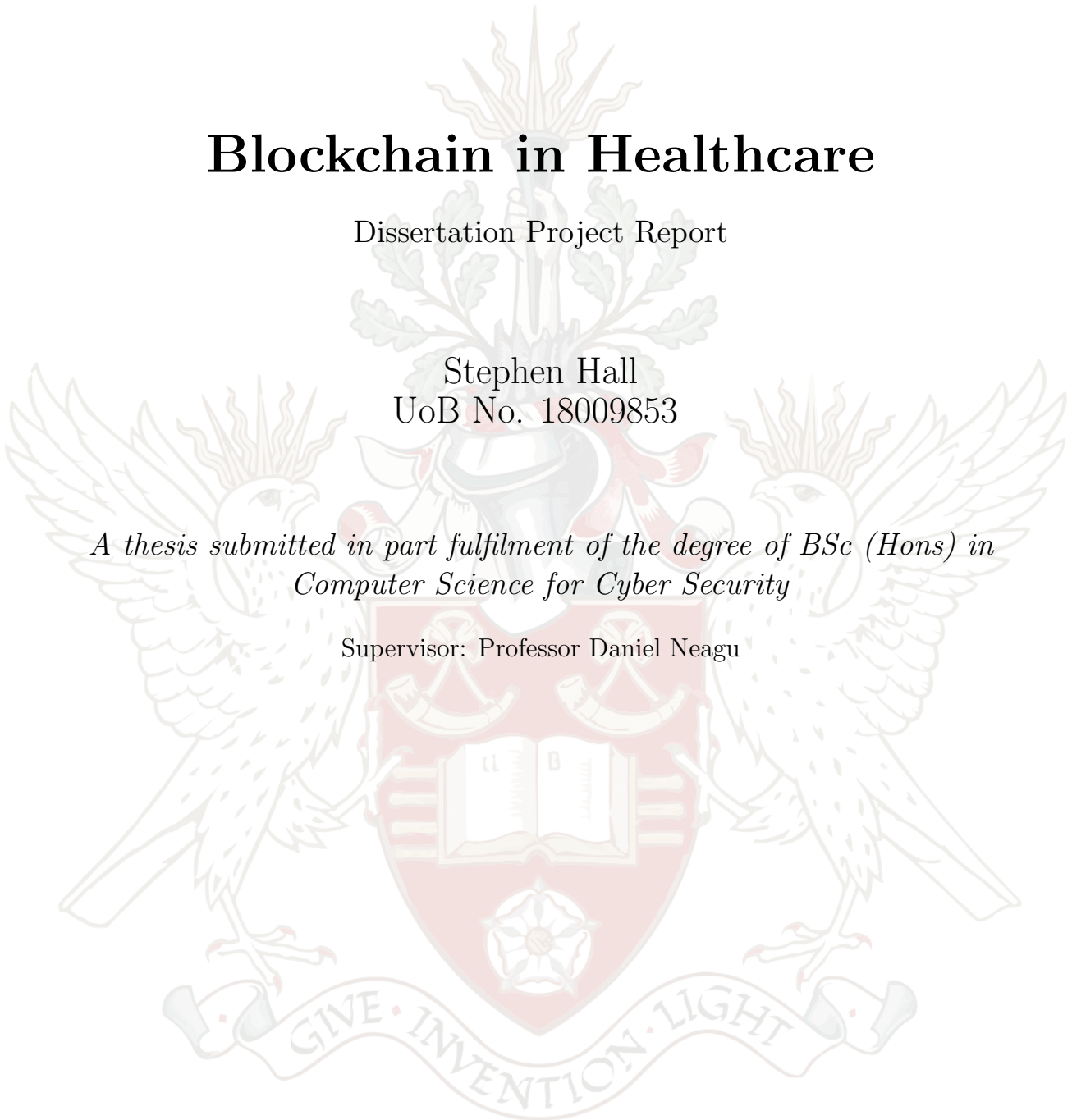
Blockchain in Healthcare

Dissertation Project Report

Stephen Hall
UoB No. 18009853

*A thesis submitted in part fulfilment of the degree of BSc (Hons) in
Computer Science for Cyber Security*

Supervisor: Professor Daniel Neagu



Declaration

The candidate confirms that the work submitted is his own and that appropriate credit has been given where reference has been made to the work of others. The candidate agrees that this report can be electronically checked for plagiarism.

Stephen Hall

Acknowledgements

Thank you to my Project Supervisor, Professor Daniel Neagu who's help has been invaluable throughout the course of this project, both inside the weekly meetings and with prompt email replies. A special thank you to my Grandad, who first set me on the path of reading and being my light to greater things - as without you, I would not of grasped all the opportunities that crossed my path. I would not be the man I am today, nor reached my fullest potential without you.

Abstract

The author has created a web base blockchain database interface which allows for a user to access information about themselves or patients, with data pulled from the database with a key provided by a blockchain. This application is easy to use for users irregardless of their technical abilities. In addition to this software, there is a report regarding where blockchain technologies currently stand, what blockchain is and how it works, the design of the project, how it was tested along with the Legal, Ethical, Social and Professional issues associated with this project.

Contents

1	Introduction	2
1.1	Project Description	2
1.2	Aims & Objectives	3
1.3	Challenges	3
1.4	Personal Interest	3
1.5	Report Overview	4
2	Literature Review	5
2.1	What is Blockchain?	5
2.1.1	Proof of Work	6
2.1.2	Proof of Authority	6
2.2	Smart Contracts	7
2.3	Review: Scalable and secure access control policy for healthcare system using blockchain and enhanced Bell–LaPadula model - Kumar et al.,2021	7
2.4	Review: Scalability Challenges in Healthcare Blockchain System—A Systematic Review Mazlan et al.,2020	9
3	Requirements and Analysis	11
3.1	Basic Requirements	11
3.2	Security Requirements	12
3.3	Usability Guidelines	13
4	Prototype Description	14
5	Design, Implementation and Testing	17
5.1	Blockchain Design Overview	17
5.1.1	SIEM Implementation	18
5.2	User Interface Design	19
5.2.1	Index	19
5.2.2	Login	20
5.2.3	Register	21
5.2.4	Welcome	21
5.2.5	Results1	22
5.2.6	Admin Panel	22
5.2.7	Admin - Audit Log	24
5.2.8	Admin - Information to Blockchain	24
5.2.9	Admin - New User	25
5.2.10	Admin - User Roles	25
5.3	Testing	26
5.4	Testing... continued	35
5.5	Implementation	36
5.5.1	Blockchain Integration	36
5.5.2	Audit Log	37

5.5.3	Interactions with the Database	38
6	Results & Discussion	40
6.1	Changes Undergone	40
6.2	Limitations & Effectiveness	40
6.3	Legal, Ethical, Social, and Professional Issues	41
6.3.1	Legal Issues	41
6.3.2	Ethical Issues	42
6.3.3	Social Issues	42
6.3.4	Professional Issues	42
7	Conclusion	43
8	References	45

1 Introduction

We, as the Human race are skyrocketing through the Digital Age, coming to grips with new innovations - specifically with the introduction of the Internet of Things and Cyber Security - the general public are becoming more self aware of digital threats, putting their data at risk.

One of the few things that will happen to each Human being at one point or another, is a run-in with the Health services of whichever country they reside within. From birth, along with every other subsequent visit to the General Practice or Hospital is fed to their internal system. All doctor notes, prescription, medical issue, everything is on record for theoretically anyone to view. As such, should not you have access to the very same information which is accessed by your doctors?

Patients currently have access to a limited amount of information through the various interfaces available to them through their current health service they are established within. Needless to say, this doesn't include all their information, only the bare minimum. However with that access, security comes into the question. How can we ensure that an app, accessible by anyone with a modern phone is secure? How can we ensure that the architecture behind the app and the health service's systems themselves are secure? Is there anyway we can easily answer these questions without analysing the backbone of what protects our data currently?

Enter Block chain, a technology which will theoretically solve our issue with Security. Allowing patients to access to their information in a secure manner. This will also allow medical professionals to modify the information and keep track of it, all with the patient being able to view all changes made.

1.1 Project Description

This thesis will therefore explore the opportunities that block chain provides, as well as the Legal, Social, ethical and professional issues which inundates it.

My main goal is to create an interface which uses blockchains to securely secure patient data. This will mainly be achieved via a web interface to ensure that this is indeed user friendly. With there being multiple access levels such as those used in the Bell-laPadula model which will be shown later on in figure 2 which will be able to perform different actions.

Cyber Security will be a pivotal point of my project as this is what my degree focuses on. Therefore I will be exploring the security side more than the development and manufacturing side - as Cyber Security has become increasingly more relevant in our production of systems applications. However, as this project's focus is "Blockchain in Healthcare", our focus should on how we can sanitise our project to prevent against malicious intent. This involves us donning our "Blue Team Hat" to analyse whatever design we come up with and to ensure that our client's sensitive data does not get into the wrong hands.

Additionally we will be looking at various different blockchain technologies which could assist us with determining the suitability of blockchain within the healthcare setting. These technologies are still in the early days of production, however they may potentially be of use.

1.2 Aims & Objectives

Scenario

Using a secure system, allow for users of different access/security levels to view/edit patient data while also auditing the systems to add and remove users as and when necessary.

- Allow for ease of use for both technical and non-technical users
- Ensure the data stored on the blockchain is secure
- Audit all connections and changes
- Maintain a high level of security and ensure that sanitation has taken place to reduce risks

1.3 Challenges

The main challenge throughout this project will be two fold; the Security aspect and the data storage aspect.

Blockchains ordinarily cannot store a huge amount of information, thus information should only be stored on it which you would not want to change. For example; a DPD parcel is shipped out - you would want to use blockchain to establish where about in the distribution chain it is at. This information would be immutable and would be quick to inspect. If we would need to know specifically more information, we would have to provide a link to a 3rd party data storage service which would provide additional information which we could not fit in to the blockchain.

Additionally, an easy way to access the blockchain would have to be provided, which would require no software which would be difficult to use. The likely best solution would be an app based system or even better, a web based system. With Cyber Security in mind, it would be best to minimise the amount of the external connections - however by keeping it completely offline, you'd be sacrificing accessibility. Which you could easily need in a pinch if you have an accident.

1.4 Personal Interest

This was a project which peaked my interested, especially after finishing my student placement at a Cyber Security Company. After 14 months working full time I returned to university for my final year, setting my sights ahead, and more specifically on my Final year project, however - I wanted to do something Cyber Security related which would give me more insight later on in my career in potentially up and coming technologies. This is when Professor Daniel Neagu mentioned that a group of 3 French interns were working on a

project which related to Blockchains in healthcare. However their focus of the project was to get a working prototype. While they did manage to get a portion of it working through Python scripts they ran into some issues.

While ordinarily, this project would be continued by just taking over production - I instead decided to take it in a different direction, using different software and focusing mainly on the Cyber Security aspect and how we could store information within a blockchain. As these are the problems which I have found to be the current existing issues within blockchains, at least from a security perspective.

1.5 Report Overview

This report contains five sections; "Introduction", "Literature Review", "Requirements and Analysis", "Prototype Description" and lastly "Conclusion".

The first section - Introduction briefs you on what I plan to accomplish in this project, the challenges I will potentially face and my own personal interest in this field. This will also go in depth for a use case for why we need to have a secure technology like blockchains to allow for users to have access to their data.

The second section, the literature review will give you an background on blockchain technology, the quirky features it has and some information found in other papers which reflect how I could potentially structure my project to avoid pitfalls in the project. This will also discuss what is currently happening in the blockchain world - if there has been any recent breakthroughs and some recent relevant papers.

