

Blockchain-based Secure Medical Record Sharing System

BY

TIAN XIANG YANG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMMUNICATION AND NETWORKING

Faculty of Information and Communication Technology

(Perak Campus)

January 2019

REPORT STATUS DECLARATION FORM

Title: _____

Academic Session: _____

I _____
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor's signature)

Address:

Supervisor's name

Date: _____

Date: _____

Blockchain-based Secure Medical Record Sharing System

By

Tian Xiang Yang

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMMUNICATION AND NETWORKING

Faculty of Information and Communication Technology

(Kampar Campus)

January 2019

UNIVERSITI TUNKU ABDUL RAHMAN
DECLARATION OF ORIGINALITY

I declare that this report entitled “**Blockchain-based Secure Medical Record Sharing System**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : _____

Date : _____

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Dr. Lee Wai Kong who has given me this bright opportunity to engage in an Blockchain development project. It is my first step to establish a career in Blockchain field. A million thanks to you. Finally, I must say thanks to my parents and my family for their love, support and continuous encouragement throughout the course.

ABSTRACT

Database storage is a traditional way to keep all the data on a central database server. It may keep data expose in integrity issue and accuracy problem especially for the medical sector. Blockchain as a popular topic nowadays, it is able to ensure data security while using the distributed topology. The data is also protected by only allowed the permissioned personnel whom granted access by Blockchain.

The main issue of medical institutions is the synchronization of patients' data. Each medical institutions have their own database, but the patients may not go to a specific clinic every time. Therefore, it is difficult to get the updated record of patient in order to provide the diagnosis accurately. The another problem is the research institution difficult to obtain the medical record for research purpose from medical institution due to the privacy issue.

In this project, the framework of Blockchain 3.0 Hyperledger Fabric will be implemented to solve the problem stated above. The Hyperledger Composer will be used to setup the development environment, running environment and playground for running the Blockchain. Besides that, Channel and Chaincode will be used in this system.

At the end of project, a system will be designed for the usage of medical and research institution. Patients are able to use mobile application to view their medical records. In addition, the medical records are able to update by medical institution and view by permissioned research institution through web application.

CONTENTS

CONTENTS.....	6
LIST OF FIGURES.....	9
Chapter 1: Project Background	1
1.1 Project Motivation and Problem Statement.....	1
1.2 Project Scope	2
1.3 Project Objectives	3
1.4 Impact, Significance and Contribution.....	4
1.5 Proposed Approach	5
1.6 Background Information	6
1.7 Report Organization.....	8
Chapter 2 : Literature Review	9
2.1 MeDShare:Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain	9
2.2 Blockchain Technology for Improving Clinical Research Quality	10
2.3 Healthcare Data Gateways: Found Healthcare Intelligence on Blockchain with Novel Privacy Risk Control.....	11
2.4 A Secure System for Pervasive Social Network-Based Healthcare	12
2.5 “MedRec” prototype for electronic health records and medical research data	13
Chapter 3 : System Methodology	14
3.1 System Development Model	14
3.2 System Requirement and Technology Involved	15
3.2.1 Hardware involved:.....	15
3.2.2 Software involved:	17
3.2.3 Programming Language Involved:	20
3.2.4 Other components involved:	22
3.3 Functional Requirement	24
3.3.1 System role and actions	24
3.3.2 Update medical records to Blockchain	24
3.3.3 Retrieve medical records to database server	24
3.3.4 Design Chaincode to allow data retrieval from Blockchain network.....	24
3.4 Project Milestone.....	25

3.5 Estimated Cost	26
3.6 Concluding Remark	26
Chapter 4 : System Design	27
4.1 Overview of System Design	27
4.2 Functional Modules in the System.....	32
4.2.1 User login and register module.....	32
4.2.2 Data request module	32
4.2.3 Private and public key authentication module	32
4.2.4 SMS authentication and authorization module.....	32
4.2.5 Administrator Management Module.....	32
4.3 System Flow	33
4.4 Database Design.....	34
4.5 System GUI Design	35
4.5.1 User Login Interface	35
4.5.1 Web application interface for doctor	36
4.5.3 Web-based mobile responsive design	37
4.6 Concluding Remark	38
Chapter 5 : System Implementation	39
5.1 Blockchain setup	39
5.1.1 Install Ubuntu 16.04.....	39
5.1.2 Install Pre-requisite	41
5.1.3 Install Hyperledger Composer	42
5.1.4 Setup Composer Playground	43
5.1.5 Setup Hyperledger Fabric	43
5.1.6 Configure in Composer Playground	44
5.1.7 Deploy Business Network	47
5.1.8 Start REST API.....	49
5.2 Website and database setup	50
5.2.1 Pointing domain to web server.....	50
5.2.2 Install Positive SSL to website	50
5.2.3 Connect to FTP and upload template files.....	53
5.2.4 Configure database through PHPMYADMIN	54
5.3 Program the system.....	56

5.3.1 Hyperledger modeling	56
5.3.2 Program the Chaincode	56
5.3.3 Design the Interface	57
5.3.4 Program functional modules	58
5.3.5 Integrate with Public/Private Key Generator: ECC library	58
5.3.6 Integrate with SMS Gateway API	59
5.3.7 Integrate with QRCode Generator Library.....	59
5.4 System Operation	60
5.5 Conclusion Remark	61
Chapter 6 : System Evaluation and Discussion	62
6.1 System Testing and Performance Metrics	62
6.2 Testing Setup and Result.....	62
6.2.1 Network Connection Test	62
6.2.2 System Response Speed Test.....	63
6.2.3 Rest API Retrieval Speed	64
6.3 Project Challenges.....	65
6.4 Objective Evaluation	66
6.5 Concluding Remark	66
Chapter 7 : Conclusion and Recommendation	67
7.1 Conclusion.....	67
7.2 Recommendation.....	68
REFERENCES.....	69
APPENDIX 2-Plagiarism check result.....	70

LIST OF FIGURES

Figure Number	Title	Page
Figure 1.1	Overall architecture of application	5
Figure 1.2	The structure and component of Blockchain	6
Figure 1.3	The model of hashing and encryption algorithm used by Blockchain's transaction	7
Figure 3.1	The flow of Agile Development Methodology	14
Figure 3.2	Linux Kernel-based Virtual Machine VPS	15
Figure 3.3	Peers in Hyperledger Fabric	16
Figure 3.4	Malaysia web hosting server	16
Figure 3.5	Ubuntu 16.04 (64bits) operating system	17
Figure 3.6	Hyperledger Fabric 1.4	18
Figure 3.7	Hyperledger Composer	18
Figure 3.8	Docker of Containers Technology	18
Figure 3.9	Concept of Docker compare to Virtual Machine	18
<i>Figure 3.10</i>	MySQL Database	19
Figure 3.11	cPanel Web Hosting Control Panel	19
Figure 3.12	VNC viewer developed by REALVNC	19
Figure 3.13	Java Programming	20
Figure 3.14	Python	20
Figure 3.15	cURL	21
Figure 3.16	Hyperledger Modeling Language	21
Figure 3.17	Web coding needed for website development	22
Figure 3.18	medrec.com.my registered with MYNIC	22
Figure 3.19	SSL Certificate	23
Figure 4.1	System Architecture Diagram	28
Figure 4.2	Flowchart of login process	29

Figure 4.3	Flowchart of request data from doctor to patient	30
Figure 4.4	Flowchart of doctor updates patient's record	31
Figure 4.5	System flow and the action can be performed by system users	33
Figure 4.6	Database relationship on MySQL	34
Figure 4.7	Web Application Design	35
Figure 4.8	Web Application Design for Doctor	36
Figure 4.9	Design for patient interface	37
Figure 5.1	VPSCHEAP KVM Basic detail page	39
Figure 5.2	Operation to VPSCHEAP KVM VPS	39
Figure 5.3	VPS OS selection	40
Figure 5.4	Remote login VPS via VNC Viewer	40
Figure 5.5	Create new user on Ubuntu 16.04	41
Figure 5.6	Hyperledger Fabric Pre-requisite	42
Figure 5.7	Composer Playground	44
Figure 5.8	Deploy new business network	45
Figure 5.9	Configure new business network	45
Figure 5.10	Connect to business network	46
Figure 5.11	Define model and testing on Playground	46
Figure 5.12	Playground model file editor	47
Figure 5.13	Generate new Peer Admin Card	48
Figure 5.14	Composer REST API	49
Figure 5.15	Configure nameserver of medrec.com.my	50
Figure 5.16	Issue SSL on NameCheap	50
Figure 5.17	SSL/TLS configuration on cPanel	51
Figure 5.18	Install SSL for website	51
Figure 5.19	Enter CRT of the certificate	52
Figure 5.20	SSL/TLS is running on web server	52

Figure 5.21	FTP tools Filezilla	53
Figure 5.22	Connect Filezilla to website FTP	53
Figure 5.23	Upload template to website	54
Figure 5.24	Database Management on cPanel	54
Figure 5.25	Create new database on cPanel	55
Figure 5.26	Manage database by using phpMyAdmin	55
Figure 5.27	Program the model with Hyperledger modeling language	56
Figure 5.28	Program Chaincode in JAVA	56
Figure 5.29	Designing MedRec interface	57
Figure 5.30	Program functional module in PHP	58
Figure 5.31	Import PHP ECC Library	58
Figure 5.32	Implement SMS Gateway API	59
Figure 5.33	Implement QRCode Generator Library	59

LIST OF ABBREVIATIONS

MSP	Membership Service Provider
PSN	Pervasive Social Network
SDLC	Software Development Life Cycle
LTS	Long Term Support
RDBMS	Relational Database Management System
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
PHP	Hypertext Preprocessor
SQL	Structured Query Language
FYP	Final Year Project
DV	Domain Validation
OV	Organization Validation
EV	Extended Validation
GUI	Graphic User Interface
ECC	Elliptic-Curve Cryptography
QR Code	Quick Response Code
SMS	Short Message Service
MYNIC	Malaysia Network Information Centre
VPS	Virtual Private Server
KVM	Kernel-based Virtual Machine
SHA	Secure Hash Algorithms
RSA	Rivest–Shamir–Adleman
FTP	File Transfer Protocol

LIST OF TABLES

Table Number	Title	Page
Table 3.1	Gantt chart show the project milestone (FYP 1 + 2)	25
Table 3.2	Estimated Cost Table	26
Table 6.1	Web server network connection test	62
Table 6.2	VPS network connection test	63
Table 6.3	Web page response speed test	63
Table 6.4	Rest API retrieval speed	64

Chapter 1: Project Background

1.1 Project Motivation and Problem Statement

With the development of technology, people are paying more and more attention to their health. Typically, medical records of patients are stored separately by each medical institutions. In this case, the medical records are not synchronize between every medical institutions. Therefore, when we go to new clinics that we did not visit before, we need to register repeatedly. Besides that, the doctors are not able to handle your case efficiently because they do not have any your previous medical records including surgery, medicine and medical supply. Another point is that current method of storing these sensitive medical data is stored in the local database of medical institution or using cloud storage, this will result that the data privacy exposes to the risk of data loss and vulnerable in the hands of malicious data users. The problems stated above also caused the research institutions difficult to obtain quality medical records and accurate data for further analysis.

In this project, a system which using Blockchain technology will be developed in order to amend the insufficient in the current methods to manage medical records. The system will create both public and private ledger to store the patients' data. Also, applying the Chaincode, Member Service Provider and Kafka consensus mechanism to ensure the privacy of the records. By implementing this system, each of the medical institutions can selectively store the records either publicly or privately. For example, genetic disease, allergy, surgery, medicine and the healthy condition of the patients can store in the public ledger, so permissioned parties such as medical center, insurance company or medical research institution can use these records easily and effectively. On the other hand, other data related to higher privacy like doctors and patients personal data, the cost of surgery, the date of patient visit the clinic will be recorded in the private ledger, which can only access by the institutions themselves.

This system is able to reduce the failure rate of surgery because the doctors are able to check the past medical records of the patient immediately and prevent the waste of time

when requesting data manually from other third party. As a result, patient's life will be more guaranteed and the success rate of surgery can be greatly improved.

1.2 Project Scope

This project is to develop a Blockchain 3.0 by implementing Hyperledger Fabric for protecting medical records of patient from security and integrity issue. In this project, we focus on the confidentiality and integrity of data, whereby all the medical records must up to date and synchronize between multiple institutions. In Hyperledger Fabric, all the members are enrolled Membership Service Provider to create a permissioned network. MSP is able to manage the role and permission of each users in order to use the application. Therefore, all the users are authenticated and identified by the system when using it. Therefore, the users request and operate to the medical records must be granted permission in order to access to the certain information. Chaincode will also apply in this situation to grant the permission for accessing medical records when certain requirements are fulfilled.

At the end of project, a web application that support mobile responsive is designed for patient to access their data and grant access to any third party. Besides that, a web-based backend system will be developed upon the Blockchain infrastructure. Therefore, the permissioned personnel such as doctors, hospitals, laboratories, pharmacists and health insurers are able to update and retrieve the latest medical records of users in a more efficiency way without conflict to the users' privacy.

1.3 Project Objectives

The aim of this project is subject to develop a Blockchain to store medical records in a more secure way and make the data become more useful for research institutions.

1. To store medical records in more secure way.

The traditional method of storing medical records is keep the data in a database server. Sometime the data will be inaccessible if the database server is down or the data will be exposed to attackers if the database server is vulnerable. The feature of Blockchain will assure that the blocks in ledger are not modifiable.

2. To synchronize the records between each system members.

Each medical institutions will keep their own copy in an independent database so it is difficult to trace the latest update of the records. Blockchain is a decentralized system, it store data distributed on multiple machine. The transaction added to the ledger is also make sure the data is not duplicated and ease to trace.

1.4 Impact, Significance and Contribution

In this project, there will bring benefit to patient, medical institution and research institution. For the patient, their privacy is more secure compared to the database-storing scheme. The patients are able to view their own record and grant access to any third party. Medical institutions are able to update and retrieve records with higher efficiency. Doctors can get the medical records of patients immediately if emergency surgery is need to perform. Consequently, the surgery success rate will increase due to the doctors have detailed records of patient. Research Institutions can get certain records from Blockchain according to the privileges assigned to them. The quality of records are directly affect to the quality of research, so that the researcher are able to produce a more trusted outcome. In long-term view, this system can potentially enhanced human life span and give a very positive impact to push the development of medical sector.

For the traditional way to store medical records, each of the medical institution store the records at their own centralized database server. Patient might keep duplicated records on different hospitals or clinics. Usually, the patients are not able to check on their own medical records. Furthermore, the medical institutions will restrict other parties to retrieve patient's records from their database due to security concern. For this reason, each parties need to manually request records from the specified institution. It may take a long time to get the full record and caused a delay of an emergency operation. This situation is also occurred on research institutions. There is only a limited records can be collected by the research institution due to the patients' privacy issue.

The problems stated above will make the medical institution spend a lot of human resources and time to solve the issues. In the proposed system, Hyperledger Fabric provides an alternative way for medical data storing. This project is mainly focused on storing medical records in a confidential way at the same time the data can be used by researchers who required the related data. By having this system, the medical institutions will increase the success rate if emergency surgery and lead the medical research to have a better development environment.

1.5 Proposed Approach

A web application will be designed for patient, doctor, researcher, insurance agency and so on. After they login to the system, they will be redirect to a mobile responsive web page to view the medical records. The web page is able to open on both computer or mobile phone and the web page will be adjusted automatically according to the device's screen size. This is able to provide user a convenient and high mobility application to control their medical records anytime and anywhere. However, doctor will be redirected to another web page which are more encourage to browse by computer. The reason is that doctors are more tend to update patient's medical record. Therefore, computer interface can ensure the high accuracy and provide more functionalities that can perform by the doctors.

The overall architecture is that we have web server, database server, Chaincode and Blockchain server. The web application is located at web server. When someone is requesting a patient's medical record, the request will send to Chaincode for authentication. Once the authentication done, the web application will retrieve the medical record from Blockchain server. The medical record will store in the database server and display the related information for users in web application.

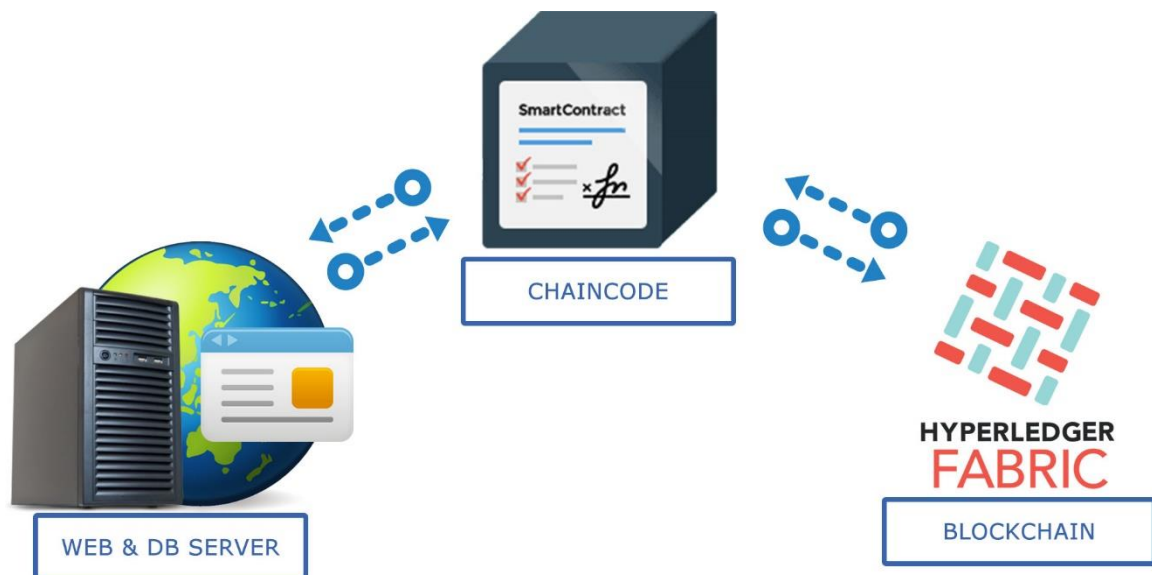


Figure 1.1 Overall architecture of application

1.6 Background Information

Information Technology has developed rapidly in this era. In this case, the secure and confidentiality of data is taken care of by all parties no matter application developers and the users. Blockchain as one of the popular technologies recently, it provides a secure way to store transaction either publicly or privately by applying distributed computing, asymmetric encryption and hashing. Block is the basic elements in Blockchain, which contained the record of new transactions such as medical data, cryptocurrency, client data or even voting records. Once the new block is created, it will be become unmodifiable and added into the Blockchain.