Requirements and analysis, Section three will give a rundown of the project, a list of features, the UI and the running theory of the project. I will also state the methods which I'll use to evaluate my project, including justifying my solutions.

The prototype designs will discuss what I have accomplished in my prototype and also my future plans for the project.

Lastly, the conclusion section will evaluate what is left to do on the project and round off what I believe is essential for the project to undertake.

2 Literature Review

2.1 What is Blockchain?

A blockchain is a chain of 'Blocks' which contains information. These chains can be described as a distributed ledger, where anyone can view their contents. Originally, they were designed as a way to timestamp digital records, so if anyone had tampered with the blocks - it would be immediately obvious. The way this would be detected would be through the block's individual hash. (Simply Explained 2017)

A hash is a 'screenshot' of how the block looked when it was last updated. If a value/attribute is changed even slightly - then the hash would change as well. This means that it is no longer the same block. (Blockholic 2020)

Each block not only contains the hash, but it also contains: Data, the Hash of this Block, and the Hash of the previous block.

The blockchain, as previously mentioned is made of a chain of multiple blocks. Each only knowing the hash of the block it was attached to. To put this into perspective, envision three blocks.

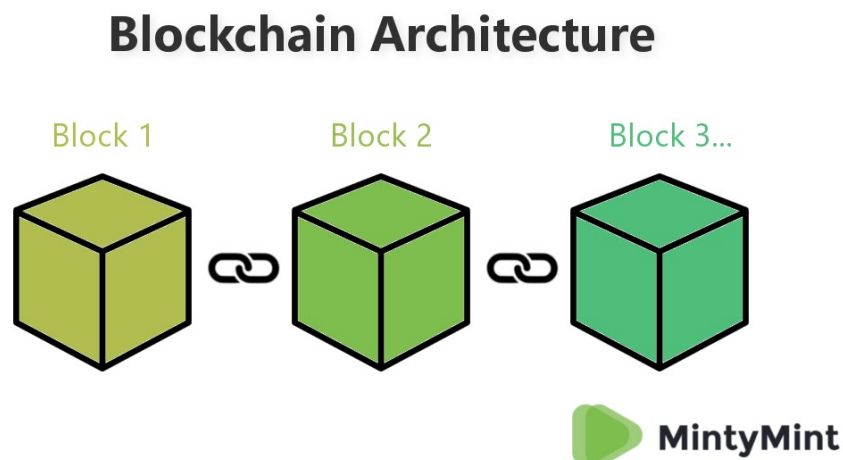


Figure 1: Blockchain - 3 Block Model (Kowalskiy, No Date)

Block 0, is known as the Genesis Block. As it does not possess the previous block's hash as it is the first one created. This gives it a clear and distinct starting point.

Block 1 stores the information of the Block 0's hash, as it was created after it and is therefore tethered to it. For example, this is what block 1 would contain:

- The data stored

- Block 1's current hash value
- Block 0's hash value (the prior block)

It is worth noting that once a block is chained together to a previous block - the previous block becomes immutable. This is because if the previous block were to change its information, the hash value would change. This means the block after it would have hash data of the previous iteration of the prior block. It would not know where to look as its 'address' has changed, making it effectively the new Genesis block. If you were indeed of a mind to change the data within a block, you would have to update the hash data of each subsequent block after it so they all know where to point to.

As if you change the hash of the block after it, its own hash would change. This means the block's hash after that would then be in the same position as it was before. Needless to say, depending on how huge the chain is - updating each subsequent block will take a lot of resources and time to accomplish.

There are a fair few systems at work to protect Blockchains from altercations some of these are: Proof of Work (PoW) and Proof of Authority (PoA)

2.1.1 Proof of Work

Proof of Work is a mechanism built in to blockchains which places a time limit on the updating previously created blocks and subsequent new blocks. Depending on the Proof of Work severity - in the case of bitcoin it could take up to 10 minutes to update/create new blocks. However, with Ethereum has a Proof of Work time which is considerably shorter. However a lot of power is consumed when handling the proof of work algorithm (Iredale, 2021)

Once someone found out they just needed to change the blocks after the one they changed, and accomplished that - they would be in the clear, right? Unfortunately not - as blockchains operate within a Peer to Peer network. This allows a user to receive a full copy of the blockchain. When a block chain is created, it is sent out to all of its peers who validate the entire chain - if a block's hash has been tampered with, they will reject the new chain. Therefore, if an attacker wishes to ensure that the blockchain is modified, they will have to change the Proof of Work for over 51% of network's blockchains, which would be a task to behold. This is all covered under Proof of Work. (Bogdanov, 2021)

2.1.2 Proof of Authority

This is a relatively new concept - the basic idea is that while normally, when you view a public blockchain, your identity does not need to be confirmed to view and download what's on the chain. However, with PoA requires a user to stake their identity and on the line when they are validating new blocks, or validating modifications of previous blocks. (Binance Academy, 2020)

While this is impractical for Public Blockchains, it's ideal for Private, Federated and Hybrid blockchains due to its link with Bureaucracy - everything needed to be done by the

book. You can then track who makes changes within a blockchain, and if they attempt to push the changes to other blocks, or change more than 51% of the network's blockchain.

2.2 Smart Contracts

A huge leap in the blockchain technology came from the introduction of Smart Contracts - this allows for automation of tasks through the use of scripting.

Smart contracts work through scripts which have been embedded into a blockchain block. Once the prerequisites of this script has been met; eg. The script looks for a single bitcoin to be sent to the block - it then forwards that bitcoin towards the company's wallet and returns a product-code for a piece of software which is emailed to the customer's email address. This will then add another block to the chain to ensure that the transaction has taken place.

Smart contracts can call other smart contracts - which effectively allows them to execute more complex jobs by chaining them up. (Rosi, 2020)

The greatest advantage of smart contracts is that they're fully transparent and independent. There is no third party which you need to trust. Everyone on the network can view the blockchain and see exactly what it does - which allows for the autonomous script to be executed when prerequisites have been met. As soon as the conditions have been met, the smart contract is executed instantly. There's no waiting for any third party to verify that the conditions have been appeased, everything is executed flawlessly, transparently and reliably.

Cryptocurrency is one of the main users of Smart Contracts - with the highlight user being Ethereum. Unlike bitcoin, when validating a block - Ethereum takes seconds, not 10 minutes. This is because Ethereum uses a protocol dubbed 'GHOST'. (Peyrott, 2017)

GHOST allows for stale blocks 'waste' blocks which have been computed by other nodes, which would otherwise be repudiated as newer blocks have been created to be re-integrated back into the blockchain. This allows for a huge power save, as blocks do not need to spend as much computational power to generate new blocks. Which makes this more environmentally friendly than Bitcoin, as not as much electricity is used when mining new blocks.

2.3 Review: Scalable and secure access control policy for healthcare system using blockchain and enhanced Bell–LaPadula model - Kumar et al.,2021

This research paper focuses on two specific aspects; Scalable & secure access policies, and the enhanced Bell-LaPadula model. While the paper itself is not entirely relevant for what I want to do - as it does not include any mention of the patient within the 'enhanced' Bell-LaPadula model. However I believe it can be adapted to ensure that the patient can observe any changes and observe who has access to the data. I believe that their model is a rather good idea to keep important information secure by restricting access levels and linking them with their job titles.

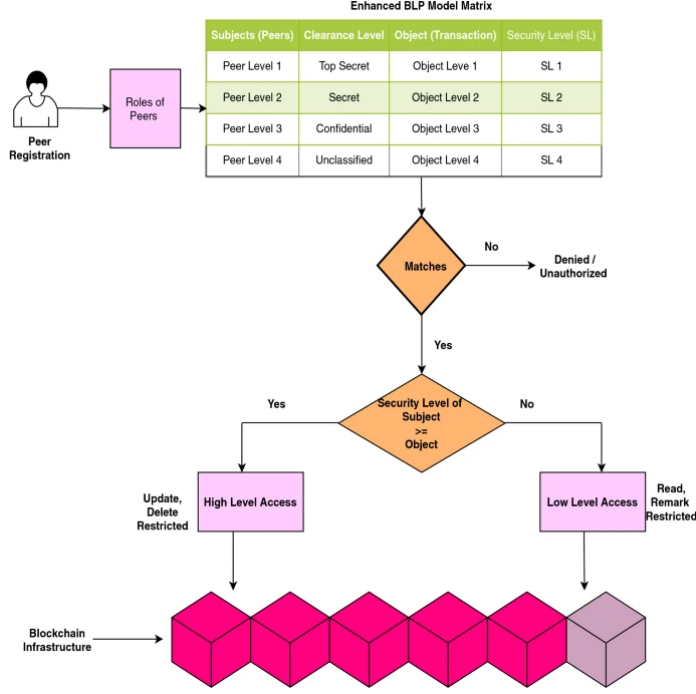


Figure 2: Enhanced Bell-LaPadula Model

As seen above, this model starts with the 'Peer Registration' which checks to see what 'role' the peer has. The role is then checked against a table which assigns it a security level. If there is no security level, the 'peer' is denied. They then run a check to see whether the object they are trying to access is available to them. It then does one of three things, if the 'Peer Level' is higher or the same as the 'High Level Access' then they are given update, with delete being restricted (as it should be to all levels). (Kumar, Tripathi, 2021)

If however, their security level (Peer level) does not allow them 'higher level access' then they are given low level access, allowing them to read and remark is also restricted.

This is a great concept, especially within a healthcare environment. As you can properly control who has write/ update access, and involving a patient in would be simple - so long as you include them at the bottom of the chain. However I believe that there should always be a way to audit who looks at patient files - including a way to sign off on those updating, such as through a signature.

Within this paper they discuss the implementation of the Enhanced Bell-LaPadula method using 'Hyperledger' - this is an open source software that hosts Enterprise-level blockchain projects. This uses JSON logs to allocate access levels. This also has examples of smart contracts and how they work. After looking at the software and the screenshots from the document, I believe that this could be a suitable proving ground for later development of a prototype, incorporating bits of the paper - mainly the extended Bell-LaPadula model.

2.4 Review: Scalability Challenges in Healthcare Blockchain System—A Systematic Review Mazlan et al.,2020

Scalability within blockchain software is, as it currently stands one of the biggest downsides of blockchain on a monumental scale. This paper by Mazlan et al.,2020 explores the issues surrounding scalability, breaking it down and offering solutions to our issues. There were five main challenges targeted in this paper. (Mazlan et Al. 2020)

- Block Size

This is in relation to the maximum capacity a block can store before it is rejected by the network. Mazlan et al.,2020 notes that the immediate issue with this could be the unprocessed patient data - as if we wish to store it on a blockchain we would need to work out a solution to store this data.

- High Volume

A large number of transactions occurring in real-time. With a large institution, especially with the increased stretch on the healthcare services due to COVID-19 - healthcare institutions are performing at capacity. As such, an overflow of information could take place, causing an issue for the data storage at said institutions.

- Transactions

When generating a new block in a blockchain, the new block needs to be validated by all nodes in the network. This, with the high amount of transactions can result in an incredibly slow and tedious system which will only get worse with each additional node, and with the increasing size of the block chain.

- Number of Nodes

Nodes are essential components of a blockchain system. Each node has a copy of the full blockchain. However with the increase of the number of nodes, the workload to verify new blocks will also increase. This will also produce multiple entries on the blockchain distributed ledger. All of these issues coupled together will decrease the performance of the system as the size of the blockchain continues to increase.

- Protocol

Each validator on a network must verify each transaction, at least with Ethereum this is the case. However protocol also refers to Security, storage validation and access of the blockchain. This is a more ethical issue - how does the patient consent to their information being accessed, especially with the size of some blockchains, it could take some time to get the necessary information.

Per the above image, you can see that this paper has broken the problems down into two main components: Storage Optimisation and Redesigning Blockchain.

Needless to say, the biggest issue seems to be the storage optimisation, as blockchains cannot store a large amount of information. One solution to this issue could be implementing a one time use link through the blockchain which would be able to access a database which

Challenges	Solutions
1. Block Size 2. High Volume of Data 3. Transactions 4. Number of Nodes	Storage Optimization
1. Protocol 2. Block Size	Redesigning Blockchain

Figure 3: Challenges Solutions

houses the information - only accessible through the one use link. This would be a great solution to the storage issue of blockchains, as they would act as a directory more than a data storage solution. However, you would then need to generate a new block each time you connect to the database and use the code, while also including a new link in the blockchain block.

This could be difficult to administrate, as would you have a blockchain per user? Or would you have one generic account, one for read access and one for write access? This is an issue which will have to be addressed to potentially solve the storage optimisation issues. As this solution would eliminate the need to store massive amounts of information on the blockchain, allowing for a quicker and more seem less blockchain navigation. However how would you audit who has connected to the blockchain? This would also need addressing if this project was to incorporate this principle.

3 Requirements and Analysis

The project will have requirements that it will need to hit. I have decided to make several different types of requirements which reflect the importance and the type of features that will need to be fulfilled; **Basic Requirements**, **Security Requirements** and **Usability Requirements**.

3.1 Basic Requirements

Below, in table 1 we can see that there is a table of very basic features. These features are the minimum of what is needed for a successful project and who should be allowed access to these features. This is not however delving into how secure we can make the website, nor how user friendly we can make it. This features purely on functionality. This table shall be refereed to when evaluating the system's usability and functionality.

Feature	Description	Patient	Receptionist/ Staff Member	Doctor/ Equivalent	Admin
Logging In	Allows the user to logon to the web interface	✓	✓	✓	✓
Create new accounts	Allows user to create new user accounts				✓
Read Access to Databases	Allows the user to view certain databases	✓	✓	✓	✓
Write Access to Database	Allows the user to write to certain databases			✓	✓
Assign User Privileges	Allows the user to assign read/write privileges to other users				✓
View Access/audit Log	Allows user to see who has viewed/made changes to a database	✓	✓	✓	✓
Add Information to Database	Allows user to add information to the database			✓	
Create new blockchains	Allows user to create new blockchains which will be linked to new tables				✓

Table 1: Basic Guidelines

3.2 Security Requirements

The Security requirements is where the project will come into its own. While, yes we should be looking at the overall design and ease of access and functionality - as a Cyber Security student, I want to lock down as much as possible. By narrowing the scope we reduce risks, this is what I have built my requirements upon, using practical knowledge from my time working in the industry and common sense to ensure that it is achievable without fully sacrificing ease of access. As this comes from security all around, I will not be focusing on any specific account.

Feature	Description	Priority
Prevent XSS in all entry fields	Rated the #3 in the OWASP top 10 most critical web application security risks. Injection can occur in input fields or even in the url itself.	H
Ensure that blockchain is immutable	Blockchains should remain immutable to ensure that the information inside of them does not change, breaking the hash value of all subsequent blocks	H
Ensure that other tables cannot be accessed via the admin page	Limiting the admin to certain tables is a necessity, as they could otherwise bypass the blockchain to access the tables.	H
Enforce 2 factor authentication all accounts	2 factor authentication (2fa) provides an additional layer of security when securing our accounts, this will allow for us to secure accounts with ease.	M
Ensure website is using HTTPS	HTTPS will ensure that all information provided by the site is encrypted, making it more difficult for malicious actors to sniff packets on the network.	H
Reset blockchain password and generate new block to store the password within once accessed	Allows for no one to retain the password past one login, user will also never see the password.	H
Include an audit system within the blockchain	Allows for the patients to see who has viewed their information and track actions performed by higher privilege users	H

Table 2: Security Requirements

3.3 Usability Guidelines

The audience for this application would be the population of a country. Currently within the United Kingdom, there is an ageing population which could potentially struggle with technology.

Feature	Description	Priority
Consistency and Standard	Ensure that throughout application, there is consistency with the theme and the look of the product. This will provide seamless experience for the user no matter their background.	H
Minimalistic & Aesthetic Design	By using a minimalistic design, ensuring there is not a large variety of options on screen will ensure that the user will remain engaged with the product, ensuring that they will not get confused as easily.	H
Help & Documentation	Providing Help & Documentation is essential for both the user and the designer. It will limit the number of interactions required with the user, as they will not have to contact the designer for help. Ensuring they can resolve issues on their own.	H
Help users Recognize, Diagnose, & Recover From Errors	By using clear, traditional and distinctive error messages, users will recognise an error and know what the solution to the problem could be.	H
User Control & Freedom	Clearly provide a way to exit every interaction 'Exit' or 'Back' buttons etc.	H
Use Basic Language & Conversation Language	All users may not know technical jargon, therefore the language used needs to be simple English in a conversational manner (natural mapping).	H

Table 3: Usability Guidelines

During the process of designing this project, a key factor which is required to be kept central is ease of use to those who may not be tech literate. Hence the reasoning for using a website opposed to an phone application.

All of the above requirements will be a sufficient guideline to analyse the project and be a good way to railroad it. I have attempted to use language that is not too specific - this is mainly to ensure that I am locked in to one specific goal. If the development does not go to plan, then I will require wiggle room to ensure that I can still adapt and move forward.

Irregardless, below is a use case diagram specifically for the 'patient', who's data would be in question. This includes the standard layout of how they would use the website. While it is supposed to be used as their link to their information, it is not to act as anything other than the portal to that.

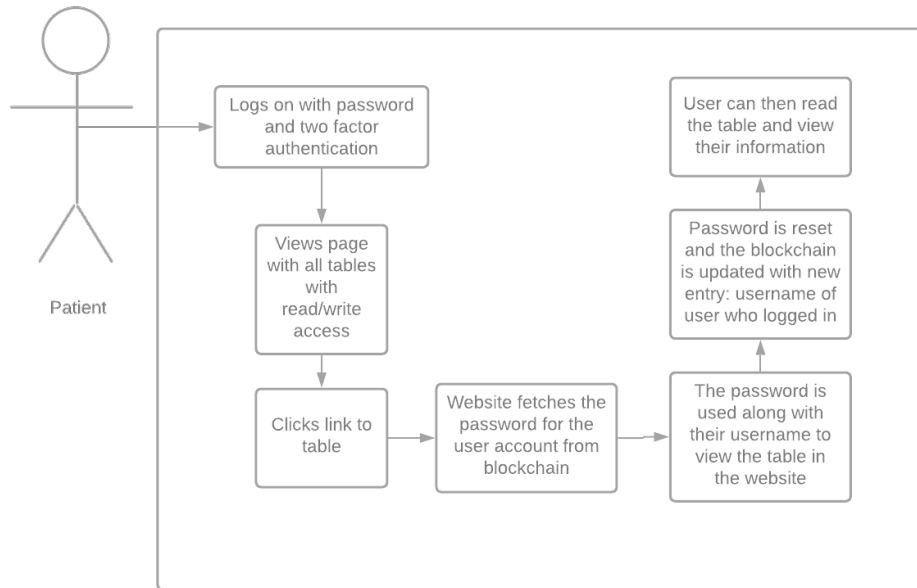


Figure 4: Blockchain - Use Case Diagram - Patient/user

4 Prototype Description

The prototype which I have developed accomplishes some of the basic requirements. Using XAMPP as my server solutions software. It's open source and lightweight - allowing for a high level of adaptability and customisability.

There is currently there is four pages; `index.php`, `login.php`, `welcome.php` and `results1.php`. Each of these have a role to play.

The first page that a user would view is "Index.php", this is effectively the homepage, with a link to the login page. In the future I wish to put a few articles on how the information is stored specifically targeted towards a patient who is tech illiterate. This may help them understand and put them at ease, promoting confidence in the solution. Additionally, an FAQ will be linked to this page with a walk through of how to use the site.

As seen, this is very basic and has the standard Latin placeholder text. On the Navigation bar, we can see "Blockchain for Healthcare" which links back to this page, and the "Login page" which leads us to the login page.

This page is linked to an MySQL database hosted on lamp, where the username and password of authorised accounts is stored. The username is stored with the password. The password is stored using "password_default" However this will be changing to a stronger encryption method in the future, potentially using Sha256.

Once the user has logged in, they are taken to the "Welcome page". This lets them know