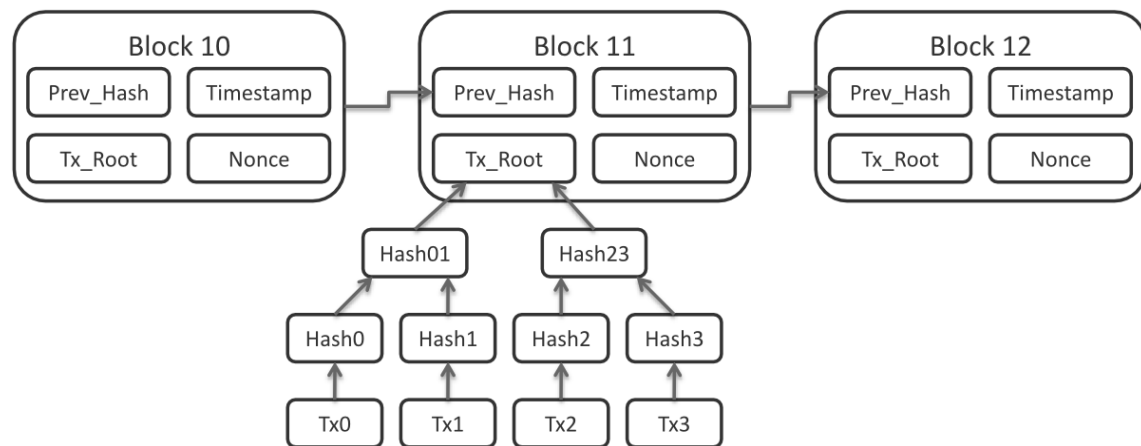


Figure 1.2 The structure and component of Blockchain

One of the most important features of Blockchain is decentralized. It means that Blockchain is not rely in single server to work, all the records are stored in the distributed database which hosted by millions of computers in worldwide. The distributed database also known as Public Ledger. In addition, Public Ledger is truly public and easily verifiable by anyone over the internet. If a hacker want to modify the records in Blockchain, the hacker must at least owned the computational power which greater than millions of computers. Another feature of Blockchain is digital signature, the new records will be hashing by using SHA-256 to generate a hashing value. After that, using the private key to encrypt the hashing value. Miners, who provide computational power to the Blockchain

will verify the encrypted information to ensure the information is real and correct before adding into the public ledger.

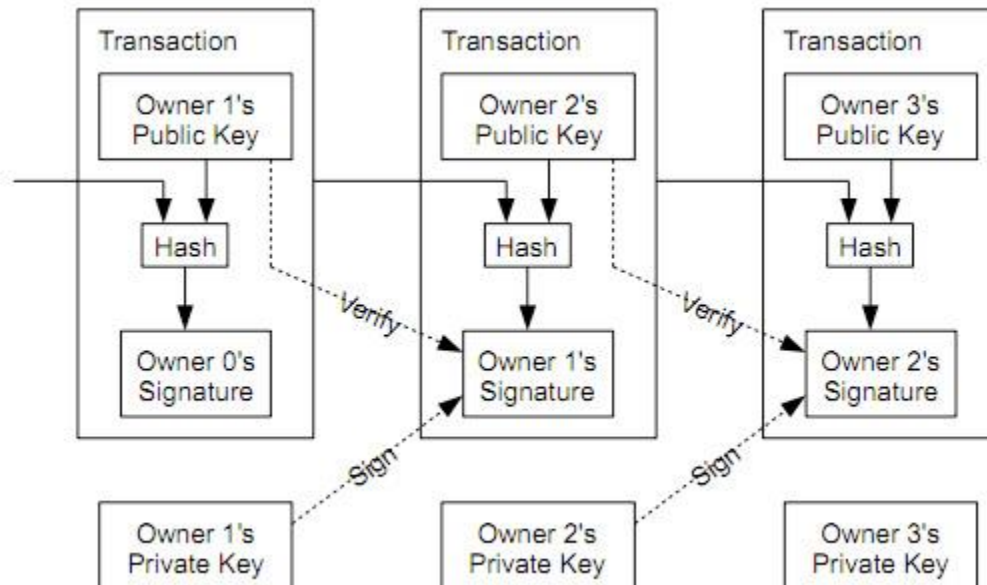


Figure 1.3 The model of hashing and encryption algorithm used by Blockchain's transaction

Currently, there are 3 versions of Blockchain which are Blockchain 1.0, Blockchain 2.0 and Blockchain 3.0. Bitcoin, the most famous cryptocurrency in the world, is the best example for Blockchain 1.0. Bitcoin stores the transaction records in the Blockchain. Bitcoin 2.0 introduced smart contract, software development and also decentralized autonomous organizations. This allows Blockchain to be implemented not just on financial transactions but virtually everything of value. In Blockchain 3.0, there are 3 main components which are Member, Blockchain and Chaincode. Channel also introduced to create separate ledger for publicly or privately usages.

1.7 Report Organization

In this report will be organized from chapter 1 to 7. The first chapter of the report is introduction about this project and Blockchain. The second chapter is Literature Review. In this chapter contains review of technologies about existing systems or related applications. The third chapter is about system methodology, system requirement, functional requirement, project milestone and estimated cost. The next chapter is about system design. In this chapter, we have the overview of the overall system. Many flow chart is also provided in this section in order to elaborate our system flow. Besides that, the database architecture, system GUI design are also included in this chapter. The chapter 5 shown the steps we implemented the system which including the Blockchain network, Chaincode and web application. Chapter 6 is about the system testing. We have run multiple of testing on our system in order to know the capability of our system to serve users. This chapter also included the project challenges and objective evaluation. The last chapter is about the conclusion and recommendation about what can be improve in future.

Chapter 2 : Literature Review

2.1 MeDShare:Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain

Qi Xia and her teammates (2017, p14765) having a project to purpose a system called “MeDShare”, which focus on applying cloud services, Blockchain and big data to design a data sharing model between cloud service providers in order to secure patient’s privacy from malicious activities. This project’s model has 4 main layers which are user layer, data query layer, data structuring and provenance layer and existing database infrastructure layer. The cryptographic scheme is that the authenticator will generate key pair for requestor and attached to smart contract in a package used for encrypting reports. Private Key is used by data requestor to digitally sign requests for data access. Besides that, Public Key is sent to the data owner in order to identifying for data access.

The strength of this system is to ensure secure data provenance and auditing by implementing smart contracts, which closely monitor all actions performed on a data thereby exposing the data threat level for a given malicious user by applying access revoking methods. Another strength is that ensure secured confidentiality of reports by self retrieving keys attached to smart contracts for encryption and also limiting the actions of malicious users on smart contact report.

Unfortunately, there will occur some security issue when sharing data between cloud service provider. The data stored on the cloud will also have the opportunity to be modified by malicious users. The problem can be solved through the implementation of Blockchain 3.0. In Hyperledger Fabric, all members are known by the system. Therefore, any of the malicious action perform by the users can be strictly avoid in order to ensure the confidentiality and integrity of the medical records.

2.2 Blockchain Technology for Improving Clinical Research Quality

Mehdi Benchoufi (2017, p4) purposes a system which are applying Blockchain 2.0 to collect the consent of participant for the clinical trial usage. This system use the features of Blockchain for tracking, sharing and caring of data. Besides that, this system also uses “Smart Contracts” to allow third-party to use the patient data for data analysis if the patient enables the request and the condition stated in the “Smart Contract” is fulfilled by the requestor. The system includes a decentralized tracking system with high security for data interactions for the purpose of clinical trials.

This system is able to make the clinical research worked in more trusted environment. Besides that, the clinical research becomes more transparency in order to enhance the trust between research communities, patient communities and research. The system will give a specific timestamp to each of the patient’s consent in Blockchain. After that, the system will request again the permission from patient for the consent renewal when each of the protocol revision. Therefore, the historicity and traceability of the records are assured. Apart from this, each of the transaction will be processed by a cryptographic validation before added into the Blockchain, so that the data integrity is also protected by the system.

Unfortunately, the system is not allow the users to manage their own data. The medical records are the valuable assets of the patients so it is necessary that the patients have the ability manage their records including the rights to grant permission to any parties for using the records. In this case, the solution to resolve this problem is implement Membership Service Provider (MSP) of Blockchain 3.0 into this system, so each members in the Blockchain enrolled by MSP and can set the role and permission of each users in order to use the system. As a result, the users will have the rights to control their own data.

2.3 Healthcare Data Gateways: Found Healthcare Intelligence on Blockchain with Novel Privacy Risk Control

Xiao Yue and his teammates(2016, p2) proposed a mobile application called Healthcare Data Gateway which architecture based on Blockchain to enable patient to own, control and share their own data easily and securely without violating privacy. A big challenge for healthcare data systems to become smarter is how to gather, store and analyze personal healthcare data without raising privacy violations. Patient data is one valuable asset of patient. A lack of adequate security measures in the past has caused in large amount of valuable data breaches, leaving patients exposed to economic threats and misuse by third party. Another problem is that, medical institutions and patients generally each have a different set of medical records gathered over the years. A central medical records storage is needed for the permissioned personnel to access data that are latest and more accuracy about the patient's health status.

The strength of this application is that it manages personal electronic medical data on Blockchain storage system instead of using traditional database and gateway. It also utilizes secure multi-party computation to enable third party to conduct processing on patient data without risking patient privacy. No any third party or single party has absolute power to affect the processing of patient's healthcare data. Moreover, this application provides a smarter way to use data and users are able to manage their own data by assign who have the permissions to view the data through their mobile application.

Unfortunately, the weaknesses of this application is that a party only can access the data after the data owner granted the permission although in some emergency case. For example, if the patient experienced an accident and need a surgery urgently, he lost the ability to grant permission to the hospital to access his healthcare data. Therefore, it will occur some delay if the hospital need to retrieve data from other traditional methods. The solution for this issue is applying "Smart Contact" to the application. Smart Contact is a hardcoded programmatic contact in Blockchain. It will execute only if the written conditions are happen.

2.4 A Secure System for Pervasive Social Network-Based Healthcare

Jie Zhang and his teammates(2016, p9250) proposed a secure system for PSN-based healthcare by using Blockchain technique. There are 2 protocols designed for the system. The first protocol is an improved version of the IEEE 802.15.6 which is used for authenticated association. It established secure links with unbalanced computational requirements for mobile devices and resource-limited sensor nodes. The next protocol used is Blockchain technique to share health data among PSN nodes.

The strength of this system is to reduce the burden on PSN nodes. As we known, PSN network is constrained by computational power and electrical supply. Therefore, a secure link is established between the nodes and smart phone to share and process the complex healthcare data easily on the mobile devices. Besides that, this system uses Blockchain to ensure the storage of PSN nodes will not be heavy load by the healthcare data. On the other hand, this system avoids data leakage caused by illegal behavior of an untrustworthy third party, since data stored in user's smart phone and the healthcare data are able to manage by users themselves.

Unfortunately, the limitation of this system is on efficiency of data sharing between medical institutions. The reason is that the data captured by PSN nodes is the records of physical representation of human body such as blood pressure, heart beat rate, respiratory rate and so on. These data can help the doctors remotely monitor the healthy status of the patient but not easily reuse by any other permissioned parties such as research institutions or insurance company. In this case, the solution for this issue is create both public and private ledger separately. The physical conditions captured by the PSN nodes are stored in the private ledger that only can be view by the doctors. After the data is review by the doctors, a diagnosis report will be produced to state the healthy status of the patient. The diagnosis report will store in the public ledger and users are able to share the access rights to any parties.

2.5 “MedRec” prototype for electronic health records and medical research data

Ariel Ekblaw and his teammates (2016, p11) proposed a system called “MedRec” which is a decentralized medical records management system to process against the Electronic Medical Records by applying Blockchain technology. This system allows patients to access their own medical information easily through their mobile devices, the information is updated from time to time which are retrieved from the Blockchain system. The purpose of MedRec system is to ensure the confidentiality and interoperable for the Electronic Medical Records system.

There are 4 strengths addressed by MedRec Blockchain implementation. The first advantage is avoid slow access to medical data. In the past, patient’s medical records are scattered across various organization. Therefore, it is difficult to access past data by anybody including the patients themselves and also will waste a lot of time to request, update or remove data if a record erroneously added. Second, the system interoperability is optimized. The implementation of Blockchain is able to resolve the interoperability issues that occurred between different services provider and medical systems and also make the data sharing process more effective. Next, the patient agency is prioritized in order to make the medical records and history are more transparent to the patients. The last strength is to improve the data quality and increase the data quantity for the purpose of medical research based on the past medical records of patients that are well organize. In this case, the records will easier to analyze by research institutions.

Unfortunately, MedRec is not able to ensure the data safety when the data is stored on the database of each individual provider. This means that if the system security of provider is not strong enough, there is a risk that the medical records will expose to unauthorized person at the provider site. This problem can be solve through avoid copying the patient’s record to provider’s local database. Smart Contract can be enhanced in this system in order to keep the data more safety.

Chapter 3 : System Methodology

3.1 System Development Model

The Software Development Life Cycle (SDLC) selected for the project management is Agile Development Methodology. This is a famous methodology adopted by software development companies nowadays for the purpose of minimizing security risk, cost, and changing requirements. The feature of this methodology is developing software in iterations. New functions are added into the software in each iteration. The benefit of multiple iterations is able to improve efficiency by finding and fixing bugs and expectation mismatches early on. This methodology relies on real-time communication with users to get up speed of development.

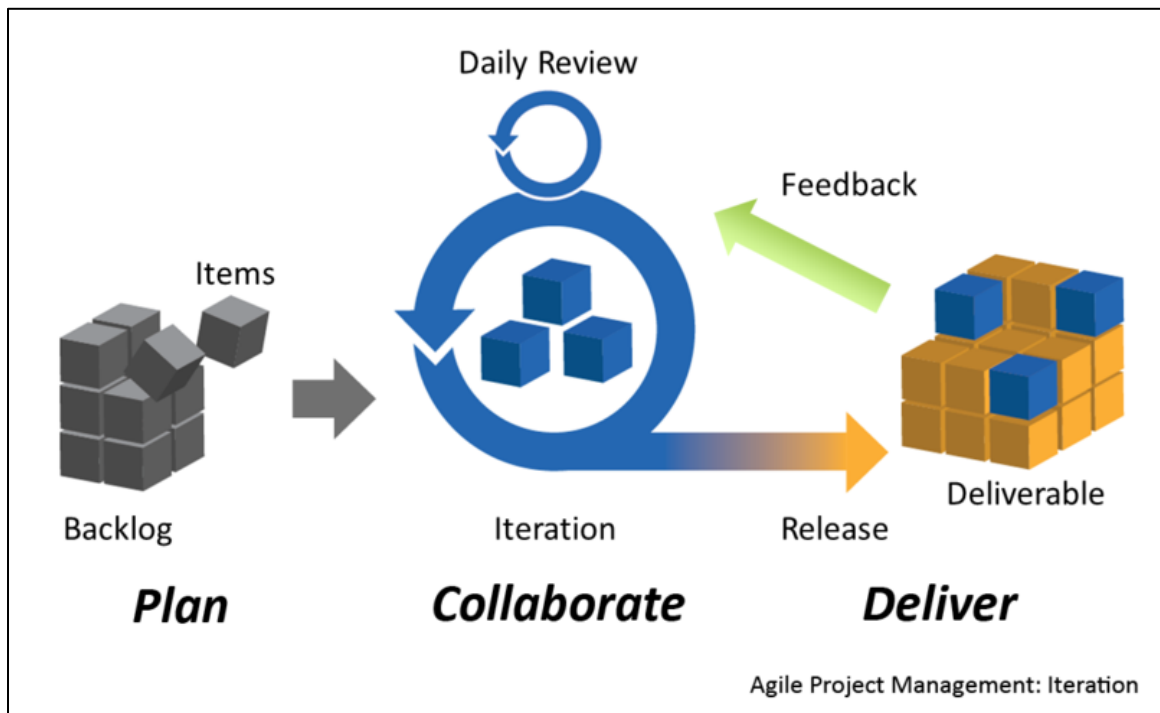


Figure 3.1 The flow of Agile Development Methodology

Agile Development Methodology requires a huge time commitment from the users due to each iteration need for user approval. Although this is inefficient in large organization, but undeniable this is quite suitable for the Blockchain development. The

reason is because we use Blockchain to store sensitive data, it is highly concern to the security of the system. Hence, several iterations in the development is able to strengthen the system by fixing the risks and bugs in order to build a powerful system to host the medical records.

3.2 System Requirement and Technology Involved

Hyperledger Fabric supports Ubuntu, MAC OS and Windows, but windows are highly not recommended to use because it will bring a lot of troubles when we develop the environment. In this case, we chose Ubuntu 16.04 64bits as our operating system to install the Hyperledger Fabric. The operating system is installed on a Linux Kernel-based Virtual Machine that hosted by VPSCHEAP.NET.

3.2.1 Hardware involved:

1. KVM VPS

The VPS server used in this project is provided by an US web hosting company, VPSCHEAP.NET. The package we used is SSD Basic package with Linux Kernel-based Virtual Machine (KVM). KVM allows us to run docker upon the kernel so that our Hyperledger can be operate with the docker engine.

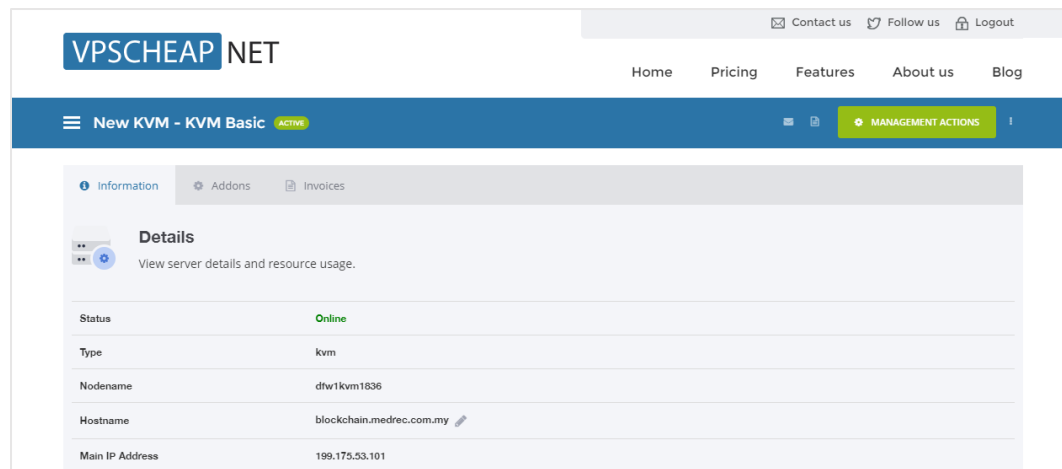


Figure 3.2 Linux Kernel-based Virtual Marchine VPS

2. Computers for Peer Node

Multiple computers are needed as the peer node of the Blockchain. Peers are the fundamental element inside Blockchain network because they are used to host ledgers and smart contracts.



Figure 3.3 Peers in Hyperledger Fabric

3. Malaysia Web Server

The web server we used in this project is provided by CloudStudio Technology. The web hosting panel is cPanel and the domain name is medrec.com.my. This web server is used to host the web application and the database so that all the users can visit the web application. This web application will take initiate to communicate with Chaincode in order to retrieve data from Blockchain Network.