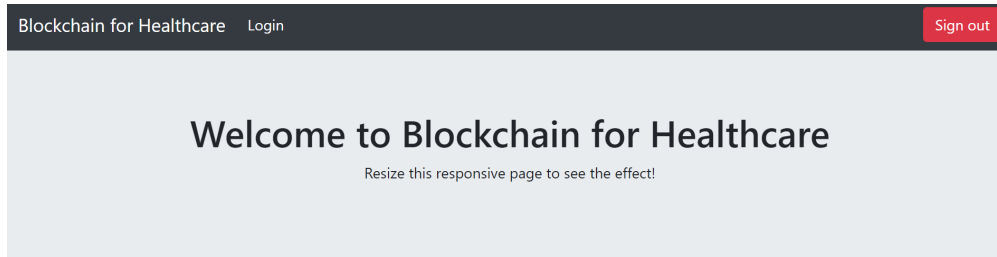


Figure 5: Alpha Landing page

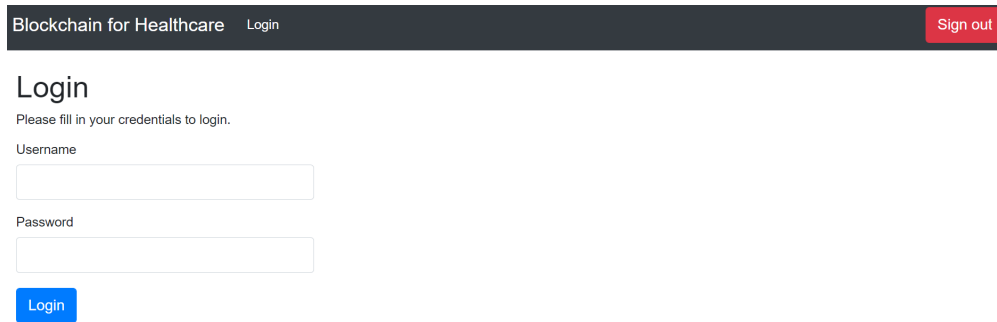


Figure 6: Alpha Login Page

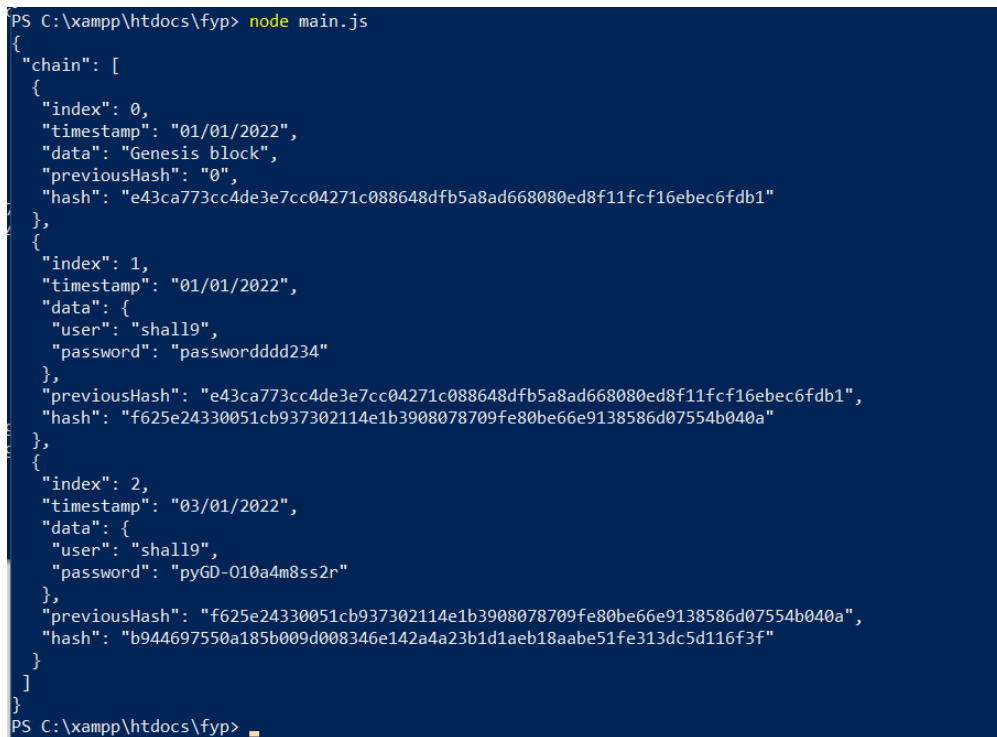


Figure 7: Alpha Blockchain example

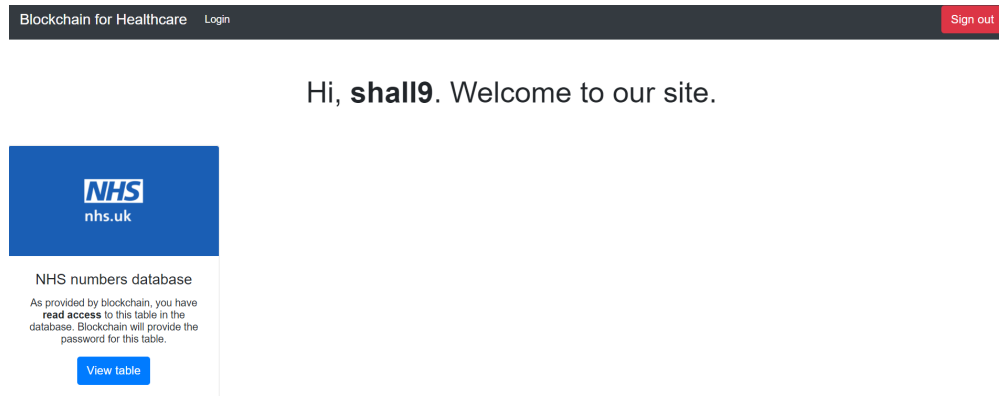


Figure 8: Alpha Welcome page

their username and it also shows what databases they have access to and what access level. For patient it will always be Read. As they should not be adding to nor editing their own information.

The "NHS Numbers Database" is an example data set I made which includes three entries. This can be overridden at any point if I were to find a large size fake data set. However either currently works.

The button takes the password from the last chain of a blockchain and uses that to login to the MySQL database which is currently setup. However, I have not managed to find a way to create a new addition to the blockchain that provides a new password and takes the user's username who access the database and adds that to the new block so there is immutable evidence that they viewed the database. Once a user clicks the "view table" they are taken to where their data is stored.

Blockchain for Healthcare		Login	Sign out
Hi, shall9, and welcome to the NHS numbers table.			
NHS Number	Forename	Surname	
18009853	Stephen	Hall	
45781245	Jaque	De Baugy	
16012541	Ginge	Hopper	

Figure 9: Alpha NHS Number table

This information is received from an MySQL server using the password provided by blockchain. There still needs to be a back button installed and for the formatting to look a bit prettier. However, while I am trying to reduce user input, I will allow the user to filter through their information with a basic drop down table search in the future to increase user accessibility.

5 Design, Implementation and Testing

5.1 Blockchain Design Overview

Previously discussed, earlier within this paper it was mentioned that blockchain has a limited amount of data which can currently be stored within it. This posed a problem when thinking about a pure blockchain system.

Due to this, the idea of blockchain databases was thrown around. The key to access a database would be stored within a blockchain, where an authorised user would then be able to pull this key from. Then, through the use of a smart contract - the username and timestamp of access for this key would be stored within the new block of the blockchain, revoking the access of the previous key, and then communicating with the database to generate a new key and then storing that on the blockchain for the next user.

While this mitigated the issue which blockchain represented in regards to the lack of information being able to be stored upon it - the issue then came of administrating it. How could you ensure that users could not cover their track once they had initially logged into the database?

If, for example they were given access to a database with full admin rights - they could effectively cover their tracks by deleting any logs which are stored upon the database. This is why it seemed necessary to use a third part when interacting with the database which would limit their access and also provide a platform to have an offshoot for event logs to for example a Security Information and Event Management system.

Therefore to recap - it is important to understand the current limitations of blockchain and understanding how much limited data can be stored upon the system. Hence why the need for having a data base keys stored within the blockchain

- Blockchain

With the database keys stored within this, along with all access logs stored within this.

- Interface

With the interface acting as the primary user interface between these two systems

- Database

Using this as the back end interface to store all of the information, with access only being allowed once the key has been taken from the blockchain, passed through the interface to interact with the database and make changes based upon their access levels as dictated within the database.

Within this blockchain-based-database - the most crucial part is the interface, which is linking together the two parts together and making a seamless solution to ensure that no

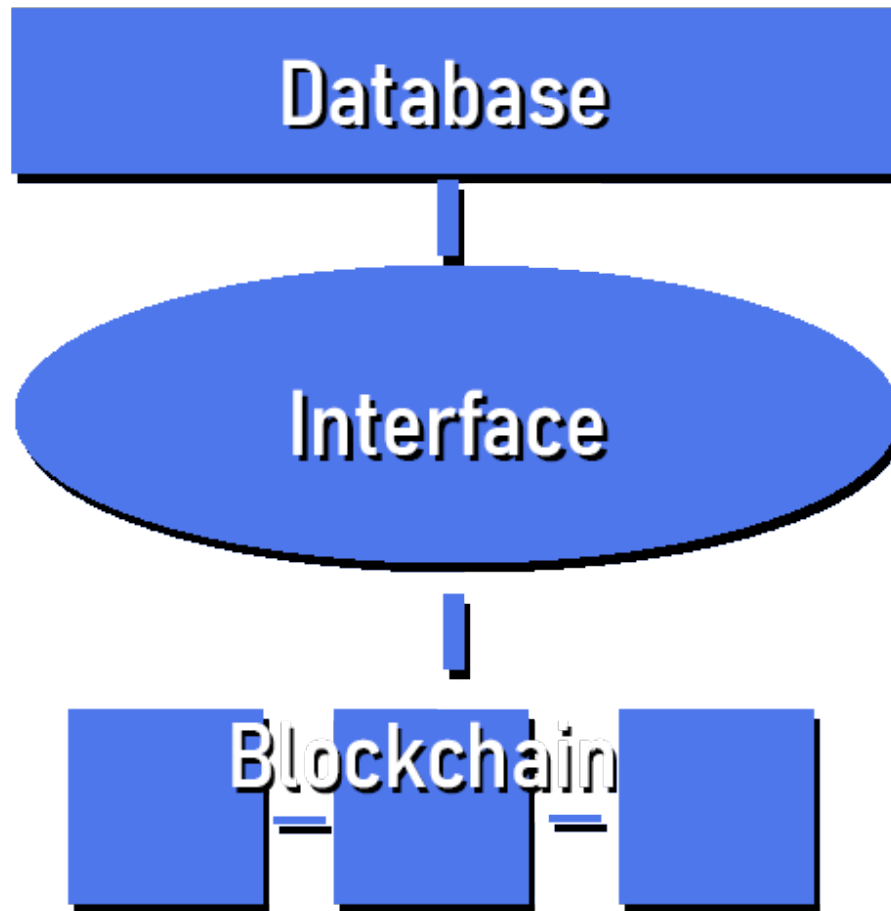


Figure 10: Blockchain-database example

misuse of privilege occurs. However this is one of the uses of a audit log which is outputted to a SIEM.

5.1.1 SIEM Implementation

While a SIEM may seem unnecessary as apart of this solution - it is a crucial part of any cyber security solution. Ensuring that, even if an threat actor breaches the interface - without the key from the blockchain they cannot access the back end. Even if the blockchain is read from - the audit system would capture the time of this access and send it to the SIEM for a Security Operations Centre Analyst would then investigate the threat if it has occurred outside of intended hours, or if a ruleset has been hit.

Within the prototype, there has not been plans included for a SIEM, however if this project would be taken to the next level, with it being developed into a full fledged NHS system, then NHS digital would ensure that the logs from the database, interface and blockchain

would feed into a SIEM or potentially SOAR (Security Orchestration, Automation and Response) solution. The connection of the logs would only take a bucket to be created and linked into the SIEM - this is something which phpmyadmin, the SQL database chosen can support.

5.2 User Interface Design

As showed within table 1, 2 and 3 it was illustrated exactly was needed to bring this this blockchain software into fruition. The basic features were important, however the usability guidelines and security features were equally important.

Blockchain offers a lot of useful security features due to the immutable feature which is built into all block chains. However for this, as a Cyber Security study, the focus should be more on how blockchain can ensure that information can stay the safe.

However, as this is also a product which has a target market of a wide range of audiences - the usability of the software must also be easy to use and follow a set of guidelines as stated within table 3. Within the development stage, all of these are very important - however it is also important to note that this is just the foundation of a project - there were a lot of changes throughout the course of the development, such as breaking apart multiple pages to ensure that it was not cluttered and to make it easier for the user to navigate. Ensuring that it is clear - using language which would be 'talkative' English, not using professional words. Additionally, there should be an intuitive design and error messages which will indicate exactly what the user did wrong.

Depending on the level of the user/ privilege level, they will be able to access different things - as seen within table 1. In total however, there should be 15 pages, with the majority of these being assigned to the admin and the doctor role. However, no one user will have access to all 15 pages at any one point unless they are a developer - which is the role which has currently been granted to the admin for ease of access.

The framework of the website has not changed much since the initial prototype - however the content has quite a fair bit. This was mainly because of the development not going to plan - however that was due to a lack of knowledge about the programming languages used by the author. As all project should be, this was a learning experience to see how far the knowledge and skills gained throughout the course of university stacks up with against a monolithic project such as this.

5.2.1 Index

The Index page is the first thing a user will see. Needless to say, information of the latest NHS articles of interest were included to give it more substance and to give it a sense of authenticity. However, noted within the header is mention of this as a demonstration of

what blockchain could accomplish with a fully fleshed development team with the budget behind it to make it a reality.