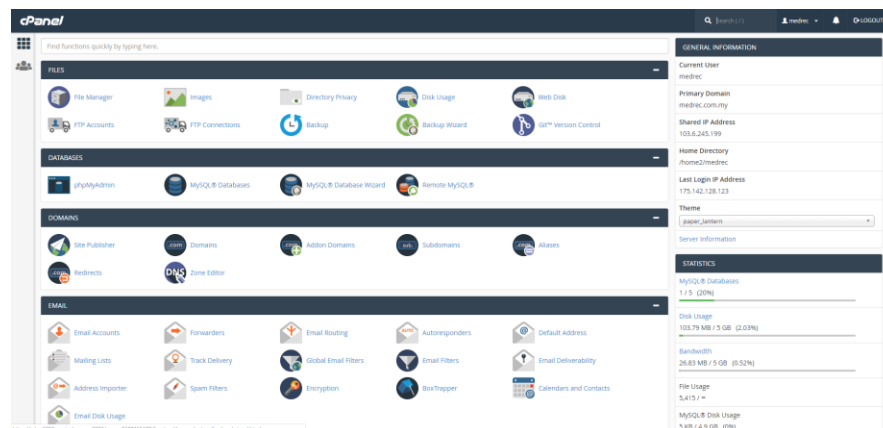


Figure 3.4 Malaysia web hosting server

3.2.2 Software involved:

4. Ubuntu 16.04 (64bits)

Ubuntu is an open source Linux operating system based on Debian. Ubuntu offered in three official editions which are Ubuntu Desktop for personal computer, Ubuntu Server for server and Ubuntu Core for internet of things. The editions we used to build up the Blockchain is Ubuntu Server.



Figure 3.5 Ubuntu 16.04 (64bits) operating system

5. Hyperledger Fabric 1.4

Hyperledger Fabric is a Blockchain framework which hosted by The Linux Foundation. It allows modules to be plug into its architecture such as consensus and membership services. Hyperledger Fabric leverages container technology to host smart contracts called “Chaincode” for applying specified code when certain condition are fulfilled.



Figure 3.6 Hyperledger Fabric 1.4

6. Hyperledger Composer

Hyperledger Composer is a set of tool for constructing the environment of Hyperledger Fabric for business usage. It make the installation process become simple and fast for business owners and Blockchain developers to build Blockchain application and smart contracts.



Figure 3.7 Hyperledger Composer

7. Docker

Docker is a computer program to perform operating-system-level virtualization by running application within different “containers”. Each container can isolate the system resources and communicate through well-defined channel. All containers are more lightweight than virtual machine because they use same kernel.



Figure 3.8 Docker of Containers Technology

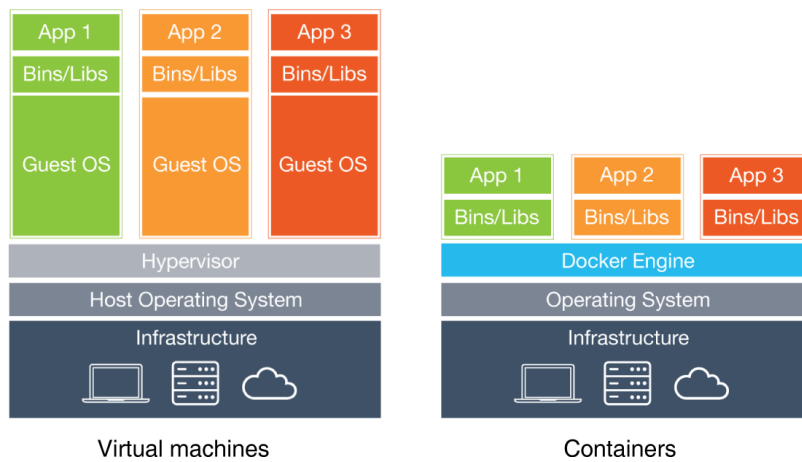


Figure 3.9 Concept of Docker compare to Virtual Machine

8. MySQL Database

MySQL is an open source relational database management system. The database is used to store the medical records which retrieve from Blockchain so that those data can be used in web application.



Figure 3.10 MySQL Database

9. cPanel Web Control Panel

cPanel is an online Linux-based web hosting control panel since 1997. It is famous on its first-class support and rich features for hosting the website. It provides graphical interface so users can directly modify the website coding on web browser.



Figure 3.11 cPanel Web Hosting Control Panel

10. VNC Viewer

VNC viewer is a software installed on the local computer and used to connect to the server. The server transmits a duplicate of the remote computer's display screen to the viewer. We used this software to remotely manage our KVM VPS in US.



Figure 3.12 VNC viewer developed by REALVNC

3.2.3 Programming Language Involved:

11. JAVA Programming

Java programming language is a general-purpose computer-programming language that is concurrent, class-based and object-oriented. It is intended to let application developers "write once, run anywhere". As a result, we used Java to write the Chaincode instead of using GO language.



Figure 3.13 Java Programming

12. Python Programming

Python is a high-level programming language for general-purpose programming. Some of the files of Hyperledger Fabric is written in Python. Therefore, python IDE with the version of 2.7.x is required in our project.



Figure 3.14 Python

13. cURL

cURL is a computer software project providing a library and command-line tool for transferring data using various protocols. In this project, cURL is used in the Hyperledger setup and the REST API. The REST API allow us to use GET, POST, PUT and DELETE method to manipulate the data in Blockchain network.



Figure 3.15 cURL

14. Hyperledger Modeling Language

Hyperledger Composer includes an object-oriented modeling language that is used to define the domain model for a business network definition. We need to use this modeling language to define the assets, participants, transactions, and events. The output modeling file will in .cto extension.

```
/**
 * A vehicle asset.
 */
asset Vehicle identified by vin {
  o String vin
}
```

Figure 3.16 Hyperledger Modeling Language

15. Web Programming

HTML, CSS, JavaScript and PHP are required to create a website for medical institutions to control the medical records of patient. So that the new transactions are able to add into the ledger through this website.



Figure 3.17 Web coding needed for website development

3.2.4 Other components involved:

16. MYNIC domain name

MYNIC is a company limited by guarantee (CLG), which is regulated by Malaysian Communication and Multimedia Commission (MCMC). We have registered a domain name `medrec.com.my` with MYNIC and point it to the web server. Therefore, the users can direct access the web application through domain name.



Figure 3.18 medrec.com.my registered with MYNIC

17. SSL Certificate

SSL Certificates are small data files that digitally bind a cryptographic key to an organization's details. When installed on a web server, it activates the padlock and the https protocol and allows secure connections from a web server to a browser. SSL can be classified into Domain Validation(DV), Organization Validation(OV) and Extended Validation(EV). The SSL certificate used in our project is Domain Validation. In future, we are going to upgrade it to Extended Validation.



Figure 3.19 SSL Certificate

3.3 Functional Requirement

3.3.1 System role and actions

There have 4 different roles designed in our system, each role has different permission to perform different tasks. The first role is patient, patient is only allow to view the medical records, medical history and modify their MedRec profile. The second role is for insurance agent or researcher, they are able to view and request medical record. The next role is doctor. Doctor is allow to view, request, and update the patient's medical record. The last role is administrator which is allowed for manage the system and check the web application statistics.

3.3.2 Update medical records to Blockchain

The medical records are able to add into the Blockchain as a transaction through web application. Only the peers with authorized permission within the organization can perform the records update action. For example, the doctor can update the patient's medical profile after body checking.

3.3.3 Retrieve medical records to database server

Retrieve the medical records from Blockchain and store into the MySQL database. After that, the web application are able to query the records from database and show the result to the patient, doctors or researchers.

3.3.4 Design Chaincode to allow data retrieval from Blockchain network

Chaincode acts as an intermediary between Blockchain and web server. All the request came from users are needed to pass through the Chaincode. Therefore, Chaincode is able to verify the request. If the requirements stated in Chaincode is fulfill by the requester, then the requester can receive the medical records successfully from the Blockchain network.

3.4 Project Milestone

Task	Project Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Collection of Data														
Define project objective and scope														
Review of Technology														
Analysis for literature review														
*Report with supervisor current progress														
Determine functional requirements														
Define technologies involved														
Determine system development model														
*Report with supervisor current progress														
Planning the system architecture														
Documentation														
Initializing Environment														
*Report with supervisor current progress														
Building the system														
*Report with supervisor current progress														
Presentation of FYP 1														
Chaincode 1 Design														
Chaincode 2 Design														
*Report with supervisor current progress														
Testing the system														
Import sample data														
*Report with supervisor current progress														

Note: Task with * is the project milestone



3.5 Estimated Cost

Table 3.2: Estimated Cost Table

Agile Development Methodology was selected for the development of the project to make sure the progress is on the right track and the quality of outcome was assured through the iterations. The system requirement and functional requirement were clearly stated in order to figure out the resources needed and what skills should be handled when the development period. On the other hand, The Gantt chart shown the milestone of project in each specified time. At the end, the estimated cost needed for the FYP development and commercialization are listed. The cost for FYP development is focus on building up the system while commercialization is focus on the security and performance of the system.

Chapter 4 : System Design

4.1 Overview of System Design

Hyperledger Fabric has been chosen because it has enhanced the security and more suitable to develop application compared to the Blockchain 2.0. Besides that, SHA-256 is used to hash the medical records and RSA is used to encrypt the hashed value before attached to the Blockchain. In addition, Distributed Programming is used for Blockchain so that all the medical records are not just rely on single server. More than that, JAVA language also needed to program the Chaincode. There are 1 Chaincode will be used in this application which are used to verify the request so that the web application can retrieve the medical records from the Blockchain network. And the last, multiple of the computers are need for mining in order to generate new block for Blockchain. In another word, mining is the process that let the members of the network arrive at a consensus on the contents of the Blockchain. The orderer determines which transactions to add to the Blockchain and in what order. All members of the network rely on what the orderer says.

The objective can be achieved with these methods and technologies are because medical records is a valuable asset of the individual. Everyone are not willing that their personal medical data expose to public or misuse by any third party without permission. Blockchain as one of the most popular technology nowadays are able to resolve this problem easily due to its structure and algorithm. Furthermore, Hyperledger Fabric improve from Blockchain 2.0 and increase many new features to enhance the security and also the expandability is ensured for other industry sectors. Moreover, this application focus on flexibility to users so that the patients no need to register repeatedly when they went to different clinic or hospital. This also help the research institutions are able to obtain medical data for analyze more easily. As mentioned before, medical records are the assets of individual so everyone should have the capability to view and control their own records. As smart phones become more and more popular, it is very convenience that the web application designed in mobile responsive way in order to allow the users to access their data everywhere and anytime through their mobile phone.

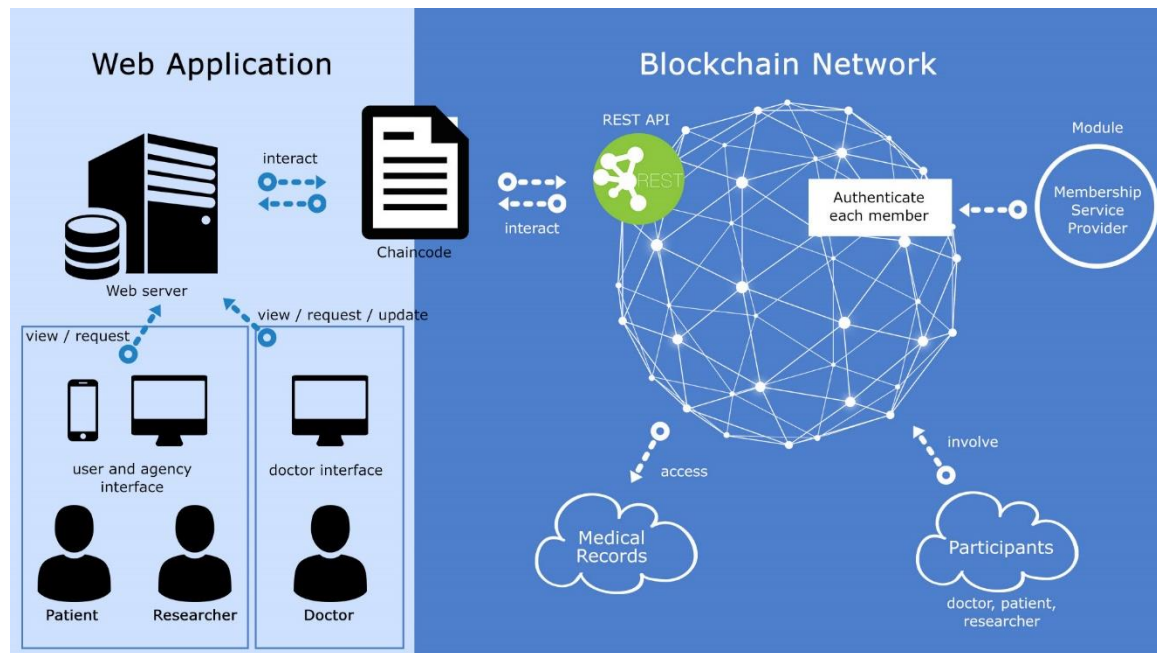


Figure 4.1 System Architecture Diagram

The Figure 4.1 shown the relationship between each entity in our Blockchain architecture. There are totally 3 parties will be involved as our target users which are patient, medical institutions and research institutions. The users will use the web application to perform view, request or update to the data. Once the web server received the request from users, then it will get verification from Chaincode. The web server is only allow to access to the Blockchain after it verified by Chaincode.

After that, the Blockchain will check the participant role and permission that assigned by the Membership Service Provider(MSP). All the members in the Blockchain must be identified in order to perform any actions. If the user permission is allowed, then the Blockchain will perform retrieve or update to the medical record according to the user's request received by Blockchain.

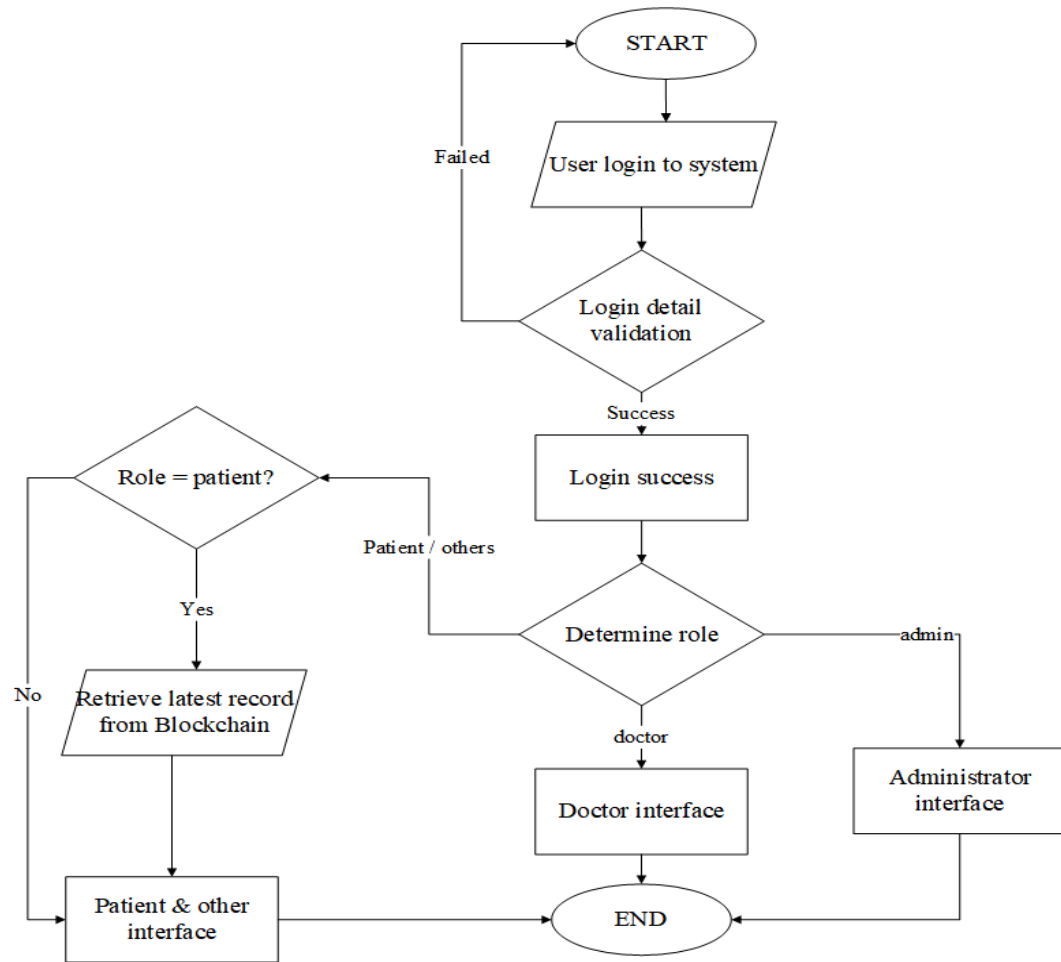


Figure 4.2 Flowchart of login process

The flowchart Figure 4.2 shows that the login process for each user role. When the users access to MedRec website, they need to login first before perform any actions. Once they login to the system successfully, then the system will check their role in database in order to redirect them to appropriate web page. The permissions of each role show as below:

Role	Permission
Admin	Full permission to manage the whole system
Doctor	View, create, update, request medical record of patient
Other	View, request medical record of patient
Patient	View own medical record and manage MedRec profile

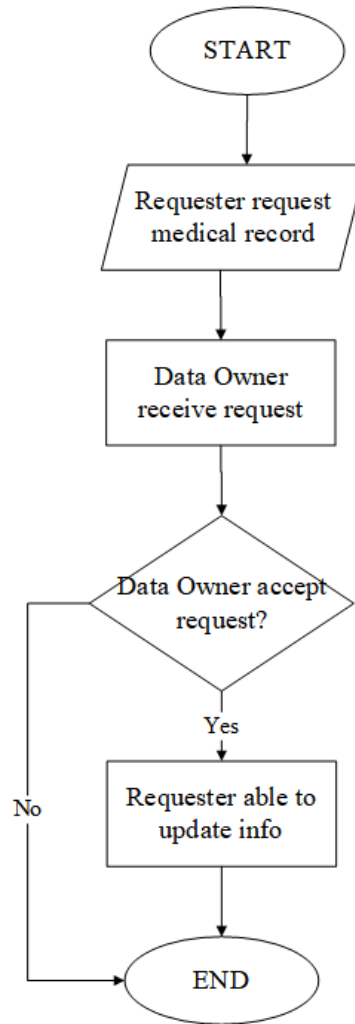


Figure 4.3 Flowchart of request data from doctor to patient

The Figure 4.3 shows that the flow of a requester to request medical record from patient. The roles whose allow to request medical records are doctor, researcher and insurance agent. All the above roles have the permission to view the patient's medical records. In addition, only the doctor has the rights to add new diagnosis records to the patient and update patient information in this system. When requester requests medical records from patient, patient will receive a SMS notification which contains the https link to approve the request. The verification and authorization of medical record is based on public and private key authentication. The patient uses the doctor's public key to assign permissions to the doctor to access the medical records. The authorization to access medical record is only valid for 1 day. If the record expired, then the requester need to request again from patient.

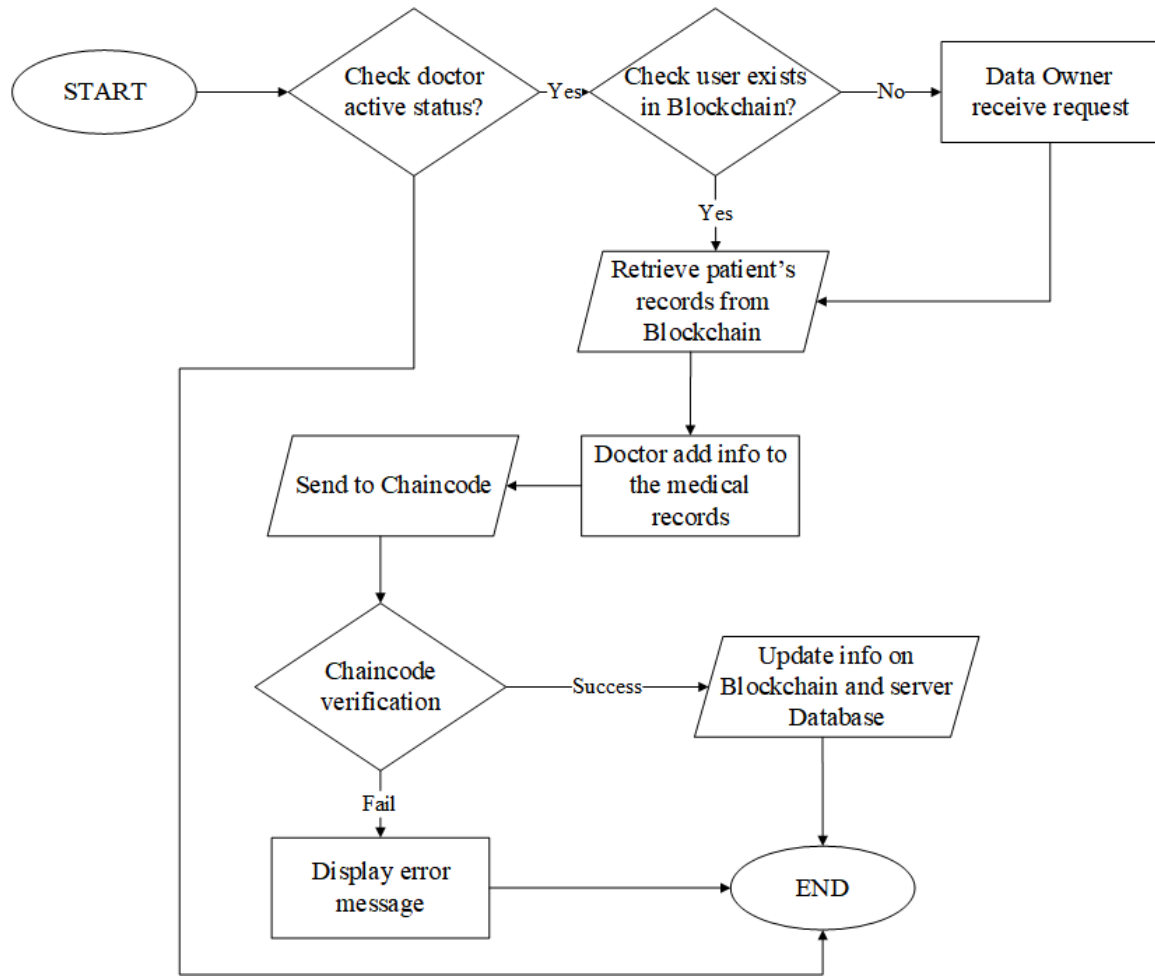


Figure 4.4 Flowchart of doctor updates patient's record

The Figure 4.4 shows that when a doctor is going to modify a patient's medical record, the system will check the doctor's active status at the beginning in order to know whether the doctor has the rights to perform this action. Then, doctor uses patient's identity card number to retrieve the latest medical record from the Blockchain network. After that, doctor can add new or modify to the patient record. The data retrieve process and data update process are required to verify by the Chaincode. All the manipulation to the medical record will be recorded and stored in the transaction database of the Blockchain. In addition, Hyperledger network is a fully permissioned network, every members in the network are identified. Therefore, every actions perform by the system users can be kept track in order to avoid any misuse of the medical record.

4.2 Functional Modules in the System

4.2.1 User login and register module

Every time the users login, the system will automatically retrieve the latest patient data from the Blockchain network. Therefore, we can ensure all the data view or modify by user are the up to date version. Besides that, this system is open for public to register. The patient account only can register by hospital. In addition, other roles are required to fill up an application form to apply an account.

4.2.2 Data request module

All the sensitive data are stored at the Blockchain network. In this case, the data request module is the critical part in this system. We need to ensure the data retrieve in high speed while the security of data is protected. The communication between web server and Blockchain network is through the Chaincode. Chaincode verify the request so that the request can be interact with REST API to manipulate data at Blockchain side.

4.2.3 Private and public key authentication module

We uses ECC library for the private and public key generation. This library is PHP based and generate the key pair through its algorithm. Each user will have their own key pair. The requester need to send the data owner his public key to get the authorization. The requester can access to the data only if he get approval from the data owner.

4.2.4 SMS authentication and authorization module

The SMS Gateway Provider we selected is iSMS.COM.MY. The reason is that they provided a good SMS API for PHP developer, so that we can easily use the API to send SMS through the website. SMS module will be applied for data request process and new registration of patient account.

4.2.5 Administrator Management Module

The admin login page is different with the normal user login page. Administrator has the highest privileges to control the whole website except the patient data authorization. Admin can create new account for doctor, researcher or insurance company. In addition, administrator can manage to all the data in the system. However, every actions done by the admin will also be recorded and store in the transaction database of the Blockchain.

4.3 System Flow

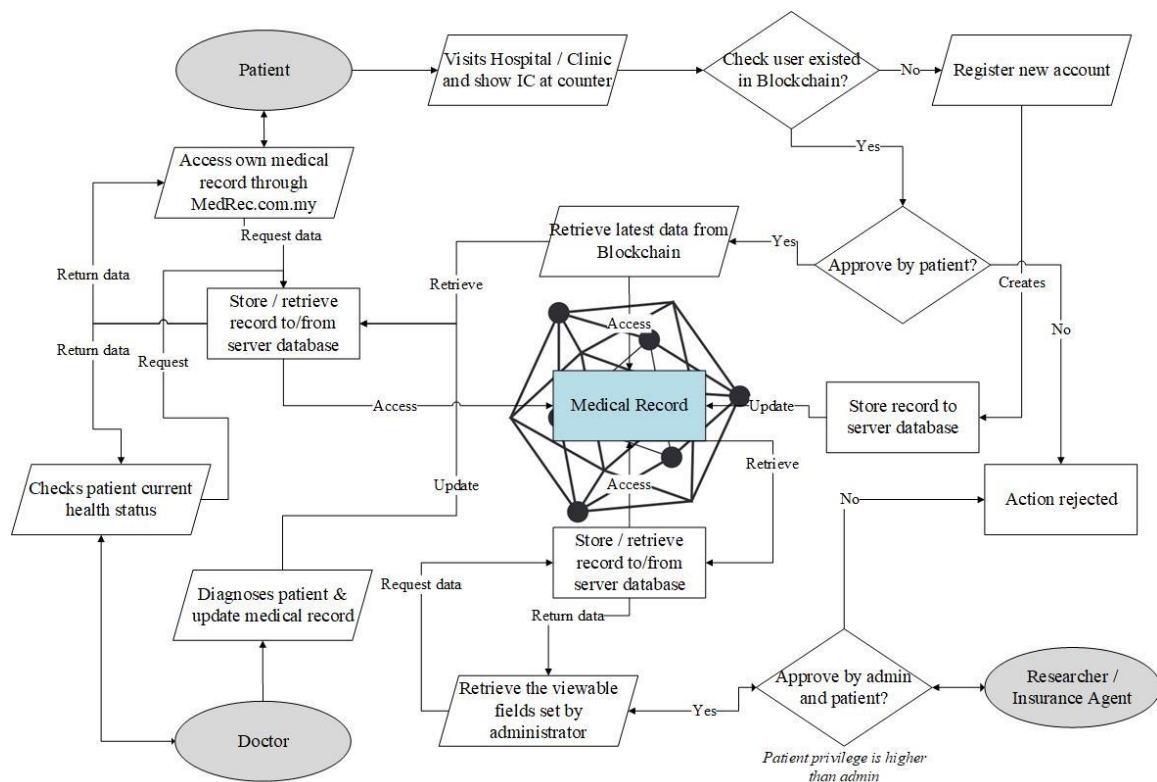


Figure 4.5 System flow and the action can be performed by system users

In the system flow diagram, there are 3 main entities which are doctor, patient and researcher/insurance companies. For the security reason, all the users are not allow directly interact with Blockchain network. They only can communication with the web & database server to access the medical record. The web & database server acts as an agent to interact with the Chaincode. Chaincode is the validator whose verify the request to medical record. Once Chaincode approved a request, the medical record will store into web & database server. A new transaction record will be created and stored at the transaction database in Blockchain network. The patients have the rights to control their own medical record to allow or avoid others to access it. The authorization methods implemented in this system are SMS authorization and public/private key authorization. The researcher only can access to the non-sensitive data which can be set by the system administrator according to the researcher institution's needs. The data get by the researchers will more focus on analysis purpose.

4.4 Database Design

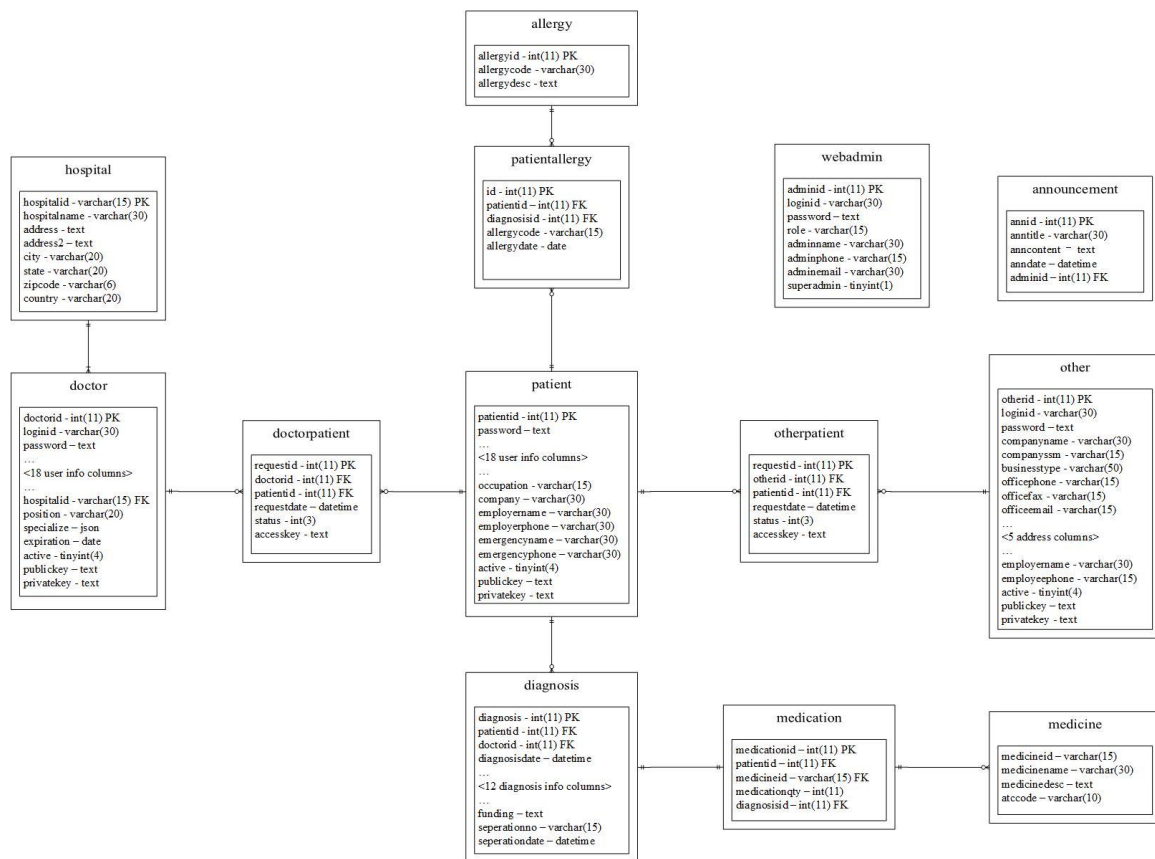


Figure 4.6 Database relationship on MySQL

The relational database management system chosen in our proposed system is MySQL. The diagram above shown the relationship between each of the table. There are totally 13 tables in our database system which are doctor, patient, other, webadmin, allergy, medication, medicine, hospital, diagnosis, doctorpatient, otherpatient, patientallergy and announcement. In this case, doctorpatient, otherpatient, patientallergy and medication are joining table to join two table which relationship in many to many. Webadmin table is store the records of admin, admin is able to manage all the records in website and the announcement is used to store the message that administrator wish to share to the users.

4.5 System GUI Design

4.5.1 User Login Interface

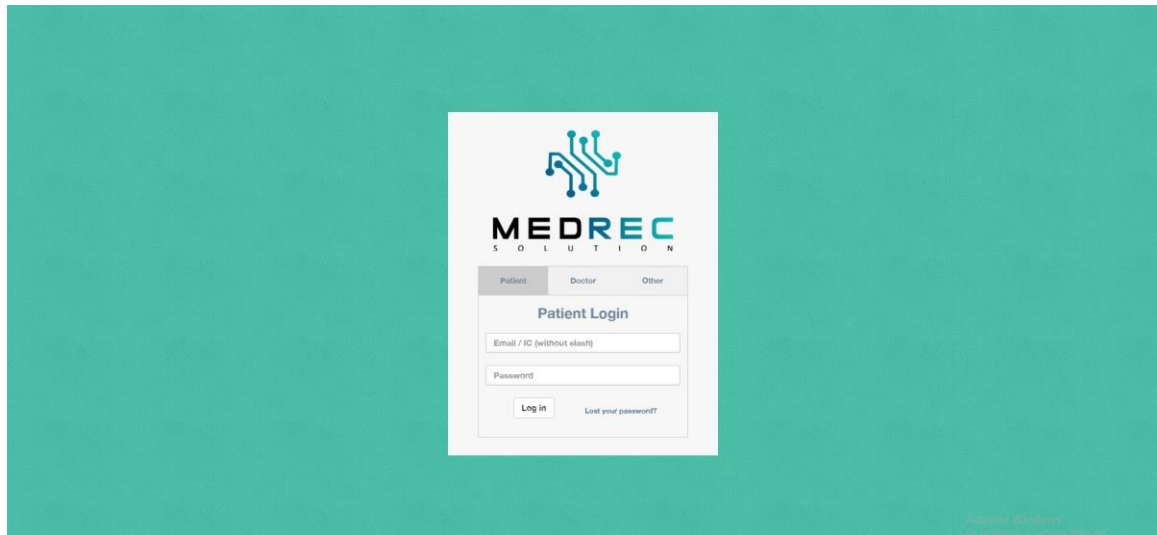


Figure 4.7 Web Application Design

This login interface allows patient, doctor and other roles to login. The login function was written in PHP. When the user login detail is match with the database, then the user session will be created. Then, the user will be redirected to the appropriate interface. The actions that can performed by the each user role is different. It is based on their permission that set in the system. Patients are required to use their Identity card number as login username. Besides that, doctor and other must login with their unique ID that assign by the system. The login URL of users are <https://www.medrec.com.my>. For the administrator role, there has another login portal for them to login. The login URL for administrator is <https://www.medrec.com.my/admin>.

4.5.1 Web application interface for doctor

MEDREC

Welcome, Dr. David

Dashboard

Patients

Diagnosis

Request Data

Profile

Logout

Back to list

Basic Information

Photo	
Patient ID	1
First Name	Xiang Yang
Last Name	Tian
IC No.	960220013019
Email	xiangyang199@gmail.com
Home Phone	072123456
Mobile Phone	0198613210
Date of Birth	1996-02-25
Gender	Male
Race	Chinese
Marital Status	Single
Spouse	
Children	
Address (Line 1)	3, Jalan Bunga Keluwa
Address (Line 2)	Taman Masai
City	Masai
State	Johor
Postal Code	81750
Country	Malaysia
Occupation	Student
Company Name	Universiti Tunku Abdul Rahman
Employer Name	Tian Xiang Yang
Employer Contact	0198613210
Emergency Name	Lim Sook Young
Emergency Contact	0187593210
Public Key	04114090c2a0b17796b0e0c436f520b653a7b1019b53e574eaf7591e281d0b0e0c1500c70a271e41005540762c5a81ab91c2388fcd87c1d95e401d51175140643

Diagnosis Records

- ▶ 2019-03-31 00:00:00 by Dr. Sam David (Kampar Hospital)
- ▶ 2019-03-30 00:00:00 by Dr. Sam David (Kampar Hospital)

Medication Records

- ▶ Medicine taken on 2019-03-30 00:00:00 by Dr. Sam David (Kampar Hospital)

Biography Records

- ▶ Record taken on 2019-03-30

Allergy Records

- ▶ Allergy found on 2019-03-30

Copyright © 2019 MedRec Solution Sdn Bhd. All Rights Reserved.

Figure 4.8 Web Application Design for Doctor

The design of online dashboard we used is the open source template designed by Aigars Silkalns who published the template on the GitHub on 2016. This template is designed based on Twitter Bootstrap. Although the overall layout is provided, but we still need to modify a bit the layout by using HTML and CSS. On the other hand, all the functions in this system are written in PHP. There is also a lot of PHP library used in this system, such as Public/Private key generator, SMS Gateway API, QRCode generator and so on.

4.5.3 Web-based mobile responsive design

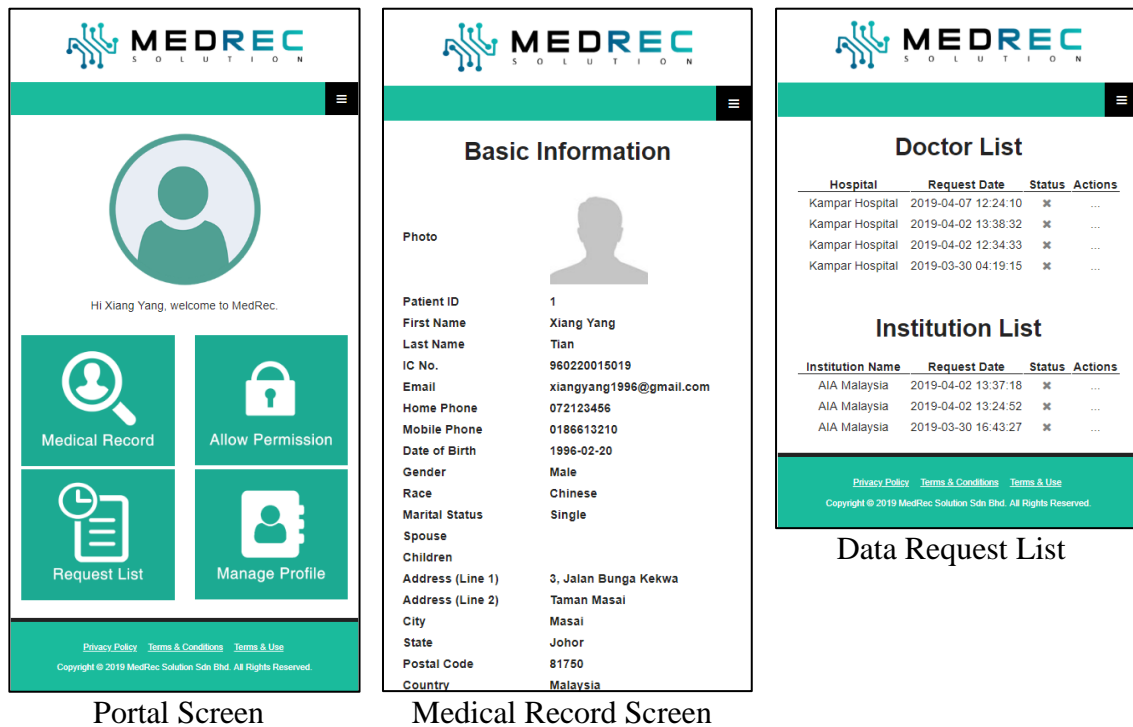


Figure 4.9 Design for patient interface

The figure above shown that the design of the patient interface. This web page is design in HTML, CSS and PHP. It is a web page but fully support for mobile responsive in order to fit various screen size of the mobile phone. Patients can use this web page to view the medical record, allow request for accessing data for any other party, check the previous medical history and so on.