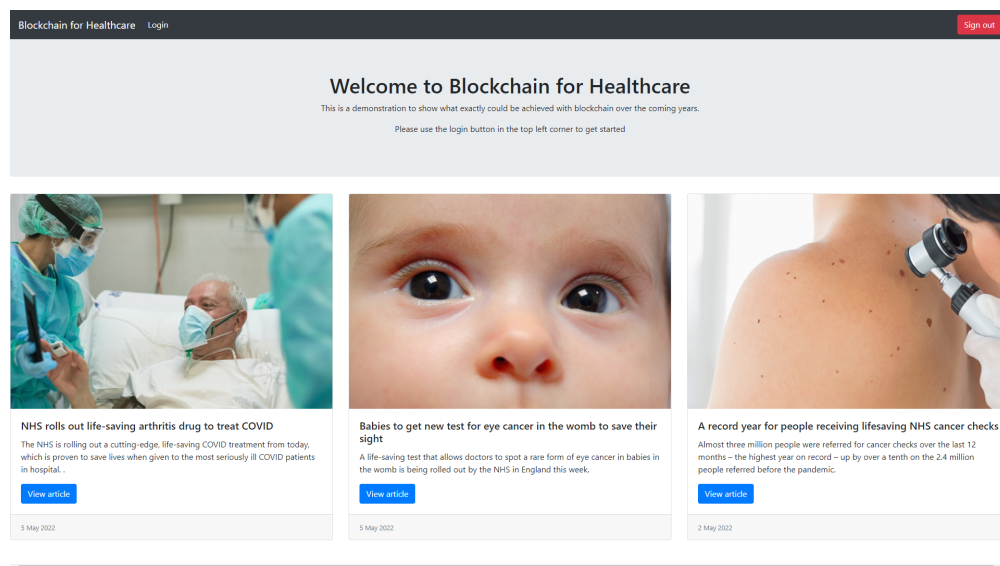


Figure 11: Index Page

Within the page, the user can do one of two things - if they are not logged in they can click the login button on the navigational bar to take them to the login page. Other than this, if the user is indeed logged in - they can click the login button to take them to the welcome page, which is where the majority of information will be kept for an end user. This will include links to the databases they can access once they access a blockchain.

This page is to be thought of as the external homepage of the site. As this project is intended to be internet facing - the users will be able to instantly see what they need to do by reading the paragraph below the title.

In regards to the login button not being more obvious, to keep in line with a theme and to ensure professionalism, it was chosen not to paint it a different colour than the other buttons. It was critical to do this with the sign out button, however that was to ensure that the user always knew where to click to exit.

5.2.2 Login

This is a fairly standard login page, as seen below in figure 12, there is a username and password box - along with the ability to create an account if a user does not have one set up already.

This web page will throw errors if anything other than a valid username is chosen, the password will also then automatically error out.

Figure 12: Login Page

5.2.3 Register

The register page is slightly more interesting than the login page - mainly as it has more fields and features built into this.

As with the login page, there is the Username field, and then two password fields to ensure you are using the right password. However, when this is running, there's several error checks running in the background, firstly to ensure you do not make an account with the same username as one currently used, and the username field also only accepts letters, numbers and underscores - this is to help prevent against xss, cross site scripting.

It is worth noting as well, that the passwords work similar, to ensure that they have the same input and to ensure that this is not malicious.

Figure 13: Register Page

When the account has completed all checks, it assigns it the level '0' - this user level has zero permissions and is reliant on an admin granting it patient rights - basically changing the role of the user to that of a patient through the admin panel. This ensures that a user can still be made, choosing your own password - which is stored with a SHA256 cryptographic hash function to ensure that, even if the database back end is breached, the malicious actor cannot steal the stored passwords.

5.2.4 Welcome

This is the internal homepage for all users apart from the admins. From here, Doctors, Staff and Patients can view whatever database access they currently have - once they click

the view table button of which ever database they have been granted access to - or this is how the technology is expected to perform - more on that later.

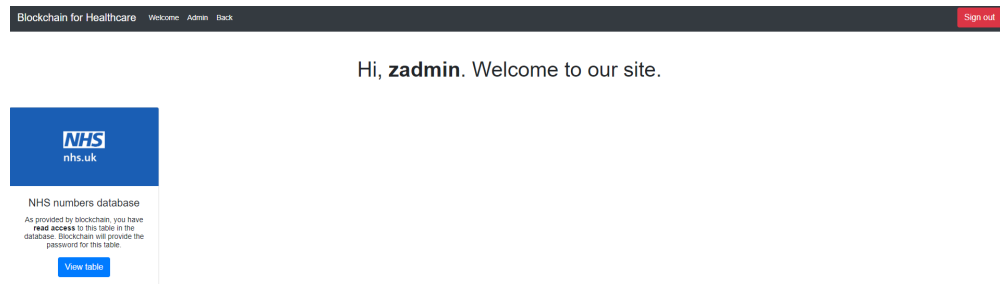


Figure 14: Welcome Page

However, the navigational tab has changed slightly to include an "welcome" "admin" and "back" tab. These navigate to the last page which was visited before visiting the current page.

5.2.5 Results1

The results page is an example of what information could be displayed when the access key to the blockchain is used. There is really nothing much to this, as it is just a database query which displays the table which the blockchain key is linked to. Without this key, the page would error and not show the database queried. Ideally this page would only be a read page for all users with access to this blockchain database - with the doctors only being able to add information to this through the admin dashboard/panel.

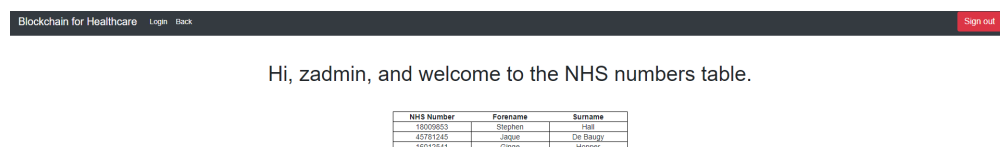


Figure 15: Results page 1

Each time this page, or any other blockchain enabled database page is visited, the access log is added to the audit log. This is to show patients who has been accessing their data. Patients can then access the audit log for this through this page - and there is a separate database table for each blockchain enabled database.

5.2.6 Admin Panel

The admin page used to be a page which used to be self contained. However due to an issue which occurred during development, which turned out to be a blessing in disguise - as it allowed the development of multiple pages instead of having everything crammed on one page. While it may of been more convenient for the end user to include all the information in

one place, by breaking up the features into one per page - it allows the user to know exactly where to navigate to, to make any changes needed.

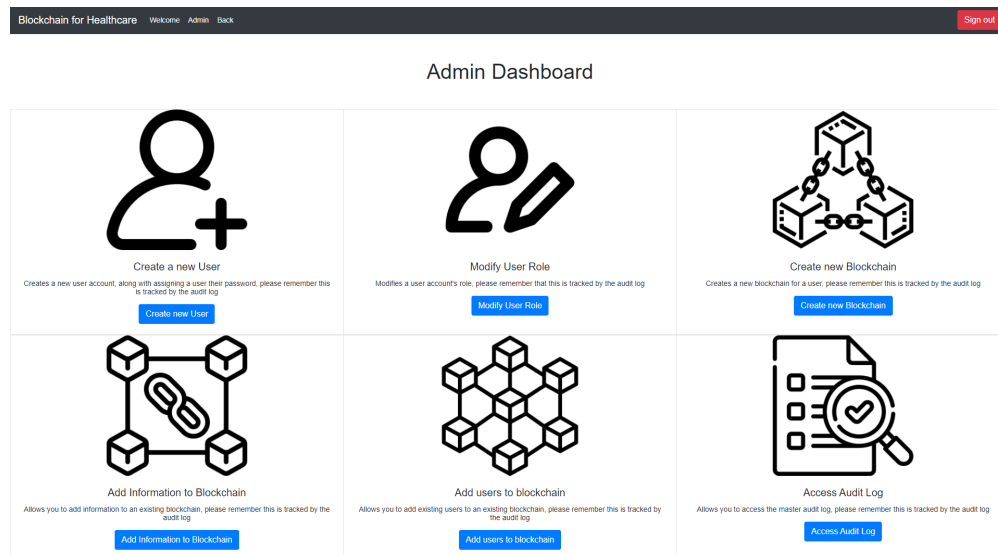


Figure 16: Admin Panel

Currently, within the developer view we can see that there are six option:

- Create a User
- Modify User Role
- Create new Blockchain
- Add Information to Blockchain
- Add Users to Blockchain
- Access Audit Log

Referring back to table 1 - we can see where these features all fit in, and which users would have access to what.

The design for this page is very similar to all other pages which give access to other main pages. Using a big blue button which clearly states where it leads which will easily catch the attention of the user before they read any other part of this page.

As one of the guidelines for this document mentions minimalist design, this is why functions have been split over multiple pages. Less is more in most cases, as a page which is not designed to be aesthetically pleasing, and is overloaded with information is not going to be user friendly. While this page will not be available to the public, and will be exclusively used by potentially NHS digital along with healthcare professionals - a system still needs to be intuitive and user friendly.

5.2.7 Admin - Audit Log

The audit log included a huge amount of information which was outputted from the audit log database seen within the MySQL web based-application.

This audit log page has a lot of information on it. There are a few features which could be included to make it more functioning and user friendly - whether it was to add a search function, or limit the audit log to be linked with a database and nothing else - having this audit log, log specifically a user creation/modification/deletion actions. However, currently in this state it is displaying all information within the audit log database.

Blockchain for Healthcare Welcome Admin Back Sign out				
Audit Log				
id	Timestamp	Username	action	log
14	2022-05-05 15:07:48	zadmin	updateUserRoles	reset to 4
15	2022-05-05 15:14:23	zadmin	createdUser	Created reset11
16	2022-05-05 15:15:00	zadmin	createdUser	Created reset11
17	2022-05-05 15:16:38	zadmin	updateUserRoles	reset to 4
18	2022-05-05 15:28:40	zadmin	accessedAuditLog	zadmin Access the Audit Log
19	2022-05-05 15:28:56	zadmin	accessedAuditLog	zadmin Access the Audit Log
20	2022-05-05 15:33:16	zadmin	accessedAuditLog	zadmin Access the Audit Log
21	2022-05-05 15:33:20	zadmin	accessedAuditLog	zadmin Access the Audit Log
22	2022-05-05 15:34:08	zadmin	accessedAuditLog	zadmin Access the Audit Log
23	2022-05-05 15:34:09	zadmin	accessedAuditLog	zadmin Access the Audit Log
24	2022-05-05 18:07:39	zdoctor	loggedin	zdoctor logged in
25	2022-05-05 18:01:48	zdoctor	accessedAuditLog	zdoctor Access the Audit Log
26	2022-05-05 18:00:35	zdoctor	accessedAuditLog	zdoctor Access the Audit Log
27	2022-05-05 18:07:59	zdoctor	accessedAuditLog	zdoctor Access the Audit Log
28	2022-05-05 18:08:11	zdoctor	accessedAuditLog	zdoctor Access the Audit Log
29	2022-05-05 18:08:16	zdoctor	accessedAuditLog	zdoctor Access the Audit Log
30	2022-05-05 18:08:31	zdoctor	accessedAuditLog	zdoctor Access the Audit Log
31	2022-05-05 18:14:28	zdoctor	accessedAuditLog	zdoctor Access the Audit Log
32	2022-05-05 18:15:30	zdoctor	accessedAuditLog	zdoctor Access the Audit Log
33	2022-05-05 18:15:40	zdoctor	accessedAuditLog	zdoctor Access the Audit Log
34	2022-05-05 18:26:30	zadmin	loggedin	zadmin logged in
35	2022-05-05 18:31:37	zadmin	accessedAuditLog	zadmin Access the Audit Log

Figure 17: Audit Log

As you can see, no Personally Identifiable Information stored within the audit log. This is to ensure that GDPR is maintained throughout the operation. However that does also mean that the only thing that can match up actions and events on the blockchain is the time frame. This is not always reliable when you have multiple doctors working on the same patient - this could result in confusion of which healthcare professional inputted the data specified by the log.

5.2.8 Admin - Information to Blockchain

The initial idea for the blockchain would be for the keys to be accessed via the interface which would only allow specific users to access blockchains. However, due to the limitations that have been faced during the project with blockchains - this was not possible. This shall be discussed further in the results and discussion section.