4.6 Concluding Remark

In this chapter, the overall system design is described which including the system flow, login process flow, request data flow and update medical record flow. More than that, the module designed in this system also stated such as User login and register module, Data request module, Private and public key authentication module, SMS authentication module and Admin Management Module. Moreover, the MySQL table relationship and the interface design for each user role are attached for the reference.

At the FYP implementation stage, we will only use the Blockchain server as the endorsing peer due we only have very small amount of data. As we known, the number of the machines involved in the Kafka consensus ordering process will proportionally affect the performance of our system. Therefore, for the commercialized stage, multiple of dedicated server will be setup to provide the computational power to generate new data into a block and attach the blocks to the Blockchain. This can allow our system to handle a very large amount of data and enhance the system security by distributed store the medical record.

Chapter 5 : System Implementation

5.1 Blockchain setup

5.1.1 Install Ubuntu 16.04

Hyperledger Fabric is fully compatible with Ubuntu 16.04. In this case, we need to ensure we install this version of operating system on our VPS machine.

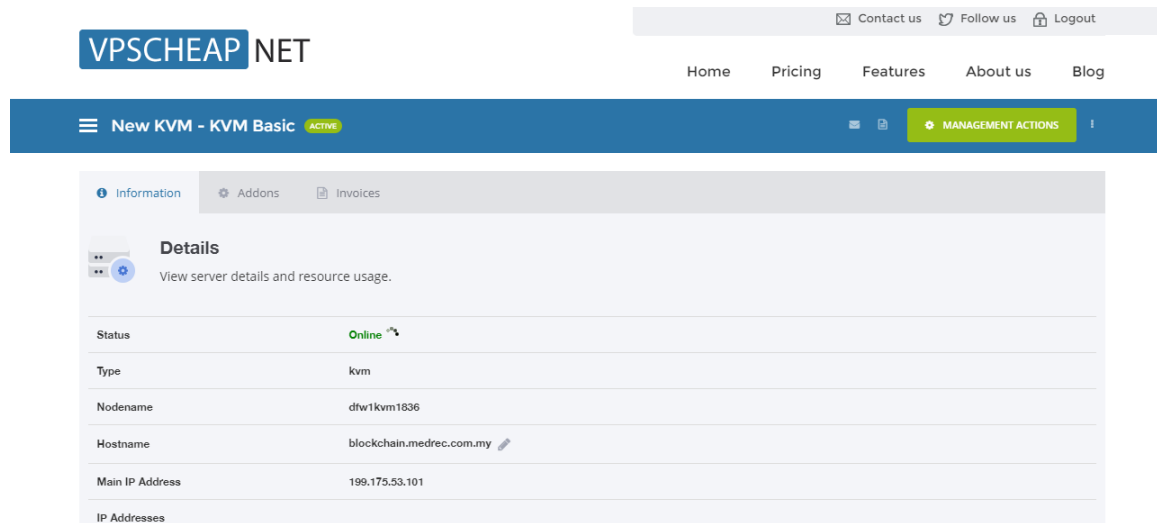


Figure 5.1 VPSCHEAP KVM Basic detail page

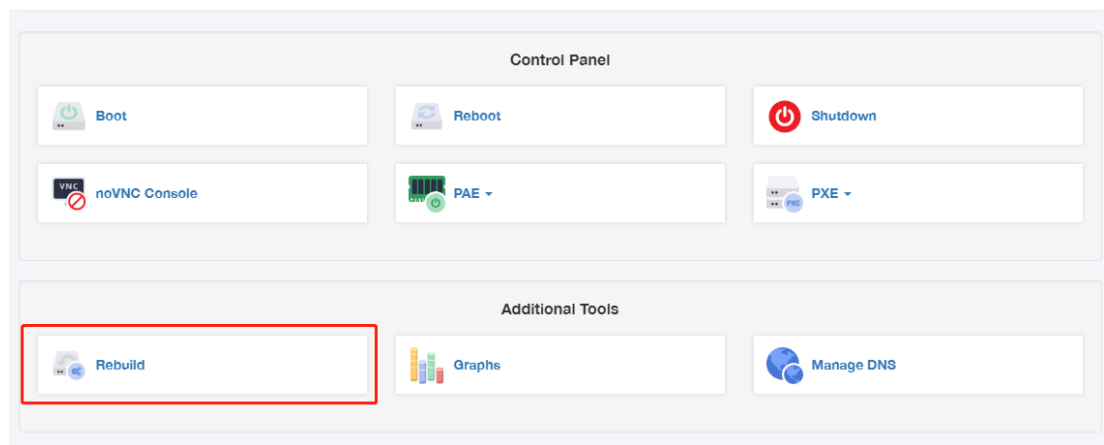


Figure 5.2 Operation to VPSCHEAP KVM VPS

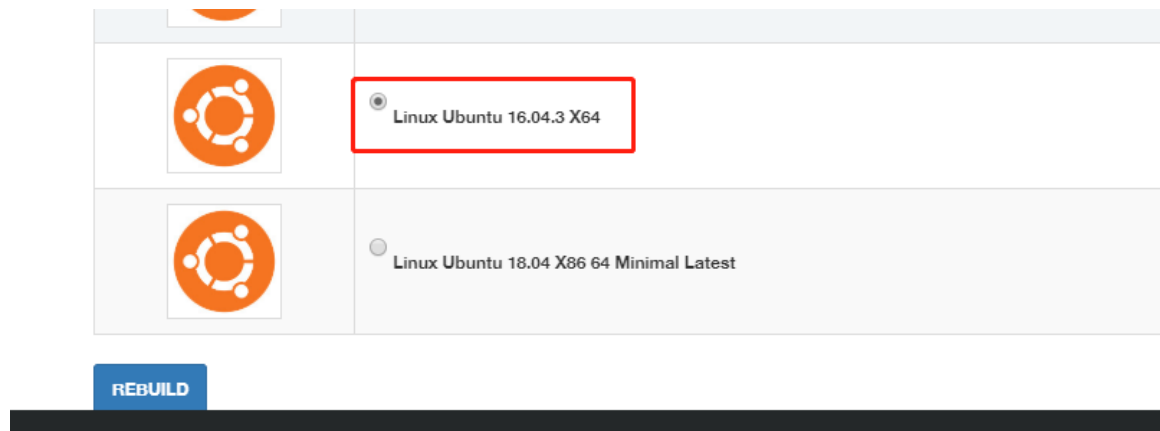


Figure 5.3 VPS OS selection

After login to the VPSCHPEAP account, then we go to “Rebuild” and select the appropriate operating system. Wait for 5 minutes for the VPS to complete initialization.

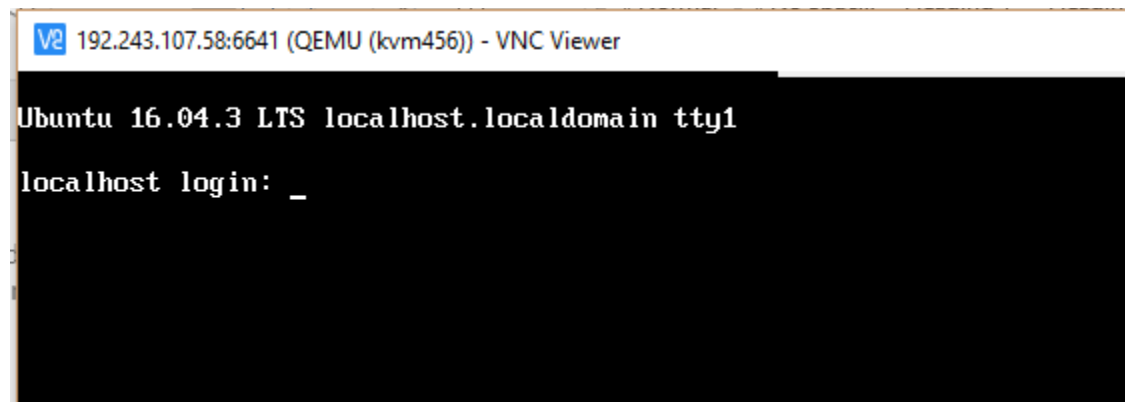


Figure 5.4 Remote login VPS via VNC Viewer

Once the initialization is completed, we are able to connect to the VPS by using VNC viewer.

```

passwd: password updated successfully
root@localhost:~# adduser medrec
Adding user `medrec' ...
Adding new group `medrec' (1001) ...
Adding new user `medrec' (1001) with group `medrec' ...
Creating home directory `/home/medrec' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for medrec
Enter the new value, or press ENTER for the default
    Full Name []: MedRec
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] Y
root@localhost:~# _

```

Figure 5.5 Create new user on Ubuntu 16.04

```
root@localhost:~# usermod -aG sudo medrec
```

```

root@blockchain:/etc# cat hostname
blockchain.medrec.com.my
root@blockchain:/etc#

```

For the first time login, we login with root. Then, we create a new user called ‘medrec’ for the installation of Hyperledger. Hyperledger is not recommended to install with root account. After created the new account, we need to assign the user to sudo group. Besides that, we need to check “/etc/hostname” to ensure the hostname is correct. If the hostname is using “localhost:localdomain”, then we need to use nano command to modify it. Note that the new hostname will not apply immediately, we need to reboot our VPS.

5.1.2 Install Pre-requisite

```
root@localhost:~# sudo apt-get update
```

```
root@localhost:~# sudo apt-get install curl
```

```
root@blockchain:/etc# sudo apt-get install nodejs
```

```
root@blockchain:/etc# sudo apt-get install npm
```

Run the first update, then we need to install curl, nodejs and npm. We will run shell script to install all pre-requisites later, so these software are necessary for the shell script to run.

```

root@blockchain:/etc# cd ../
root@blockchain:/# node.js -v
v4.2.6
root@blockchain:/# curl -O https://hyperledger.github.io/composer/latest/prereqs-ubuntu.sh

```

The “prereqs-ubuntu.sh” shell script is provided by Hyperledger. It will automatically install the following component.

- Docker Engine: Version 17.03 or higher
- Docker-Compose: Version 1.8 or higher
- Node: 8.9 or higher (note version 9 and higher is not supported)
- npm: v5.x
- git: 2.9.x or higher
- Python: 2.7.x

Figure 5.6 Hyperledger Fabric Pre-requisite

After we used curl to download the shell script, then we change the permission and run it.

```

root@blockchain:/# curl -O https://hyperledger.github.io/composer/latest/prereqs-ubuntu.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 4001 100 4001    0     0  1449      0  0:00:02  0:00:02 --:--:-- 1449
root@blockchain:/# chmod u+x prereqs-ubuntu.sh
root@blockchain:/# ls
bin  dev  home  initrd.img.old  lib64  media  opt  proc  run  snap  sys  usr  vmlinuz
boot  etc  initrd.img  lib  lost+found  mnt  prereqs-ubuntu.sh  root  sbin  srv  tmp  var  vmlinuz.old
root@blockchain:/# ./prereqs-ubuntu.sh

```

```

Ubuntu 16.04.3 LTS blockchain.medrec.com.my tty1
blockchain login: medrec

```

Once the installation is done, we need to logout the root account. Then, login again with normal user account for the next step to install the development environment.

5.1.3 Install Hyperledger Composer

Install Essential CLI tools:

```

medrec@blockchain:~$ npm install -g composer-cli

```


Install utility for running a REST Server on your machine to expose your business networks as RESTful APIs:

```
medrec@blockchain:~$ npm install -g composer-rest-server
```

Install useful utility for generating application assets:

```
medrec@blockchain:~$ npm install -g generator-hyperledger-composer
```

Install Yeoman. Yeoman is a tool for generating applications, which utilises generator-hyperledger-composer:

```
root@blockchain:/# sudo npm install -g npm_
```

Install the npm global post install to avoid installation failed of Yeoman.

```
medrec@blockchain:~$ npm install -g yo
```

After all the above tools are installed, we are going proceed to install the Hyperledger Playground.

5.1.4 Setup Composer Playground

Hyperledger Playground is also a development tool for develop Blockchain network. It run on development machine and giving the developer an UI for viewing and demonstrating the business network. We need Playground to construct the entire model file before we implement our Hyperledger Fabric network.

```
medrec@blockchain:~$ npm install -g composer-playground_
```

5.1.5 Setup Hyperledger Fabric

We need to create a folder called “fabric-tools” in our /home/medrec folder. Then we use curl to download the Hyperledger Fabric into this folder.

```
root@blockchain:/home/medrec/fabric-tools# curl -O https://raw.githubusercontent.com/hyperledger/composer-tools/master/packages/fabric-dev-servers/fabric-dev-servers.tar.gz
```

```
root@blockchain:/home/medrec/fabric-tools# tar -xvf fabric-dev-servers.tar.gz
```

Unzip the compressed file.

```

root@blockchain:/home/medrec/fabric-tools# ./downloadFabric.sh
Development only script for Hyperledger Fabric control
Running 'downloadFabric.sh'
FABRIC_VERSION is unset, assuming hlfv12
FABRIC_START_TIMEOUT is unset, assuming 15 (seconds)
1.2.1: Pulling from hyperledger/fabric-peer
b234f539f7a1: Extracting [=====>                1  29.82MB/43.12MB
55172d420b43: Download complete
5ba5bbeb6b91: Download complete
4a...

```

Run the shell script `./downloadFabric.sh` to get Hyperledger Core file from server. It will take around 10 minutes to complete the download.

```

root@blockchain:/home/medrec/fabric-tools# ./startFabric.sh

```

After that, run the file “`./startFabric.sh`” to launch the Hyperledger Fabric.

5.1.6 Configure in Composer Playground

```

root@blockchain:/home/medrec/fabric-tools# composer-playground_

```

Now, all the components are implemented completely, we can run composer playground to build our Blockchain network.

The composer playground allow us to visit through the 8080 port of our VPS. Therefore, we access 199.175.53.101:8080 to access the composer playground.

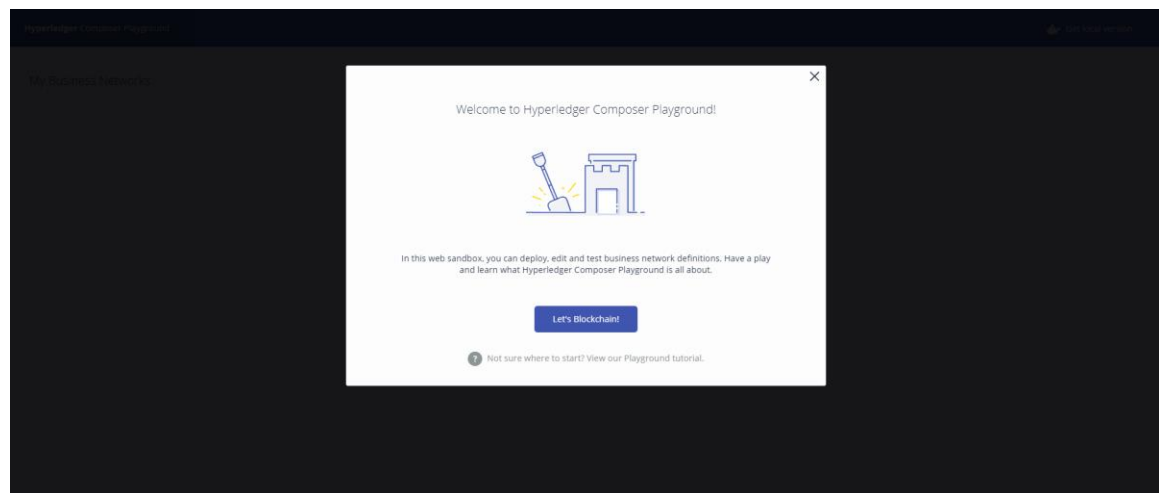


Figure 5.7 Composer Playground

Click “Let’s Blockchain” to continue the setup.

Blockchain-based Secure Medical Record Sharing System

Chapter 5: System Implementation

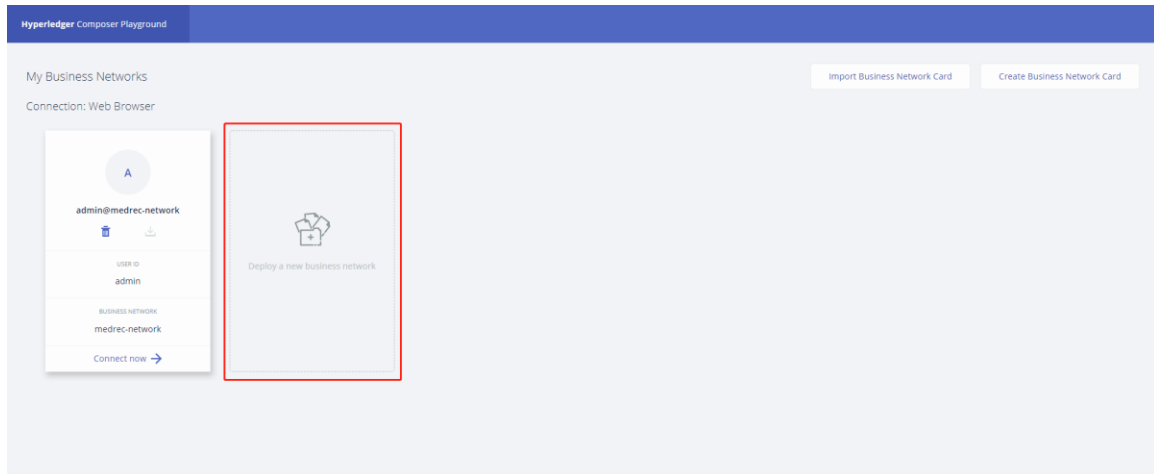


Figure 5.8 Deploy new business network

Select “Deploy a new Blockchain Network”.