This will be applied for all relevant blockchain pages - therefore those pages will be removed from this section and address them all here.

The pages in question were:

- Information to Blockchain

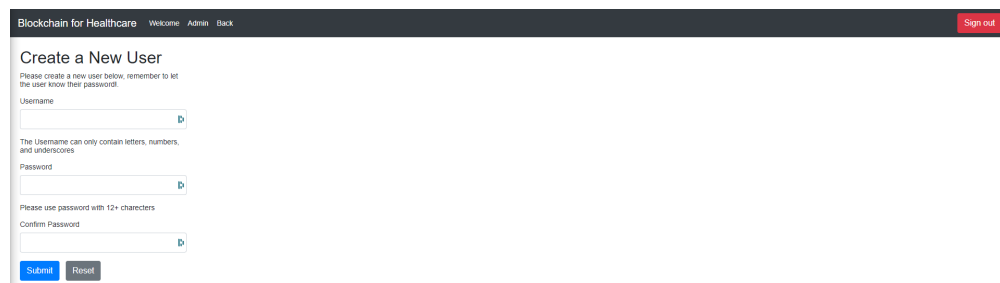
- User to Blockchain
- Create new blockchain

All of these pages relied upon interaction with a working blockchain model. Within these pages, an admin would be able to create additional blockchains and assign users of any level to these blockchains - but they would not be able to view the contents themselves - only the overhead admin log.

Each of these would have interfaces which would be accessible to those who have access - as specific within table 1.

5.2.9 Admin - New User

To create a new user, is very similar to signing up a new user - the only difference is the admin sets the password for the account and also has their name stamped in with the creation of this account which is then inputted into the audit log.



The screenshot shows a web interface for 'Blockchain for Healthcare'. At the top, there is a navigation bar with 'Welcome Admin Back' and a 'Sign out' button. The main heading is 'Create a New User'. Below this, a message states: 'Please create a new user below, remember to let the user know their password.' The form contains three input fields: 'Username', 'Password', and 'Confirm Password'. The 'Username' field has a tooltip that says 'The Username can only contain letters, numbers, and underscores'. The 'Password' field has a tooltip that says 'Please use password with 12+ characters'. At the bottom of the form are two buttons: 'Submit' and 'Reset'.

Figure 18: Blockchain-database example

The checks which is done on the fields is the exact same as the sign up part. Therefore it also prevents against XSS attacks. There is of course errors which get thrown if an admin input something wrong.

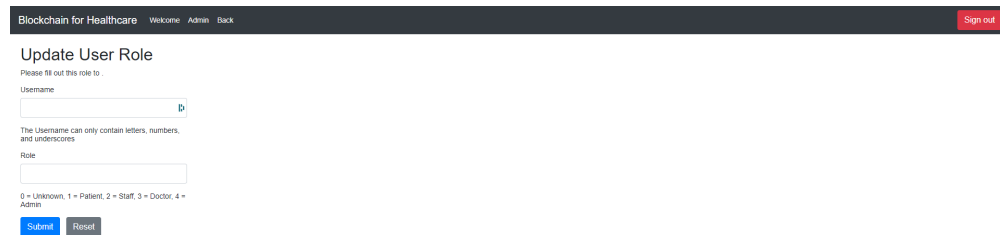
5.2.10 Admin - User Roles

User roles is very similar to creating a user, however this is updating the rule which is assigned to the user:

- 4 = Admin
- 3 = Doctor
- 2 = Staff
- 1 = Patient
- 0 = Unknown

As mentioned within table 1 you can see the permissions which should be associated with each role. However, when a user account is made, by an admin or be a user signing themselves up for one - they are granted the level 0 as default - this forces an admin to intervene and set a role level. Or access level if you will.

There are the usual catches to stop users from inputting XSS, however the input boxes for the role also does not accept anything but an integer between 0-4. If anything else is thrown - even an XSS attempt, it will reject the user input outright and give clear instructions on where the user could of gone wrong. This satisfies the usability requirements for this page,



The screenshot shows a web application interface for 'Blockchain for Healthcare'. At the top, there is a dark navigation bar with links for 'Welcome', 'Admin', and 'Back', and a 'Sign out' button on the right. The main content area is titled 'Update User Role' and includes the instruction 'Please fill out this role to:'. Below this, there are two input fields: 'Username' and 'Role'. The 'Username' field has a small blue icon to its right. Below the 'Role' field, there is a legend: '0 = Unknown, 1 = Patient, 2 = Staff, 3 = Doctor, 4 = Admin'. At the bottom of the form, there are two buttons: 'Submit' and 'Reset'.

Figure 19: Blockchain-database example

5.3 Testing

On the website designed, there are a fair number of functions/ processes which need to be tested to ensure they work as designed.

To test the website, we tested each button and function - throwing in XSS on user input fields in attempt to outsmart the regex installed within. Each page did indeed have roughly the same components, however when designed, there was not a master page, which made it necessary to test everything first.

Id	Page	Test Input	Expected Results	Actual Results	Pass/Fail
1	Index	Click Login	Go to Login Page	Goes to Login Page	P
2	Index	Clicks sign Out	Signs the user out of their session cookies	Signs the user out of their session cookies	P
3	Index	Click 1st view article button	Go to external website with the NHS article on it	Goes to external website with NHS article on it	P
4	Index	Click 2nd view article button	Go to external website with the NHS article on it	Goes to external website with NHS article on it	P
5	Index	Click 3rd view article button	Go to external website with the NHS article on it	Goes to external website with NHS article on it	P
6	Index	Click Blockchain for healthcare button	Takes user to the index page	Takes user to the index page	P
7	Login	Input invalid username	Errors saying the username or password is wrong	Errors saying the username or password is wrong	P
8	Login	Input valid username, but wrong password	Errors saying the username or password is wrong	Errors saying the username or password is wrong	P
9	Login	Input valid username and password	Logs in and takes user to the welcome page	Logs in and takes user to the welcome page	P
10	Login	Input xss attempt in username field	Executes xss and prints an alarm to screen	Errors saying the username or password is wrong	P
11	Login	Input xss attempt in username field	Executes xss and prints an alarm to screen	Errors saying the username or password is wrong	P

Id	Page	Test Input	Expected Results	Actual Results	Pass/Fail
12	Login	Input xss attempt into password field	Executes xss and prints an alarm to screen	Errors saying the username or password is wrong	P
13	Login	Click Login	Go to Login Page	Goes to Login Page	P
14	Login	Clicks sign Out	Signs the user out of their session cookies	Signs the user out of their session cookies	P
15	Login	Click Blockchain for healthcare button	Takes user to the index page	Takes user to the index page	P
16	Login	Click Back button	Takes user back to previous page	Takes user back to previous page	p
17	Register	Click Login	Go to Login Page	Goes to Login Page	P
18	Register	Clicks sign Out	Signs the user out of their session cookies	Signs the user out of their session cookies	P
19	Register	Click Blockchain for healthcare button	Takes user to the index page	Takes user to the index page	P
20	Register	Click Back button	Takes user back to previous page	Takes user back to previous page	P
21	Register	Input valid username	Errors, password must have atleast 12+ characters	Errors, password must have atleast 12+ characters	P
22	Register	Input valid username and password, leaves confirm password blank	Errors, confirm password	Errors, confirm password	P

Id	Page	Test Input	Expected Results	Actual Results	Pass/Fail
23	Register	Input valid username and password, confirm password not the same input	Errors, password does not match	Errors, password does not match	P
24	Register	Inputs already taken username with valid password and confirm password	Errors, username already taken	Errors, username already taken	P
25	Register	Inputs valid username, password, and confirm password	Successfully creates user and takes them to the Login page and an entry is made to the audit log	Successfully creates user and takes them to the Login page and an entry is made to the audit log	P
26	Welcome	Click Welcome	Goes to Welcome page	Goes to Welcome page	P
27	Welcome	Clicks sign Out	Signs the user out of their session cookies	Signs the user out of their session cookies	P
28	Welcome	Click Blockchain for healthcare button	Takes user to the index page	Takes user to the index page	P
29	Welcome	Click Back button	Takes user back to previous page	Takes user back to previous page	P
30	Welcome	Click Admin button	Takes user to admin page	Takes user to admin page	P
31	Welcome	Click view table	Takes user to results1	Takes user to results1	P
32	Results 1	Click Welcome	Goes to Welcome page	Goes to Welcome page	P
33	Results 1	Clicks sign Out	Signs the user out of their session cookies	Signs the user out of their session cookies	P

Id	Page	Test Input	Expected Results	Actual Results	Pass/Fail
34	Results 1	Click Blockchain for healthcare button	Takes user to the index page	Takes user to the index page	P
35	Results 1	Click Back button	Takes user back to previous page	Takes user back to previous page	P
36	Admin Panel	Click Welcome	Goes to Welcome page	Goes to Welcome page	P
37	Admin Panel	Clicks sign Out	Signs the user out of their session cookies	Signs the user out of their session cookies	P
38	Admin Panel	Click Blockchain for healthcare button	Takes user to the index page	Takes user to the index page	P
39	Admin Panel	Click Back button	Takes user back to previous page	Takes user back to previous page	P
40	Admin Panel	Click Admin button	Takes user to admin page	Takes user to admin page	P
41	Admin Panel	Click Create New User button	Takes user to adminNewUser page	Takes user to adminNewUser page	P
42	Admin Panel	Click Add information to Blockchain button	Takes user to adminInformationToBlockchain page	Takes user to adminInformationToBlockchain page	P
43	Admin Panel	Click Add Users to Blockchain button	Takes user to adminUsersToBlockchain page	Takes user to adminUsersToBlockchain page	P
44	Admin Panel	Click Access Audit Log button	Takes user to adminAuditLog page	Takes user to adminAuditLog page	P

Id	Page	Test Input	Expected Results	Actual Results	Pass/Fail
45	Admin New User	Click Welcome	Goes to Welcome page	Goes to Welcome page	P
46	Admin New User	Clicks sign Out	Signs the user out of their session cookies	Signs the user out of their session cookies	P
47	Admin New User	Click Blockchain for healthcare button	Takes user to the index page	Takes user to the index page	P
48	Admin New User	Click Back button	Takes user back to previous page	Takes user back to previous page	P
49	Admin New User	Click Admin button	Takes user to admin page	Takes user to admin page	P
50	Admin New User	Input valid username	Errors, password must have atleast 12+ characters	Errors, password must have atleast 12+ characters	P
51	Admin New User	Input valid username and password, leaves confirm password blank	Errors, confirm password	Errors, confirm password	P
52	Admin New User	Input valid username and password, confirm password not the same input	Errors, password does not match	Errors, password does not match	P
53	Admin New User	Inputs already taken username with valid password and confirm password	Errors, username already taken	Errors, username already taken	P

Id	Page	Test Input	Expected Results	Actual Results	Pass/Fail
54	Admin New User	Inputs valid username, password, and confirm password	Successfully creates user and takes them to the Login page and an input is made to the audit log	Successfully creates user and takes them to the Login page and an input is made to the audit log	P
55	Admin User Rolls	Click Welcome	Goes to Welcome page	Goes to Welcome page	P
56	Admin User Rolls	Clicks sign Out	Signs the user out of their session cookies	Signs the user out of their session cookies	P
57	Admin User Rolls	Click Blockchain for healthcare button	Takes user to the index page	Takes user to the index page	P
58	Admin User Rolls	Click Back button	Takes user back to previous page	Takes user back to previous page	P
59	Admin User Rolls	Click Admin button	Takes user to admin page	Takes user to admin page	P
60	Admin User Rolls	Input invalid username	Errors, This username does not exist & Please enter a role	Errors, This username does not exist & Please enter a role	P
61	Admin User Rolls	Existing Username inputted	Errors - Please enter a role.	Errors - Please enter a role.	P
62	Admin User Rolls	Existing Username inputted, inputted a role >4	Errors: Role must be 0, 1, 2, 3 or 4	Errors: Role must be 0, 1, 2, 3 or 4	P
63	Admin User Rolls	Existing Username inputted and a string	Unable to input a string due to the box being numbers only	Unable to input a string due to the box being numbers only	P