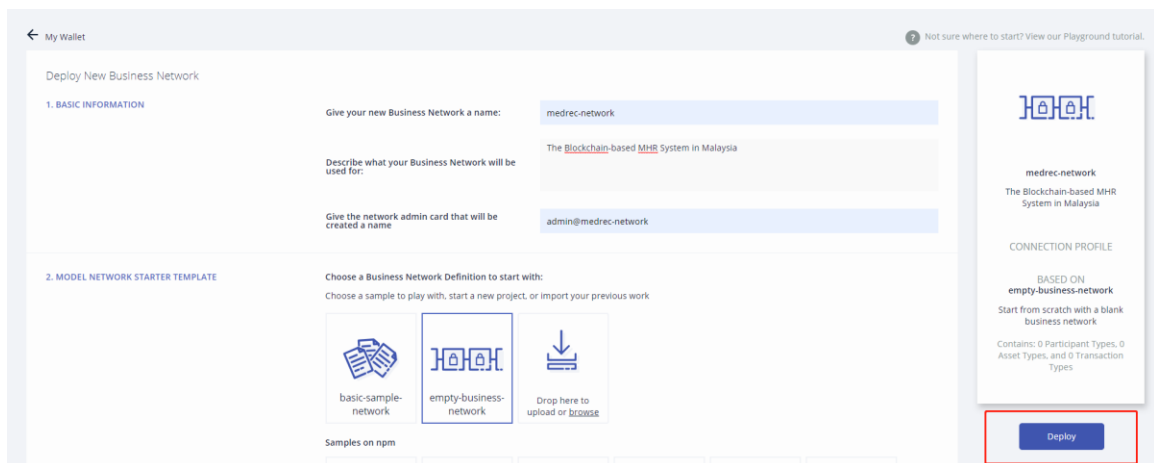


Figure 5.9 Configure new business network

Enter the basic information and deploy in empty business network.

Blockchain-based Secure Medical Record Sharing System

Chapter 5: System Implementation

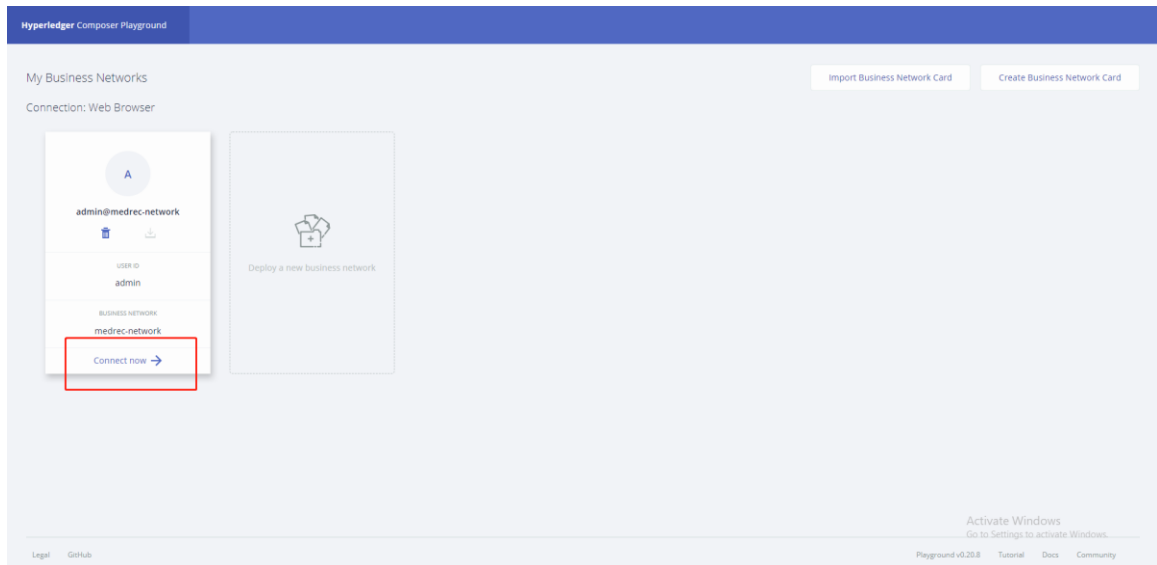


Figure 5.10 Connect to business network

Click “Connect now” to enter the medrec network.

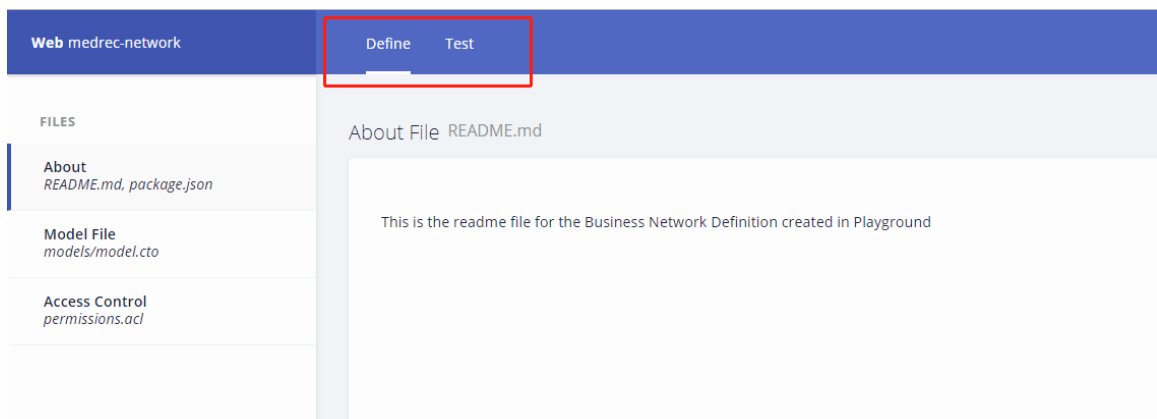


Figure 5.11 Define model and testing on Playground

At here, we can use the “Define” to define the model file and “Test” for checking what is going on through the transaction generated in the Blockchain.

```

1  /**
2   * Medical Record Blockchain Network
3   */
4
5  namespace org.medrec.network
6
7  asset Diagnoses identified by diagnosesId{
8      o String diagnosesId
9      o String diagnosesProblem
10     o String diagnosesCodeSystem optional
11     o String diagnosesCode optional
12     o DateTime diagnosesDate default="2018-08-26" // The time diagnoses record created
13     o String diagnosesOrg // Doctor visit by patient
14
15     // Bio Status
16     Blood Type
17     height
18     weight
19     o String diagnosesTemperature optional
20     o String diagnosesBloodPressure optional
21     o String diagnosesBloodSugar optional
22     o String diagnosesOxygen optional
23     o String diagnosesPulse optional
24     o String diagnosesNote optional
25
26     --> Patient owner // Record owned by Patient // Foreign Key

```

Figure 5.12 Playground model file editor

In our case, we need to define the following models:

- 4 Participants Models: Patient, Institution, Doctor, Hospital
- 2 Assets Models: Medication, Allergy
- 1 Transaction Model: AllowAccess

5.1.7 Deploy Business Network

```
root@blockchain:/home/medrec/fabric-tools# ./startFabric.sh
```

We need to run the start script before we deploy the business network.

```
root@blockchain:/home/medrec/fabric-tools# go hyperledger-composer:businessnetwork
```

Enter the command above to build the skeleton file.

```

Welcome to the business network generator
? Business network name: medrec-network
? Description: A Blockchain network for medical and research purpose
? Author name: Kevin Tian
? Author email: xiangyang1996@gmail.com
? License: (Apache-2.0)

```

Click enter to confirm the information used for build skeleton file (This information was entered when we build our business network in Composer Playground).

```
root@blockchain:/home/medrec/fabric-tools# archive create -t dir -n .
```

After that, we need to pack the business network we defined into a business network archive (.bna) file. This .bna file is used for deploy the business network. We can use command “ls fabric-tools” to see a file named “medrec-network@0.0.1.bna” is created under the fabric tools folder.

```
root@blockchain:/home/medrec/fabric-tools# ./createPeerAdminCard.sh
```

```
Successfully imported business network card
Card file: /tmp/PeerAdmin@hlfv1.card
Card name: PeerAdmin@hlfv1

Command succeeded

The following Business Network Cards are available:
Connection Profile: hlfv1
```

Card Name	UserId	Business Network
admin@medrec-network	admin	medrec-network
PeerAdmin@hlfv1	PeerAdmin	

```
Issue composer card list --card <Card Name> to get details a specific card
Command succeeded
```

Figure 5.13 Generate new Peer Admin Card

The next step is used this .bna file to create a Peer Admin card. We should copy the card name “PeerAdmin@hlfv1” for the installation step.

```
fabric-tools# composer network install --card PeerAdmin@hlfv1 --archive medrec-
network@0.0.1.bna
```

Now we can start to install the network by using the .bna file and Peer Admin Card.

```
fabric-tools# composer network start --networkName medrec-network --networkAdmin
admin --networkAdminEnrollSecret adminpw --card PeerAdmin@hlfv1 --file
```

Then, we start the network we created.

```
root@blockchain:/home/medrec/fabric-tools# composer card import --file networkadmin.card
```

After that, we need to import the network admin card by using the command above.

```
root@blockchain:/home/medrec/fabric-tools# composer network ping --card admin@medrec-network_
```

We can run this command to check whether our network is running successfully or not.

5.1.8 Start RESTful API

```
root@blockchain:/home/medrec/fabric-tools# composer-rest-server
```

```
? Enter the name of the business network card to use: admin@medrec-network
? Specify if you want namespaces in the generated REST API: always use namespaces
? Specify if you want to use an API key to secure the REST API: No
? Specify if you want to enable authentication for the REST API using Passport: No
? Specify if you want to enable event publication over WebSockets: Yes
? Specify if you want to enable TLS security for the REST API: No
```

Use the above settings to start the composer rest server. Once the server runs successfully, then we can visit it through web browser at port 3000. URL: <http://199.175.53.101:3000>

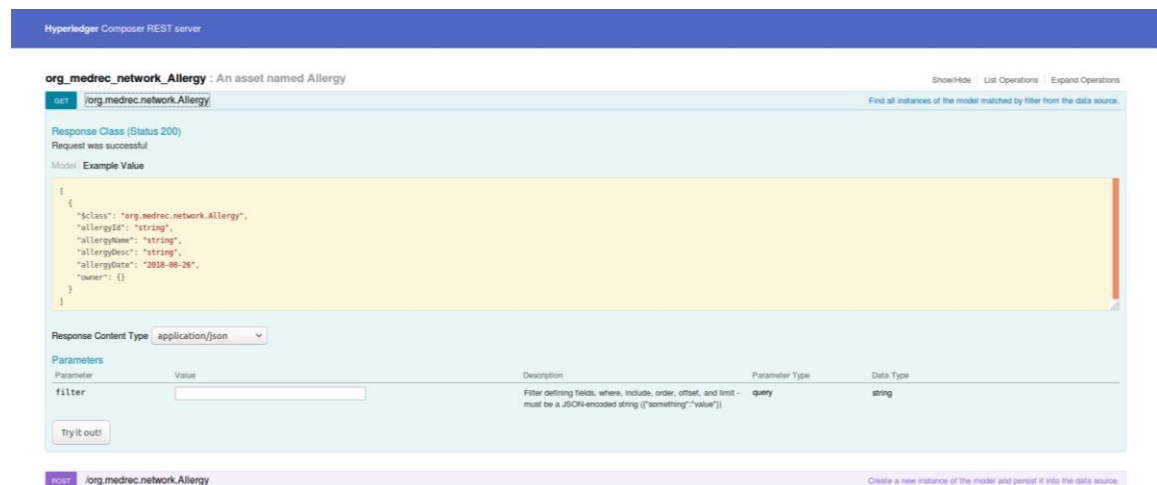


Figure 5.14 Composer REST API

The Composer REST API provided us all the methods can be used such as GET, POST, DELETE and so on. We can implement this method in our web application so that our web server can communicate with the REST API to retrieve or update the medical records.

5.2 Website and database setup

5.2.1 Pointing domain to web server

We have registered the domain name medrec.com.my. After that, we need to point the domain name to our web server through nameserver of the server.

Nameservers

You can change where your domain points to here. Please be aware changes can take up to 24 hours to propagate.

☐ Use default nameservers
☒ Use custom nameservers (enter below)

Nameserver 1	ns1.cloudwebhostserver.com
Nameserver 2	ns2.cloudwebhostserver.com
Nameserver 3	
Nameserver 4	
Nameserver 5	

Change Nameservers

Figure 5.15 Configure nameserver of medrec.com.my

5.2.2 Install Positive SSL to website

For the better security, we need to install SSL Certificate for the website. We purchased a SSL on NameCheap.

☐ medrec.com.my
 ADD CATEGORY
 Domain is with another registrar.

Mar 12, 2020
 SSL

Then we issue the certificate to activate it.

Certificate Versions

Certificate ID	Status	Secured Domains	
6191538	ISSUED	1 Domain	SEE DETAILS Reissue Download Certificate

Done

Figure 5.16 Issue SSL on NameCheap

Next, we go to our website panel: <http://www.medrec.com.my/cpanel>

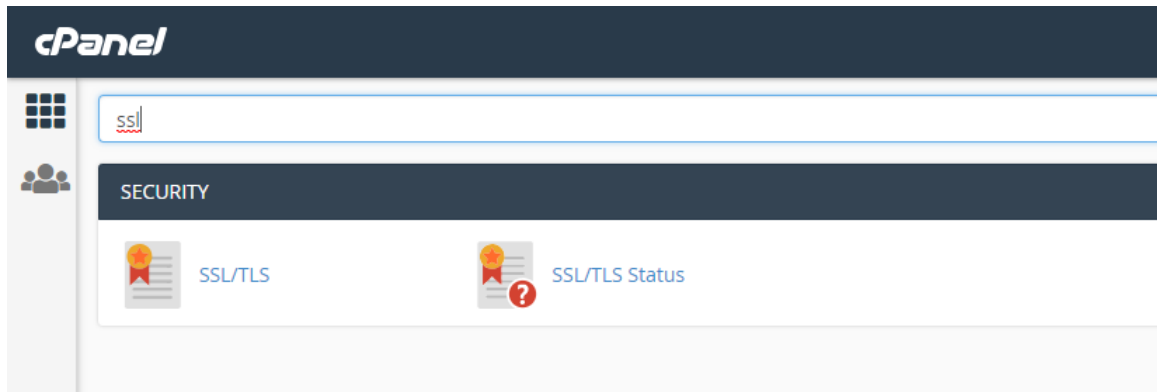


Figure 5.17 SSL/TLS configuration on cPanel

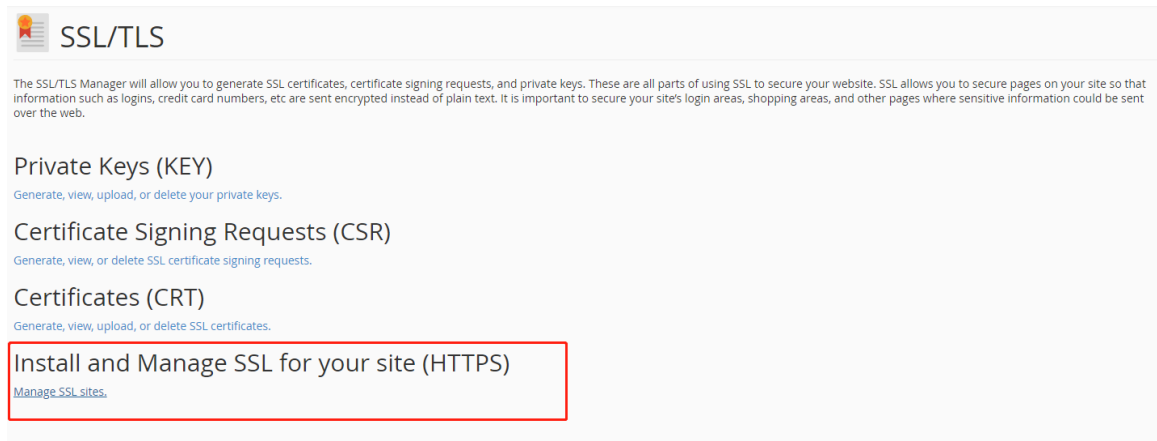
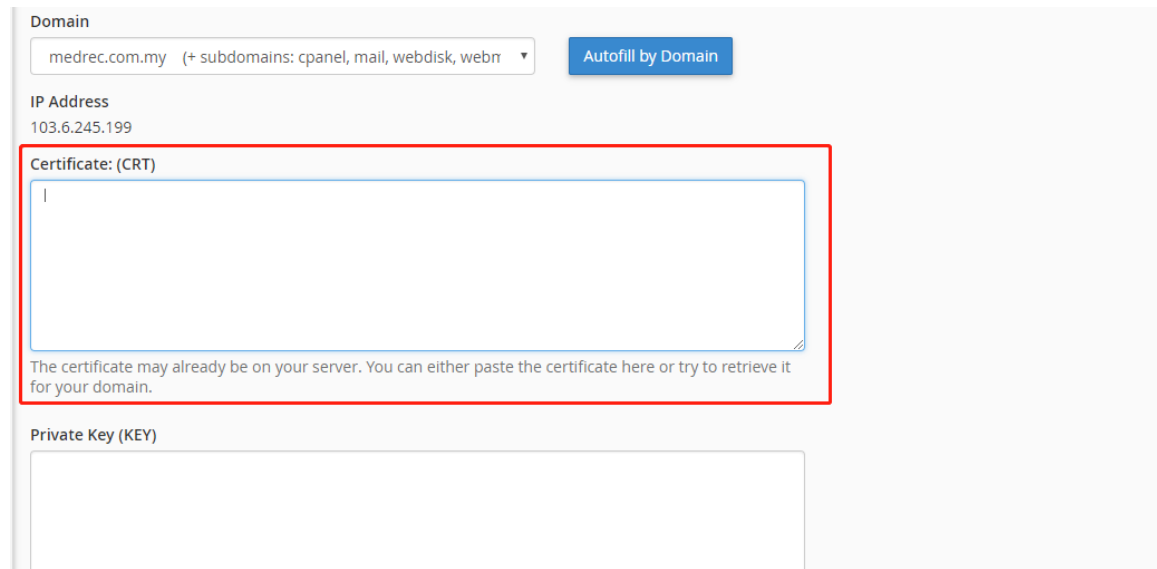


Figure 5.18 Install SSL for website

Choose “SSL/TLS”, then select “Install and Manage SSL for your site”.

Blockchain-based Secure Medical Record Sharing System

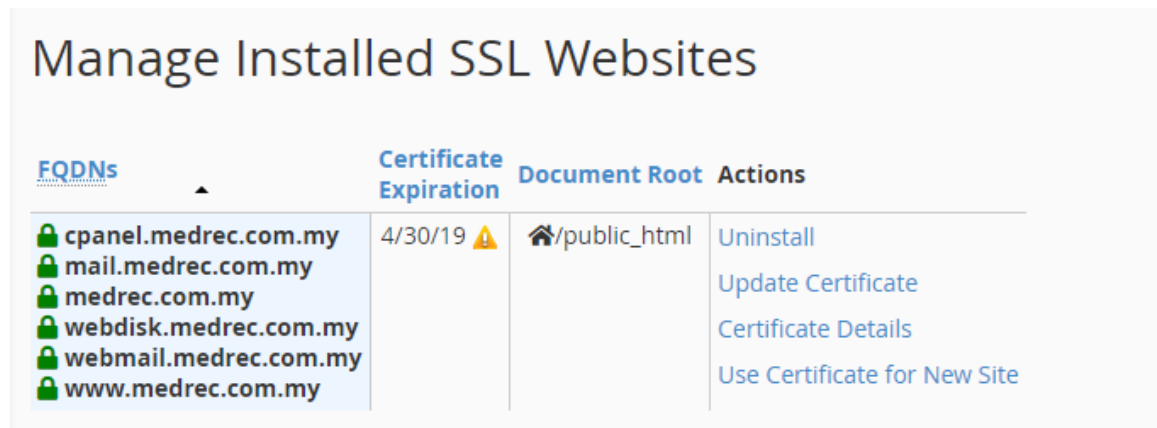
Chapter 5: System Implementation



The screenshot shows a web interface for installing an SSL certificate. At the top, there is a 'Domain' section with a dropdown menu showing 'medrec.com.my' and a list of subdomains: 'cpanel, mail, webdisk, webmail, www'. To the right of the dropdown is a blue button labeled 'Autofill by Domain'. Below the domain section is the 'IP Address' field, which contains '103.6.245.199'. The main section is titled 'Certificate: (CRT)' and contains a large, empty text area for pasting the certificate. Below this text area is a small note: 'The certificate may already be on your server. You can either paste the certificate here or try to retrieve it for your domain.' At the bottom of the form is a 'Private Key (KEY)' field, which is also empty.

Figure 5.19 Enter CRT of the certificate

Copy and paste the CRT which we get from the Certificate Authority when we issue the SSL.



The screenshot shows the 'Manage Installed SSL Websites' page. It features a table with four columns: 'FQDNs', 'Certificate Expiration', 'Document Root', and 'Actions'. The table lists six FQDNs: 'cpanel.medrec.com.my', 'mail.medrec.com.my', 'medrec.com.my', 'webdisk.medrec.com.my', 'webmail.medrec.com.my', and 'www.medrec.com.my'. The 'Certificate Expiration' column shows '4/30/19' with a yellow warning icon. The 'Document Root' column shows '/public_html'. The 'Actions' column contains links for 'Uninstall', 'Update Certificate', 'Certificate Details', and 'Use Certificate for New Site'.

FQDNs	Certificate Expiration	Document Root	Actions
cpanel.medrec.com.my	4/30/19 ⚠	/public_html	Uninstall
mail.medrec.com.my			Update Certificate
medrec.com.my			Certificate Details
webdisk.medrec.com.my			Use Certificate for New Site
webmail.medrec.com.my			
www.medrec.com.my			

Figure 5.20 SSL/TLS is running on web server

We can see that the installation of SSL is completed.

5.2.3 Connect to FTP and upload template files

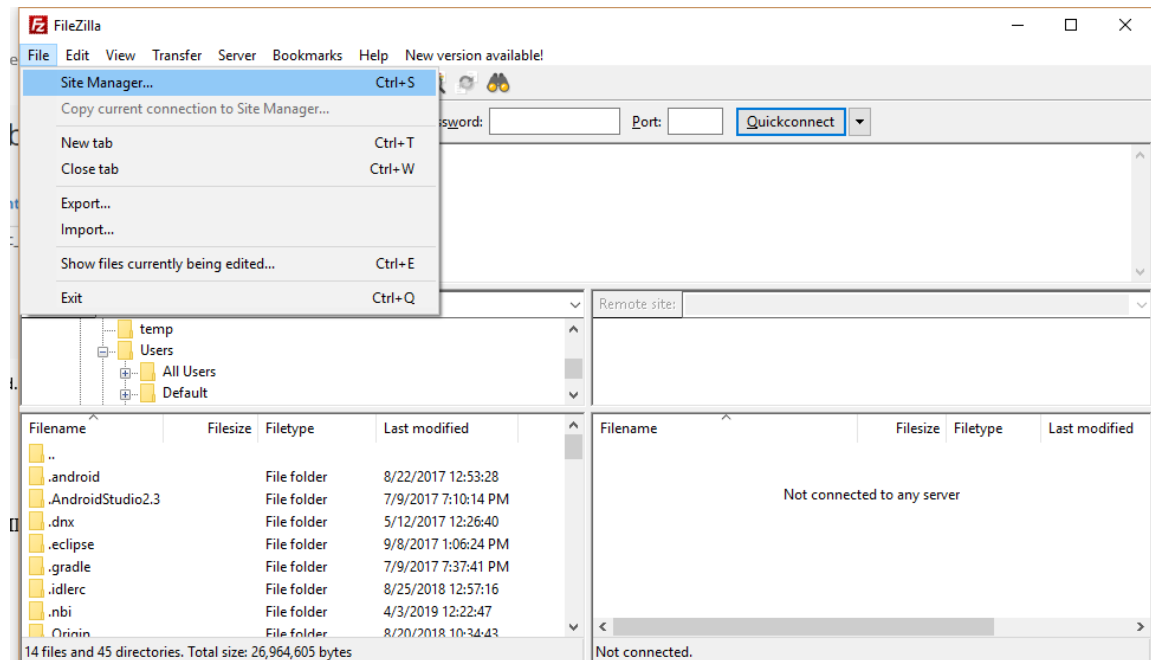


Figure 5.21 FTP tools Filezilla

Download Filezilla from <https://filezilla-project.org/>

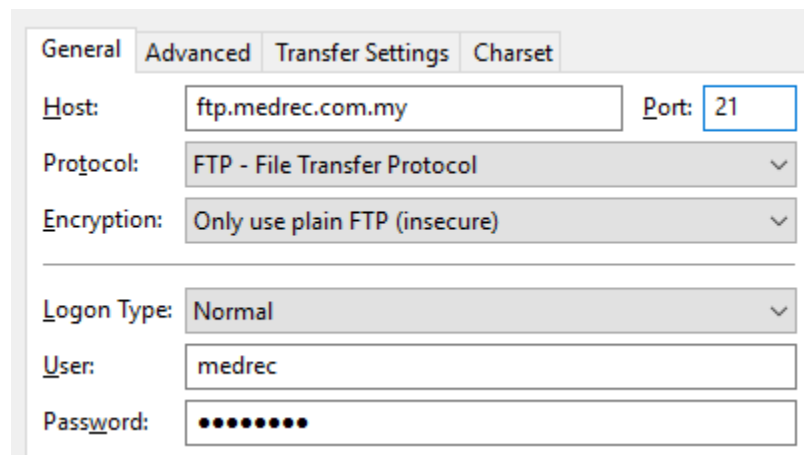


Figure 5.22 Connect Filezilla to website FTP

Fill in the host, port, username and password to login into FTP.

Filename	Filesize	Filetype	Last modified
..			
.well-known		File folder	1/30/2019 4:1
admin		File folder	3/30/2019 5:1
build		File folder	2/14/2019 1:1
cgi-bin		File folder	1/29/2019 5:3
classes		File folder	4/2/2019 4:41
common		File folder	4/3/2019 2:17
css		File folder	3/31/2019 10:

Figure 5.23 Upload template to website

Once connected successfully, we can upload the template file to “/public_html” directory.

5.2.4 Configure database through PHPMYADMIN

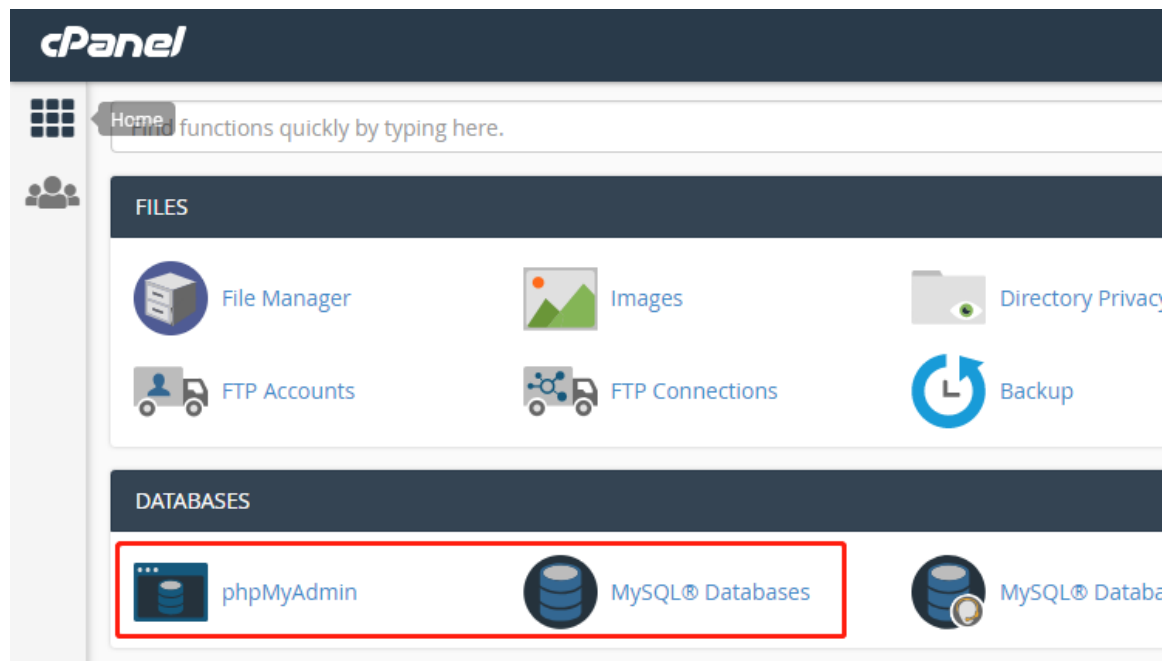


Figure 5.24 Database Management on cPanel

For the Database, we need login into our cPanel. We use “MySQL Databases” to create a new database and use phpMyAdmin to manage it.

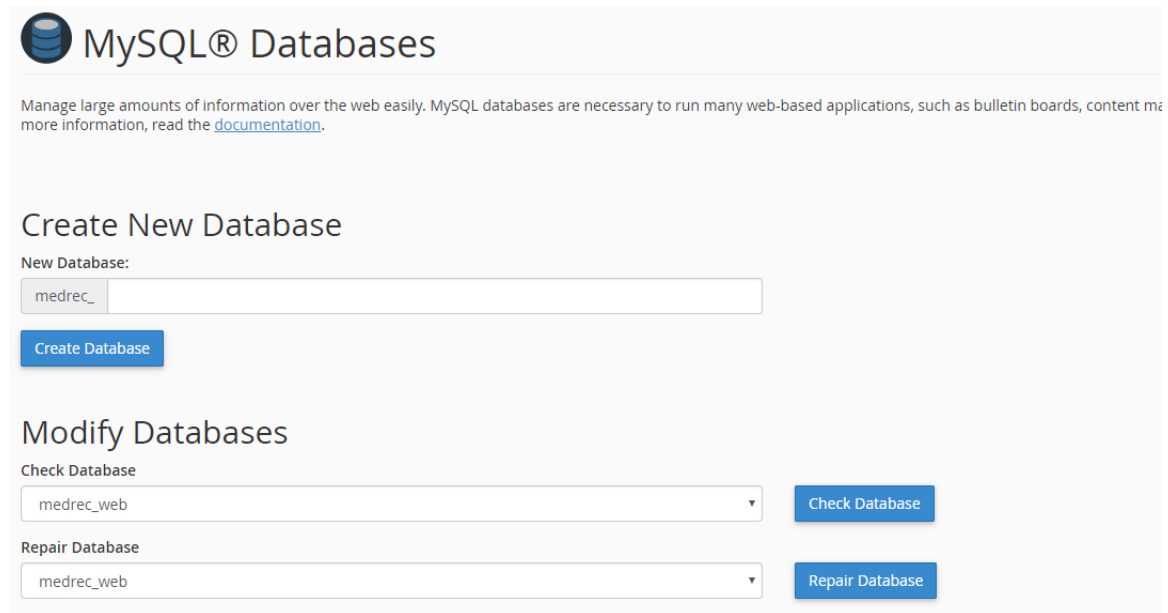


Figure 5.25 Create new database on cPanel

MySQL Databases page allow us to create new database and assign user privilege to user.

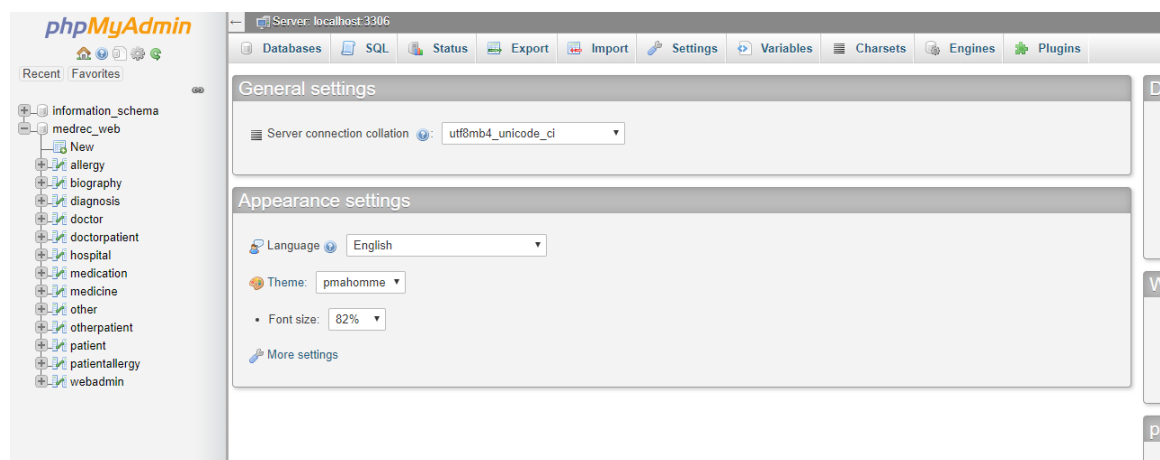


Figure 5.26 Manage database by using phpMyAdmin

phpMyAdmin allow us to manage our tables, rows and records in GUI.

5.3 Program the system

5.3.1 Hyperledger modeling

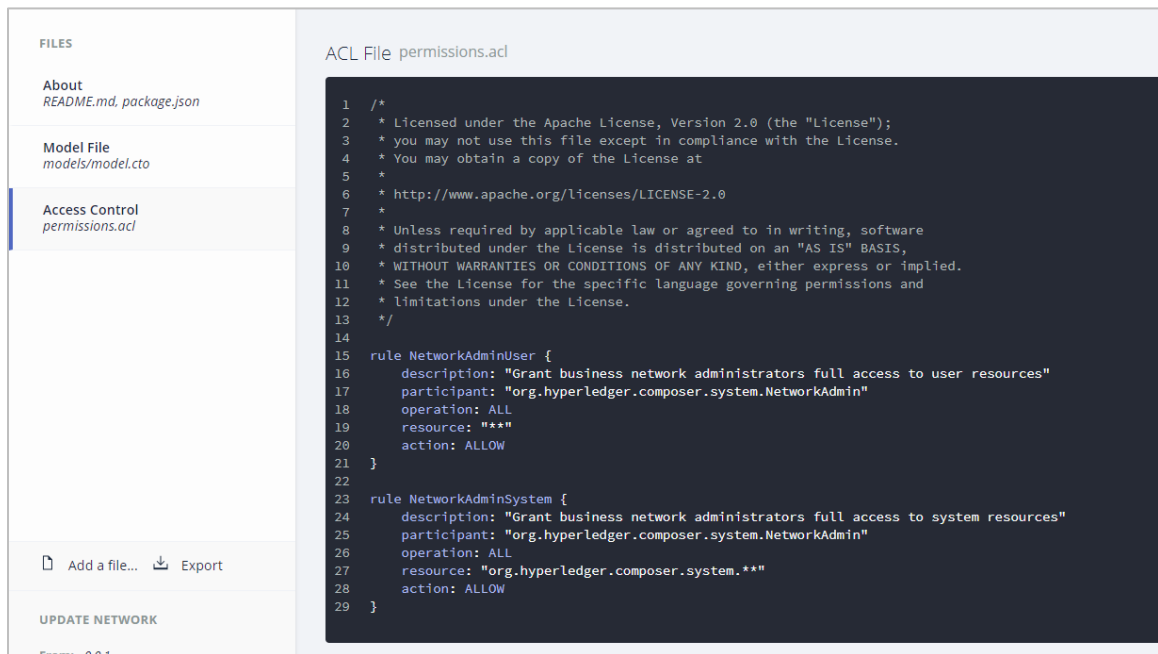


Figure 5.27 Program the model with Hyperledger modeling language

Hyperledger modeling language is used for creating the participants, assets, transactions and events of the Blockchain network. In our case, we need to create 4 participants, 2 assets and 1 transaction. As a result, the participant can create new transaction to transfer the assets to others. Moreover, we also need to write Access Control List (ACL) which are used to define the permission and action that can be taken by the role.

5.3.2 Program the Chaincode

```

func (t *SimpleChaincode) Init(stub *shim.ChaincodeStub, function string, args []string) ([]byte, error) {
    fmt.Printf("Init called, initializing chaincode")

    var A, B string    // Entities
    var Aval, Bval int // Asset holdings
    var err error

    if len(args) != 4 {
        return nil, errors.New("Incorrect number of arguments. Expecting 4")
    }

```

Figure 5.28 Program Chaincode in JAVA

Chaincode in our system is used to verify the connection between web server and Blockchain network. The verification method is based on ECC public and private key. Hyperledger offers Go language, Nodejs and java for developer to program the Chaincode. We choose JAVA to write the Chaincode for the key authentication.

5.3.3 Design the Interface

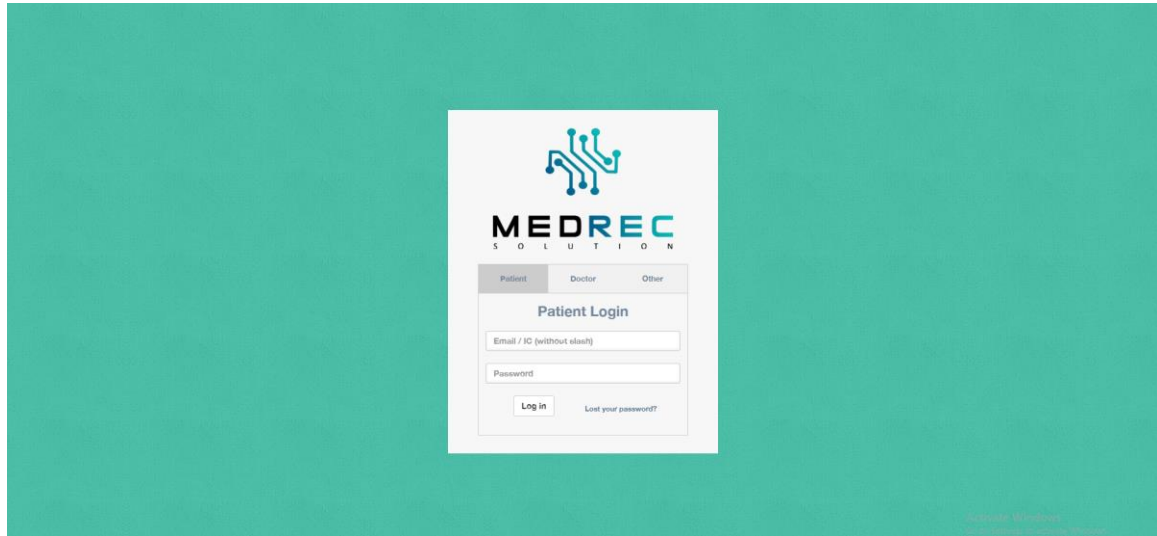


Figure 5.29 Designing MedRec interface

Design the WEB UI for 4 user roles. More than 20 pages are design for user to perform action. Besides that, Some of the libraries such as Twitter Bootstrap, JQuery and Font Awesome are used in the UI development.