Id	Page	Test Input	Expected Results	Actual Results	Pass/Fail
64	Admin User Rolls	Existing Username inputted and a number between 0-4	The SQL statement is executed, the user is updated to the role level, an addition to the audit log was made and user is taken to the admin panel	The SQL statement is executed, the user is updated to the role level, an addition to the audit log was made and user is taken to the admin panel	P
65	Create New Blockchain	Click Welcome	Goes to Welcome page	Goes to Welcome page	P
66	Create New Blockchain	Clicks sign Out	Signs the user out of their session cookies	Signs the user out of their session cookies	P
67	Create New Blockchain	Click Blockchain for healthcare button	Takes user to the index page	Takes user to the index page	P
68	Create New Blockchain	Click Back button	Takes user back to previous page	Takes user back to previous page	P
69	Create New Blockchain	Click Admin button	Takes user to admin page	Takes user to admin page	P
70	Add Information to Blockchain	Click Welcome	Goes to Welcome page	Goes to Welcome page	P
71	Add Information to Blockchain	Clicks sign Out	Signs the user out of their session cookies	Signs the user out of their session cookies	P
72	Add Information to Blockchain	Click Blockchain for healthcare button	Takes user to the index page	Takes user to the index page	P

Id	Page	Test Input	Expected Results	Actual Results	Pass/Fail
73	Add Information to Blockchain	Click Back button	Takes user back to previous page	Takes user back to previous page	P
74	Add Information to Blockchain	Click Admin button	Takes user to admin page	Takes user to admin page	P
75	Add Users to Blockchain	Click Welcome	Goes to Welcome page	Goes to Welcome page	P
76	Add Users to Blockchain	Clicks sign Out	Signs the user out of their session cookies	Signs the user out of their session cookies	P
77	Add Users to Blockchain	Click Blockchain for healthcare button	Takes user to the index page	Takes user to the index page	P
78	Add Users to Blockchain	Click Back button	Takes user back to previous page	Takes user back to previous page	P
79	Add Users to Blockchain	Click Admin button	Takes user to admin page	Takes user to admin page	P
80	Audit Log	Click Welcome	Goes to Welcome page	Goes to Welcome page	P
81	Audit Log	Clicks sign Out	Signs the user out of their session cookies	Signs the user out of their session cookies	P
82	Audit Log	Click Blockchain for healthcare button	Takes user to the index page	Takes user to the index page	P
83	Audit Log	Click Back button	Takes user back to previous page	Takes user back to previous page	P
84	Audit Log	Click Admin button	Takes user to admin page	Takes user to admin page	P

Table 4: Feature Testing

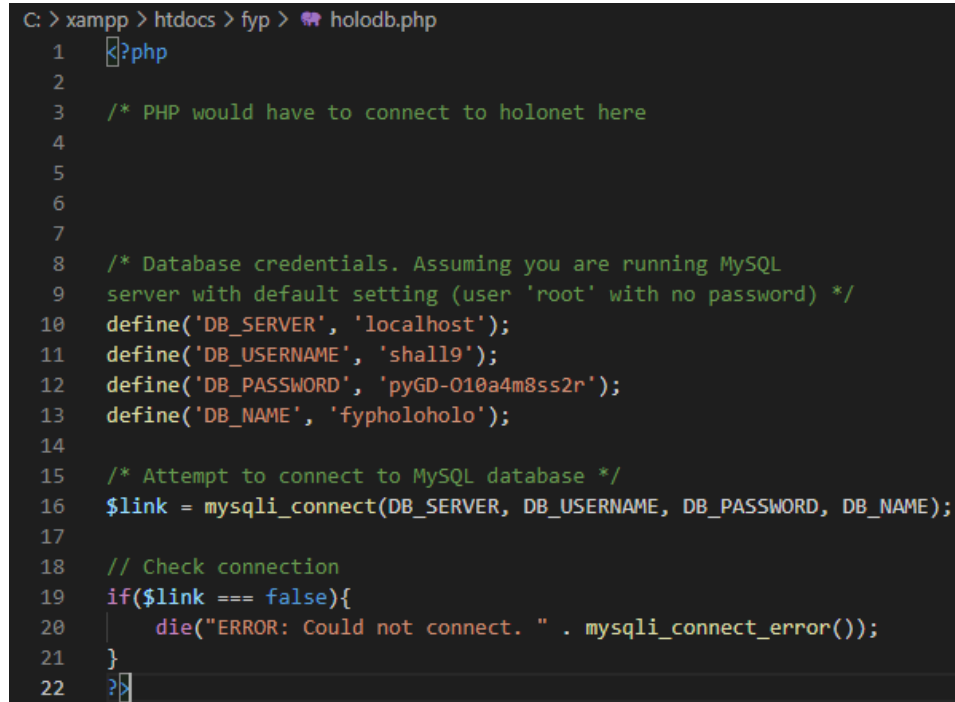
5.4 Testing... continued

As you can see, every function on the website has been tested, even testing those which the user may not notice and what happens in the background. All results were expected, as a lot of the time it was just code linked to various SQL queries or visiting different pages.

From the testing which has been performed, it can be confirmed that this website is at least safe from any 'script kiddie' attacks - these are attacks in the cyber security industry which are unintelligent copy and paste attacks - attempting to try them against multiple sites until something occurs and allows the attack through.

5.5 Implementation

The bulk of my application was made through PHP, HTML, JavaScript and CSS. This application was essentially the front end of the database that when navigating to the right pages, and using user input, would execute SQL commands to a phpmyadmin sql database. The main handler for all of this could be my config file.



```
C: > xampp > htdocs > fyp > holodb.php
1  <?php
2
3  /* PHP would have to connect to holonet here
4
5
6
7
8  /* Database credentials. Assuming you are running MySQL
9  server with default setting (user 'root' with no password) */
10 define('DB_SERVER', 'localhost');
11 define('DB_USERNAME', 'shall9');
12 define('DB_PASSWORD', 'pyGD-010a4m8ss2r');
13 define('DB_NAME', 'fypholoholo');
14
15 /* Attempt to connect to MySQL database */
16 $link = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);
17
18 // Check connection
19 if($link === false){
20     die("ERROR: Could not connect. " . mysqli_connect_error());
21 }
22 >
```

Figure 20: Database Handler/Config

As you can see, it is pretty simple commands, as it is just declaring what is needed to access the database for an account. In a perfect world, where the blockchain platform could perform, we would be able to pill the key out of the last block and have it reach out to the database via a smart contract for a new key. However as the author was never able to think of an effective way to incorporate this into the actual application, it is something which will remain as not included.

5.5.1 Blockchain Integration

As you can see above, this is the initial script for the blockchain. It is able to handle itself and create blocks perfectly fine - however the issue is having it interfaced with the web page.

Above is the blockchain code used, this does work and generates a blockchain, however

```

C:\xampp\htdocs> fyp > .\5 main.js > ...
1  const SHA256 = require('crypto-js/sha256');
2
3  class Block{
4    constructor(index, timestamp, data, previousHash = ''){
5      this.index = index;
6      this.timestamp = timestamp;
7      this.data = data;
8      this.previousHash = previousHash;
9      this.hash = this.calculateHash();
10   }
11
12   calculateHash(){
13     return SHA256(this.index + this.previousHash + this.timestamp + JSON.stringify(this.data)).toString();
14   }
15 }
16 class Blockchain{
17   constructor(){
18     this.chain = [this.createGenesisBlock()];
19   }
20   createGenesisBlock(){
21     return new Block(0, "01/01/2022", "Genesis block", "0");
22   }
23   getLatestBlock(){
24     return this.chain[this.chain.length - 1];
25   }
26
27   addBlock(newBlock){
28     newBlock.previousHash = this.getLatestBlock().hash;
29     newBlock.hash = newBlock.calculateHash();
30     this.chain.push(newBlock);
31   }
32   isChainValid(){
33     for(let i = 1; i < this.chain.length; i++){
34       const currentBlock = this.chain[i];
35       const previousBlock = this.chain[i - 1];
36       if(currentBlock.hash !== currentBlock.calculateHash()){
37         return false;
38       }
39       if(currentBlock.previousHash !== previousBlock.hash){
40         return false;
41       }
42     }
43     return true;
44   }
45 }
46 let testtest = new Blockchain();
47 testtest.addBlock(new Block(1, "01/01/2022", {user: "shall9", password: "passwordddd234" }));
48 testtest.addBlock(new Block(2, "03/01/2022", {user: "shall9", password: "pyG0-010a4m8ss2n" }));
49 console.log(JSON.stringify(testtest, null, 1));

```

Figure 21: Blockchain - Javascript

the only way to add to it would be to manually input blockchains within a script. The reason I chosen the method of attempting to create my own blockchain technology was due to all of the blockchains softwares being difficult to use, the documentation provided was not helpful and not a lot of information available online to assist with getting things to work. One such technology I attempted to use was holochain, but alas this did not function as expected.

Therefore this remains a figment of a what it could be, without blockchain implementation, three of the core features does not function - as we cannot create new blockchains, add users to the blockchains nor add information to the blockchains via the interface.

5.5.2 Audit Log

An audit log was on of the key functions which was necessary to ensure that the end user knows who has accessed their information. As you can see, there is 22. This shows the php code used to add information to the audit log table within the database.

```

62 <?php
63
64 // Include config file
65 require_once "config.php";
66
67
68 // Define variables and initialize with empty values
69 $username = $password = $confirm_password = "";
70 $username_err = $password_err = $confirm_password_err = "";
71
72     $action = "accessedAuditLog";
73     $auditsql = "INSERT INTO auditlog (username, action, log) VALUES (?, ?, ?)";
74     $sourceUser= $_SESSION["username"];
75     $log = "$sourceUser Access the Audit Log";
76
77     if($auditsmt = mysqli_prepare($link, $auditsql)){
78         // Bind variables to the prepared statement as parameters
79         mysqli_stmt_bind_param($auditsmt, "sss", $sourceUser, $action, $log);
80
81         // Attempt to execute the prepared statement
82         if(mysqli_stmt_execute($auditsmt)){
83             // Redirect to login page
84             echo "";
85         } else{
86             echo "Oops! Something went wrong. Please try again later.";
87         }
88
89         // Close statement
90         mysqli_stmt_close($auditsmt);
91     }
92 }
93 ?>

```

Figure 22: Audit-log Example

Currently, this code will execute when a user visits the page which the code is situated upon. As this specific iteration of the audit log code executes on accessing the audit log page. However, there are iterations of this attached with logging in, creating an account, changing a user role or signing up for an account which will automatically execute upon successful execution of statements on their page.

5.5.3 Interactions with the Database

Below you can see the code used to validate a username - this also prevents against SQL injection due to how the code is structured - this was tested during the testing phase of the project.

As you can see from figure 23, there are a few checks to ensure that the username field is not left blank, whether the user is conforming to the stipulations presented to them when creating a username, and whether the username exists to begin with.