5.3.4 Program functional modules

```
public static function allow_role_to_visit(){
    $redirect = 1;
    foreach (func_get_args() as $arg) {
        if($_SESSION["role"] == $arg){
            $redirect = 0;
        }
    }
    if($redirect == 1){
        echo "123";
        header("Location:". $weburl . "accessdenied.php");
        exit();
    }
}
```

Figure 5.30 Program functional module in PHP

This system is more focus on data. So we have optimize to the CRUD (Create, update, delete) of the data. These action files actually are sharing by different role of the users but they only can see the relevant information based on their user role. If user do not have sufficient permission, then they are not able to browse the webpage. Moreover, we also design the frequently used function into module form. A web page just need to plug in the module file in order to implement the functions such as ECC keypair generator, SMS gateway and QRcode generator.

5.3.5 Integrate with Public/Private Key Generator: ECC library

```
<?php
// Generate Key Pair
require dirname(__FILE__) . '/module/ecc/generate-key.php';
$keypair = ecc::genecc();
?>
```

Figure 5.31 Import PHP ECC Library

ECC module is a library published on Github. After we included the library file into our system, then we can straight away generate the keypair. As the example above, we can call \$keypair["private"] and \$keypair["public"] to display the key. We implement this function when registering new account. A unique keypair will be generate for new user and

store into their account database. For the doctor and institution, they allow to send their public key through SMS to patient so that patient can use the key to allow access for them.

5.3.6 Integrate with SMS Gateway API

```
<?php
require dirname(__FILE__) . '/module/sms/sendsms.php';
?>
```

```
$message = "Please allow access for " . $requestername . " at the link
sendsms($_POST["mobilephone"], $message);
```

Figure 5.32 Implement SMS Gateway API

The API is provided by ISMS.COM.MY. We have designed the API into a module so that we can directly send SMS to user through the sendsms() function. The SMS function is applied when new user register, doctor request record from patient and doctor send public key to patient. For the medical record request, there will include a link in SMS for the user, the user can directly approve the request by access the related link.

5.3.7 Integrate with QRCode Generator Library

```
<?php
require dirname(__FILE__) . '/module/phpqrcode/qrlib.php';
?>
```

```
<h4><u>Share by QrCode</u></h4>
<?php QRcode::png($row["publickey"], 'images/qrcode/publickeyqrcode.png-' . $row["publickey"]); ?>
" width="300px" hei
```

Figure 5.33 Implement QRCode Generator Library

The library is also an open source library published on Github. We designed it into a module so that we can call the function and pass in the value to generate a QRCode. This function is used for generate a QRCode for doctor's public key. So user can use QR Scanner to get the public key and allow access for the doctor.

5.4 System Operation

The critical part of this system is to make sure the synchronization of medical record. In this case, we have designed the system to get data from Blockchain REST API every time when user login to their own account. This can ensure that all the information that viewed by the user are up to date. For the doctor, they need to request access from user before they can add any medical record for the patient. Once the request is approved by patient, the system will retrieve the latest medical record from Blockchain network to the doctor. The request is only valid for 1 day. If the request is expired, then the doctor need to request from the patient again. If the request is approved for doctor B while doctor A is still valid for accessing the data, the system will automatically make the doctor A invalid to the medical record in order to avoid duplicate update from different doctor.

This system is also design in Object-oriented way. We have created class file for admin, patient, doctor and institution. In this case, the entity of a Class can directly use the functions provided by the Class and the object can straight a way to call the object variable to get certain information. For example, when a user login to the system, an object called \$user will be created. We are allow to get the user information from this object such as username, full name, age, user role and so on. If we want to get current role, we just need to call \$user->get_role(). As a result, this system developed in a more flexible way and easier to be maintain in future.

5.5 Conclusion Remark

We have improved the authentication mechanism and infrastructure of the application to improve the security. For the connection safety, we installed SSL Certification in order to ensure the connection between web server and users are encrypted. The SSL Certification we used is just Domain Validation for the development stage. We will upgrade it to Extended Validation for better guarantee and security in future.

Besides that, we also use a lot of library such as SMS and QRCode to improve the user experience, so that user can quickly response to the system in order to perform any action. On the other hand, we have used ECC keypair generator to generate the keypair instead on using RSA. ECC provide a faster way to generate the keypair compare to RSA. The website architecture is also design in flexible way which all the webpages are sharing the same header.php and footer.php. As a result, we can ensure the consistency of webpage and the convenience of maintain. We just need to modify one file then all the webpage will be updated to the latest version.

Chapter 6 : System Evaluation and Discussion

6.1 System Testing and Performance Metrics

System testing was conducted in order to ensure the reliability and availability of the system. The main components inside our system is the Blockchain VPS and the web server. Therefore, our system testing will based on these two components. For the first test, we use ICMP test to ensure the connectivity between the nodes. Then, we run a webpage speed test to check the time a user get into the system.

6.2 Testing Setup and Result

6.2.1 Network Connection Test

In this testing, we continuously send 1000 ICMP packet to the node and check the number of packet lost in the connectivity. We conducted this testing in 10 times. The purpose of this testing is to check the reliability of network.

Web Server

Run	Packet Sent	Packet Loss
1	1000	0
2	1000	0
3	1000	0
4	1000	0
5	1000	0
6	1000	0
7	1000	0
8	1000	0
9	1000	0
10	1000	0

Table 6.1 Web server network connection test

Blockchain VPS

Run	Packet Sent	Packet Loss
1	1000	0
2	1000	0
3	1000	1
4	1000	0
5	1000	0
6	1000	0
7	1000	0
8	1000	0
9	1000	1
10	1000	0

Table 6.2 VPS network connection test

6.2.2 System Response Speed Test

In this testing, we use a Singapore AWS EC2 node which running Google Chrome to test the loading speed of our website. We conducted the test on www.webpagetest.org.

Run	Loading Speed (s)
1	1.616
2	1.543
3	1.890
4	1.642
5	1.603
6	1.572
7	1.610
8	1.703
9	1.688
10	1.533

Table 6.3 Web page response speed test

6.2.3 Data Retrieval Speed From Blockchain

In this testing, we designed a script file on the web server. This script file was using the GET method to retrieve the data from Blockchain. So that, we get the retrieval time used for a request pass to Chaincode and retrieve the data from Blockchain back to web server. We run the test with 10 string parameters per request and 25 string parameters per request. Then, we compared the speed between these two requests.

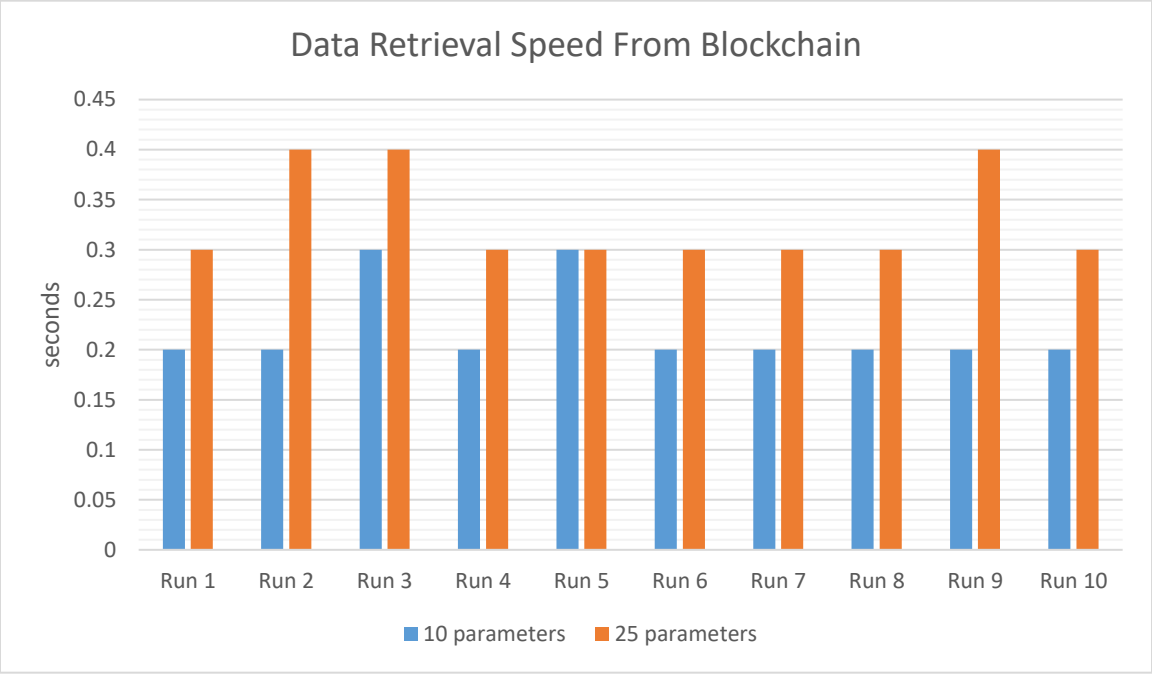


Table 6.4 Data retrieval speed from Blockchain

6.3 Project Challenges

During the designing of this system, there are many challenges we faced. The first challenge we faced is we do not have the actual medical record to build our database. For example, what is the fields that needed by the doctor to enter patient information? In this case, we tried to explore internet to try other MHR system and download a lot of sample diagnosis form for us to reference.

The next challenge is the technical issue. As we known, Blockchain is a new technology and not much implemented yet in Malaysia. In FYP 1, we tried to install an Ubuntu 16.04 in our local machine and install the Hyperledger Fabric, the Blockchain is able to run successfully. But in FYP 2 when we try to move the Hyperledger Fabric to internet, we found that the Blockchain unable to start and showing many of the errors. In this situation, we tried to open ticket and continuously communicate with the VPS service provider to find out the solution for the hardware issue.

More than that, another challenge is the data synchronization issue. Every doctor and permissioned institutions are allow to request medical records from patient. If they update the information concurrently, there will be encountered a data loss issue. In this case, we tried to lock the user request. If the user allowed another request while a request is active, then the active request will be stopped, and allow the permission for the last approved request.

6.4 Objective Evaluation

There are totally 2 objectives in this project. The first objective is to store medical records in more secure way. This objective is completed by enhanced the authentication method, data access method and the connection security. For the authentication method, we implemented the ECC key authentication and SMS authentication. For the data access method, we design the application in object-oriented way and write a lot of function to check user role before they can access to a certain page. For the connection security, we installed the SSL and Chaincode. Therefore, we can ensure the connection security from user until the Blockchain.

The second objective is achieved by synchronize the medical record and only one copy of medical record will be accessible by the network member. The medical record in web server is retrieved from the Blockchain. When doctor is updating to the record, the web server will also POST the updated contains to the Blockchain. Therefore, anyone request to the medical record will only get the up to date version. Moreover, every actions done by the network member will also be recorded. So any modification of the content will be known by everyone.

6.5 Concluding Remark

In this chapter, we have done the system testing. The system testing result is a good prove that feedback to us about the performance of our application and hardware. If the performance of server is not able to support the user, then we can plan to upgrade to a more powerful server. On the other hand, we also discuss the project challenges we faced in the development stage and evaluate the project's objective that we set at the beginning of this project.

Chapter 7 : Conclusion and Recommendation

7.1 Conclusion

In conclusion, the confidentiality and integrity is the main concern in medical sector nowadays. It may directly affected to the success rate of surgery and the outcome of researches. In this case, the Blockchain 3.0 is implemented by using Hyperledger Fabric framework to provide a secure environment for storing the medical records. The peers involved in the Blockchain are only allowed to access the specified data which are declared based on their permission and the organization they participated.

The novel idea in this proposed system is that the patients are able to control their own records and allow the request for accessing data. Besides that, the medical records of patient is unified and distributed store in Blockchain, so the doctors can retrieve it within a seconds in order to make any medical decision based on the medical record. On the other hand, the medical research institutions not need to worry for the quality and quantity of data sample anymore.

In FYP 1, the Blockchain infrastructure and environment was developed by using Hyperledger Composer. A mobile responsive web application was designed for users to review on their own medical records. In addition, the web application allows medical institutions and research institution to query data. A public ledger also created to store the sensitive data from medical institutions.

In FYP 2, the web application had been improved. Apart from this, we have integrated a lot of functions such as key pair generator and SMS Gateway. The whole Blockchain infrastructure was completely design and a fully developed web application was designed for secured medical record storing.

This project make the personal data of patients become more protected by applying the latest technology. It is also subject to replace the current method for storing medical records instead by applying Blockchain technology. In the future, the medical institutions adopted the Blockchain technology will have more advantages compared to the institutions which using the traditional method. The patients' data is able to retrieve more efficiently anywhere and anytime.

7.2 Recommendation

There are still many enhancements and improvements can be done in this project. Firstly, thumbprint authentication is more secure compared to SMS authentication. Besides that, if the web-based application can be design into mobile app. Then, it will be more convenient for patient to access their own medical information. They also can directly download the app from Google Play Store or Apple Appstore.

Furthermore, the doctor can get the medical record of patient only if the patient authorize to it. This might be caused an issue that if the patient lost the ability to authorize to the doctor due to some emergency injury, then the doctor no way to retrieve the patient information. To solve this problem, one of the solution is to allow the patient's spouse or family to authorize the access permission to the hospital.

REFERENCES

- [1] Kleinman, Jacob. (2018) What Is Blockchain?[online]. Available from: <https://lifehacker.com/what-is-Blockchain-1822094625>. [Accessed 2 April 2018].
- [2] Qi, X., Sifah, E. B., Asamoah, K. O., Gao, J. B., Du, X.J., Guizani, M. (2017). ‘MeDShare: Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain’, IEEE Access, 5, pp. 14757-14765 [Accessed 2 April 2018]
- [3] Benchoufi, M., Ravaud, P. (2017). ‘Blockchain technology improving clinical research quality’, Benchoufi and Ravaud Trials, 18(335), pp.1-4 [Accessed 3 April 2018]
- [4] Yue, X., Wang, H., Jin, D., Li, M., Jiang, W. (2016). ‘Healthcare Data Gateways: Found Healthcare Intelligence on Blockchain with Novel Privacy Risk Control’, Plenum Press New York, 40(218), pp. 1-2 [Accessed 3 April 2018]
- [5] Jie, Z., Xue, Nian., Xin, H. (2016), ‘A Secure System For Pervasive Social Network-Based Healthcare’, IEEE Access, 4, pp. 9239-9250 [Accessed 5 April 2018]
- [6] Ekblaw, A., Azaria, A., Halamka, J. D., Lippman, A. (2016). ‘A Case Study for Blockchain in Healthcare: “MedRec” prototype for electronic health records and medical research data’, IEEE, pp. 2-11 [Accessed 6 April 2018]
- [7] Matyas Danter. (2014) PHP ECC Class Library[online]. Available from: <http://www.unibia.com/unibianet/developer/php-ecc-class-library>. [Accessed 14 March 2018]
- [8] Dominik Dzienia (2013) PHP Qr Code – QR code generator, an LGPL PHP Library[online]. Available from: <http://phpqrcode.sourceforge.net/>

APPENDIX 2-Plagiarism check result

feedback studio

Xiang Yang Tian | Blockchain-based Secure Medical Record Sharing System

-- / 0

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

Blockchain-based Secure Medical Record Sharing System

BY

TIAN XIANG YANG

2

A REPORT

SUBMITTED TO

Turnitin

Page: 1 of 87

Word Count: 13357

Text-only Report

High Resolution On

Match Overview

6%

Match 1 of 2

1 Submitted to Universiti ... 2% >

2 eprints.utar.edu.my 2% >

3 Xiao Yue, Huiju Wang, ... 1% >

4 Jie Zhang, Nian Xue, Xi... 1% >

5 Qi Xia, Emmanuel Boat... 1% >

Turnitin Originality Report

Document Viewer

Processed on: 05-Apr-2019 23:56 +08

ID: 1097986135

Word Count: 13357

Submitted: 2

Blockchain-based Secure Medical Record Sharin... By Xiang Yang Tian

Similarity Index

6%

Similarity by Source

Internet Sources: 4%

Publications: 4%

Student Papers: 4%

exclude quoted include bibliographic excluding matches < 1% download print mode: quickview (classic) report

2% match (student papers from 13-Apr-2017)

Submitted to Universiti Tunku Abdul Rahman on 2017-04-13

1% match (publications)

Xiao Yue, Huiju Wang, Dawei Jin, Mingqiang Li, Wei Jiang, "Healthcare Data Gateways: Found Healthcare Intelligence on Blockchain with Novel Privacy Risk Control", Journal of Medical Systems, 2016

1% match (publications)

Jie Zhang, Nian Xue, Xin Huang, "A Secure System For Pervasive Social Network-Based Healthcare", IEEE Access, 2016

Similarity Index	Similarity by Source
6%	Internet Sources: 4%
	Publications: 4%
	Student Papers: 4%

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1



**FACULTY OF INFORMATION AND COMMUNICATION
TECHNOLOGY**

Full Name(s) of Candidate(s)	Tian Xiang Yang
ID Number(s)	16ACB01671
Programme / Course	CN
Title of Final Year Project	Blockchain-based Secure Medical Record Sharing System

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)
Overall similarity index: _____ % Similarity by source Internet Sources: _____ % Publications: _____ % Student Papers: _____ %	
Number of individual sources listed of more than 3% similarity: _____	
Parameters of originality required and limits approved by UTAR are as Follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: _____

Date: _____

Signature of Co-Supervisor

Name: _____

Date: _____



UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	16ACB01671
Student Name	Tian Xiang Yang
Supervisor Name	Dr. Lee Wai Kong

TICK (✓)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
✓	Front Cover
✓	Signed Report Status Declaration Form
✓	Title Page
✓	Signed form of the Declaration of Originality
✓	Acknowledgement
✓	Abstract
✓	Table of Contents
✓	List of Figures (if applicable)
✓	List of Tables (if applicable)
	List of Symbols (if applicable)
✓	List of Abbreviations (if applicable)
✓	Chapters / Content
✓	Bibliography (or References)
✓	All references in bibliography are cited in the thesis, especially in the chapter of literature review
✓	Appendices (if applicable)
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

<p>I, the author, have checked and confirmed all the items listed in the table are included in my report.</p> <p>_____ (Signature of Student) Date:</p>	<p>Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction.</p> <p>_____ (Signature of Supervisor) Date:</p>
--	--