By including the limitation field, it effectively prevents most forms of XSS, as XSS uses special characters such as HTML tags, or HTML code to execute attacks. As this has been limited to a-Z0-9_ - which accounts for characters from a-Z both uppercase and lowercase,

```

17 // Define variables and initialize with empty values
18 $username = $role = $confirm_role = "";
19 $username_err = $role_err = $confirm_role_err = "";
20
21 // Processing form data when form is submitted
22 if($_SERVER["REQUEST_METHOD"] == "POST"){
23
24     // Validate username
25     if(empty(trim($_POST["username"]))){
26         $username_err = "Please enter a username.";
27     } elseif(!preg_match('/^[a-zA-Z0-9_]+$/', trim($_POST["username"]))) {
28         $username_err = "Username can only contain letters, numbers, and underscores.";
29     } else{
30         // Prepare a select statement
31         $sql = "SELECT id FROM users WHERE username = ?";
32
33         if($stmt = mysqli_prepare($link, $sql)){
34             // Bind variables to the prepared statement as parameters
35             mysqli_stmt_bind_param($stmt, "s", $param_username);
36
37             // Set parameters
38             $param_username = trim($_POST["username"]);
39
40             // Attempt to execute the prepared statement
41             if(mysqli_stmt_execute($stmt)){
42                 /* store result */
43                 mysqli_stmt_store_result($stmt);
44
45                 if(mysqli_stmt_num_rows($stmt) == 0){
46                     $username_err = "This username does not exist.";
47                 } else{
48                     $username = trim($_POST["username"]);
49                 }
50             } else{
51                 echo "Oops! Something went wrong. Please try again later.";
52             }
53
54             // Close statement
55             mysqli_stmt_close($stmt);
56         }
57     }

```

Figure 23: Username Validation

numbers ranging 0 through 9 and allows the use of an underscore - the job of a malicious actor has become that much more difficult.

As seen above within figure 24, you can see that it is a lot less strict than the username examples - however one of the reasons for this is that the input box used only allows numbers, a special character or letter physically cannot be typed within it. This means that the only regex required is checking whether a role inputted ranges between 0-4.

```

59 // Validate role
60 if(empty(trim($_POST["role"]))) {
61     $role_err = "Please enter a role.";
62 } elseif(!preg_match('/^[01234]+$/', trim($_POST["role"]))) {
63     $role_err = "role must be 0, 1, 2, 3 or 4.";
64 } else {
65     $role = trim($_POST["role"]);
66 }
67

```

Figure 24: Role Validation

6 Results & Discussion

6.1 Changes Undergone

During the course of this project, there were a few changes which needed to be made. At one point, the author toyed with using a python based blockchain method - however there became the issue of intertwining with the web application to update the blockchain and become immutable, while still being able to add to it.

This became an issue, as the blockchain which was created each time was just a container which could be accessed - however the author could not work out how to allow for communication between both the web application and the blockchain. This included the JavaScript blockchain which was initially created - and then the attempted Python blockchain created later on.

It was then chosen that the project would come off the path of showing a working blockchain due to the difficulties surrounding the implementation. As it was difficult adding to a blockchain without updating a file manually.

This theoretically threw a spanner in the works of the project due to the reliance on the blockchain technologies. Throughout this project the author has had to conduct a lot of research into blockchain, and languages which they are unfamiliar with, which likely is one of the issues of blockchain being unable to be implemented. However the majority of the research conducted was to see how safe and trustworthy blockchain technology current was. As, this is a Cyber Security report, and blockchain is supposed to revolutionise the world through its trustworthiness and distributed ledger technologies.

6.2 Limitations & Effectiveness

Blockchain is far from being a reliable concept in its current form, and I am not just talking in this project. As we are reliant on pairing technologies with blockchain to make it function as we want. Ideally, a large amount of information should be able to be stored on a blockchain. However we would then face the computational power required to access and create this block, as well as a huge amount of storage which would be required once the

blockchain gets to a certain size.

However, through interlinking and mingling the technologies of blockchain and a database - we come up with an effective workaround. As the database storage can be expanded at will - so long as we store the keys to access the database within a blockchain, which are immutable, and while that was my aim for this project - unfortunately this has not been met.

In regards to all of my requirement goals, anything which mentions a blockchain has not been met found in figures 2 and 1. The usability guidelines, have all been met, which you can find in figure 3 as these do not concern blockchain and were all easily implemented from the start. However, for the security requirements - two will also go unanswered - as HTTPS requires an SSL certificate, which cost £60 a year - if this was to become a live project then this would be a drop in ocean for the amount of money behind it, however as the author of this report is a student this was feasible, as not only would we require an SSL certificate, but also a website to host this on. Additionally, two allow two factor logons from an account - the user would need access to an email account as well as a web server hosting the google authenticate integration.

Even though there were a few hiccups within the project, the author has researched the topic and executed a prototype which shows what blockchain could bring us in the not so distant future. All usability guidelines have been met and this application is straight forward to use and could be executed by anyone with a slight bit of technical know-how.

If this project would be executed again with the knowledge to this data, there are multiple ways which could improve the state of the project. Various features such as guying a web platform to host the website to ensure that it can be HTTPS ready - with a valid SSL certificate and with access to a domain which all accounts have an email linked to them to allow for two factor authentication. There is also the opportunity to salt and pepper all patient information inputted into the database, requiring the pepper from the blockchain and the salt from the application to ensure that the information cannot be viewed by an administrator in the back end.

6.3 Legal, Ethical, Social, and Professional Issues

There are several issues with blockchain and this project in its current form.

6.3.1 Legal Issues

As this database will contain PII (Personally Identifiable Information), then this project will need to subject itself to the General Data Protection Regulations (GDPR). This does complicate things slightly, but it then becomes a case of asking the patient if they want to share their information with an unknown doctor. However this is an issue which needs to be fully fleshed out - after careful consideration, the author believes that the patient having access to the audit log of who access their information should be sufficient to put their mind at ease. As if the patient can see whether there has been any unauthorised access, they can immediately tell due to the interface logging all access.

6.3.2 Ethical Issues

There are not many ethical issues surrounding this project - as the majority of the issues are concerned with data protection and to ensure that the patient data is only accessed by those with proper access. The mind of the patient is also put at ease as they can access an audit log to check who has access their data and when.

6.3.3 Social Issues

This is by far the most difficult part of the the LESPI principles. As to get people to use a software which handles their data you either need to give them confidence that this technology is secure, just like what blockchain claims to be. This is something which is rising due to crypto currencies being in the public eye as well as various governments starting to support E-Voting - in the next 5-10 years we will be using blockchain technologies everywhere.

6.3.4 Professional Issues

Ethics do not really come into this much, as it is a software handling personal data. The only thing which may have to be questioned is that doctors do not discuss the data of the patient outside of consultations or meetings. However this is something which doctors have been doing for years and will continue to do until the end of time.

So long as all guidelines are followed - and users are not given higher access then they require - there should be no issues with information being leaked.

7 Conclusion

After working on my entire Final Year Project, the apogee of my University experience - I have learned so many essential skills which I can bring to the table for my career within the field of Cyber Security. My research into blockchain has not only given me a thorough understanding of the subject and where in the next five years we could end up. Blockchain is an invaluable technology which will one day, be used in the grand majority of software and used within infrastructure everywhere - the possibilities are endless and exciting.

Through working on my project, I have learned many a soft and hard skills - unlike my other university project, this was not something which could be left to last minute. However I also believe that a combination of both my year in industry, working at ECSC Group Plc and the influence of my supervisor has pushed me to develop better time keeping skills and organisational skills. This year was a huge push for me, especially after Covid-19 messed the lives of students up - I fortunately soared.

ECSC Group plc, where I started my placement at kept me on during Covid - where other students in my year, who had secured placements were forced into their final year ill prepared. Out of the 129 of my cohort, I was one of the 9 accepted for placement. Due to Covid, I never was given the opportunity to do my job at ECSC - that being a "Cyber Security Researcher/ Analyst - Student Placement". Instead I was thrown in at the deep end, doing the job of a full analyst within a month of being there. Needless to say it was a sink or swim situation, with furlough coming into play and the lack of other analysts to learn from myself and two other students from Sheffield Hallam undertook a crash course to get us ready.

Due to this crash course and the subsequent responsibilities placed upon myself, I had to adapt pretty quickly forcing me to learn new skills, both hard and soft to ensure I passed my placement. One of these was timekeeping, which as I mentioned earlier I was required to show in this project. Which, arguably is the most important skill of this entire project.

Before this project, I had never done touched JavaScript coding - however throughout my placement I had become more comfortable with coding due to using BASH, the Bourne Again Shell - which is used on all Unix based systems. Due to this I was familiar with some of the functions and was able to more easily learn this language - as I had previously called JavaScript scripts while working with a bash Script. This is going to be incredibly useful throughout my cyber security career as this will help with automating tasks, understanding JavaScript based threats, and also some of the security limitations with it.

Naturally, with additional time I believe that this project could truly take off or form the foundation for a Healthcare Blockchain solution - however I am still proud with the work I had made with the prototype. This was a tough project for me as I was practising red team activities - which is the attacking side of cyber security, whereas my job and experience is blue team, which is defending against threats and analysing them. However with the my experience I am glad I was able to make a functioning blockchain linked database - allowing

for a secure authentication method, a straight forward design and evaluated the LESPI concerns surrounding this.

8 References

Binance Academy (2020) *Proof of Authority Explained* Bincance Academy <https://academy.binance.com/en/articles/proof-of-authority-explained> Accessed 30th Nov 2021

Blockholic (2020) *Part 1 - What is Blockchain?* [video] Blockholic <https://www.youtube.com/watch?v=T0hNDsRcrhA&list=PLkMOMH7Grb27r-AqydUYoTFGEQFSTv6rr> Accessed 28th Nov 2021

Bogdanov D. (2021) *Proof of Authority Explained* LimeChain <https://limechain.tech/blog/proof-of-authority-explained/> Accessed 30th Nov 2021

Iredale G. (2021) *Ultimate Guide To Pros And Cons Of Blockchain* 101 Blockchains <https://101blockchains.com/pros-and-cons-of-blockchain/> Accessed 24th Nov 2021

Kowalskiy A. (No date) *Blockchain and Cryptocurrency Explained: How Does It All Work?* MintyMint <https://mintymint.net/blog/tech/blockchain-cryptocurrency-explained-how-does-it-work/> Accessed 30th Nov 2021

Kumar, R., Tripathi, R. (2021) Scalable and secure access control policy for healthcare system using blockchain and enhanced Bell–LaPadula model J Ambient Intell Human Comput vol. 12 <https://link-springer-com.brad.idm.oclc.org/article/10.1007/s12652-020-02346-8#citeas> Accessed 7th Dec 2021

Mazlan A. A. , Mohd Daud S., Mohd Sam S., Abas H., Abdul Rasid S. Z. and Yusof M. F. (2020) Scalability Challenges in Healthcare Blockchain System—A Systematic Review *IEEE Access* vol. 8 <https://ieeexplore.ieee.org/document/8968381/authors#authors> Accessed 6th Dec 2021

Peyrott S. (2017) *An Introduction to Ethereum and Smart Contracts: a Programmable Blockchain* auth0 <https://auth0.com/blog/an-introduction-to-ethereum-and-smart-contracts-part-2/> Accessed 29th Nov 2021

Rosic A. (2020) *Smart Contracts: The Blockchain Technology That Will Replace Lawyers* Blockgeeks <https://blockgeeks.com/guides/smart-contracts/> Accessed 7th Dec 2021

Simply Explained (2017) *How does a blockchain work - Simply Explained* [Video] Simply Explained https://www.youtube.com/watch?v=SSo_EIwHSd4 Accessed 24th Nov 2021