

**DECENTRALIZED AUTHENTICATION SYSTEM
UTILIZING BLOCKCHAIN**

BY
WEE CHUN MING

A REPORT
SUBMITTED TO
Universiti Tunku Abdul Rahman
in partial fulfilment of the requirements
for the degree of
BACHELOR OF INFORMATION TECHNOLOGY (HONOURS) COMMUNICATIONS
AND NETWORKING
Faculty of Information and Communication Technology
(Kampar Campus)

JUNE 2022

REPORT STATUS DECLARATION FORM

Title: DECENTRALIZED AUTHENTICATION SYSTEM UTILIZING BLOCKCHAIN

Academic Session: June 2022

I WEE CHUN MING
(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:

83, Lorong Perkasa, 1, Taman
Perkasa, 13000 Butterworth,
Pulau Pinang

Gan Ming Lee

Supervisor's name

Date: 7/9/2022

Date: 8/9/2022

Universiti Tunku Abdul Rahman			
Form Title : Sample of Submission Sheet for FYP/Dissertation/Thesis			
Form Number: FM-IAD-004	Rev No.: 0	Effective Date: 21 JUNE 2011	Page No.: 1 of 1

FACULTY/INSTITUTE* OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TUNKU ABDUL RAHMAN

Date: 7/9/2022

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that **Wee Chun Ming** (ID No: **1803703**)
has completed this final year project/ dissertation/ thesis* entitled
“ **DECENTRALIZED AUTHENTICATION SYSTEM UTILIZING BLOCKCHAIN**
 ” under the supervision of **Ts Dr Gan Ming Lee** (Supervisor) from the Department of
Computer and Communication Technology, Faculty/Institute* of Information and Communication
Technology.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in
pdf format into UTAR Institutional Repository, which may be made accessible to UTAR
community and public.

Yours truly,

Wee Chun Ming
(Student Name)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled “**DECENTRALIZED AUTHENTICATION SYSTEM UTILIZING BLOCKCHAIN**” 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 : Ming

Name : Wee Chun Ming

Date : 7/9/2022

ACKNOWLEDGEMENTS

I would like to express thanks and appreciation to my supervisor, Ts Dr Gan Ming Lee and my moderator, Dr. Robithoh Annur who have given me a golden opportunity to involve in the blockchain field study. Besides that, they have given me a lot of guidance in order to complete this project. When I was facing problems in this project, the advice from them always assists me in overcoming the problems. Again, a million thanks to my supervisor and moderator.

ABSTRACT

This project is a development-based project that specific in cybersecurity area. At present time, the blockchain technology become more popular and be adopted in many types of approach and solution. The reason is blockchain offer the decentralized, integrity, immutable, security enhancement and others benefits that are many people need. At the same time, the traditional database may not secure enough anymore so the blockchain database is appeared to solve the concerns. The blockchain is suitable be used in enhance security that able to against those cyberattacks or the actions that try to damage the data records. Thus, the propose of this project is aim to produce an authentication system utilizing the blockchain database. The database will be used as public ledger but at the same time the confidentiality of the login credentials needs to be provided as well. At the meanwhile, the data integrity also needs to be ensured. Thanks to one of the blockchain unique characteristics which is immutable, this is to prevent data tampered and modification be executed on the blockchain record.

ProvenDB is a blockchain database which is MongoDB compatible database service that integrate with blockchain. Thus, precisely to say is the database that been used in this project is MongoDB which is a non-relational document-oriented database; while applied blockchain is public Bitcoin blockchain. In the end, ProvenDB is a technology that combine from these 2 mentioned technologies so this project only deals with 1 instead of 2 sides of technology at the same time. ProvenDB consist of rich and many operations on the MongoDB basis and complement with query that only can applied to blockchain. Moreover, the blockchain will play the role for responsible in store, validate, proof data. Last but not least, the blockchain also can be used to prevent some malicious attack like data tampered, data modifications, ransomware attack, record hacking and others in this project. In conclusion, this project will be utilizing the blockchain technology in order to enhance system security.

TABLE OF CONTENTS

TITLE PAGE	i
REPORT STATUS DECLARATION FORM	ii
FYP THESIS SUBMISSION FORM	iii
DECLARATION OF ORIGINALITY	iv
ACKNOWLEDGEMENTS	v
ABSTRACT	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	x
LIST OF TABLES	xv
LIST OF ABBREVIATIONS	xvi
CHAPTER 1 INTRODUCTION	1
1.1 Problem Statement and Motivation	1
1.2 Objectives	2
1.3 Project Scope and Direction	2
1.4 Contributions	3
1.5 Report Organization	4
CHAPTER 2 LITERATURE REVIEW	5
2.1 Review of used technologies for this project	5
2.1.1 Bitcoin Blockchain	5
2.1.1 Ethereum Blockchain	8
2.1.1 Hyperledger Fabric Blockchain	11
2.2 Literature review about existing project that utilizing blockchain technology in security aspect	12
2.2.1 Fromknecht, et.al. [9]	13
2.2.2 Stockburger et al. [10]	13
2.2.3 Javed et al. [11]	14
2.2.4 Lin et al. [12]	16
2.2.5 Hammi et al. [13]	18

2.2.6	Patel et al. [14]	20
2.2.7	Lim et al. [15]	20
2.2.8	Patidar et al. [16]	21
2.2.9	Truong et al. [17]	22
2.2.10	Leka et al. [18]	25
2.3	Critical Remarks of previous works	28
CHAPTER 3 SYSTEM METHODOLOGY/APPROACH		33
3.1	System Design Diagram	33
3.1.1	System Architecture Diagram	33
3.1.2	Use Case Diagram and Description	34
3.1.3	Activity Diagram	39
CHAPTER 4 SYSTEM DESIGN		51
4.1	System Block Diagram	51
4.2	Methodology Model	55
CHAPTER 5 SYSTEM IMPLEMENTATION		56
5.1	Software Setup, Setting and Configuration	56
5.2	System Operation (with Screenshot)	62
5.2.1	Start-up Page	62
5.2.2	Register an account / Signup	62
5.2.3	Login	68
5.2.4	Reset Password	79
CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION		83
6.1	System Testing and Testing Result	83
6.1.1	Recover Account	83
6.1.2	ProvenDB Query Operations	87
6.2	Project Challenges	92
6.3	Objectives Evaluation	92
6.4	Concluding Remark	93

CHAPTER 7 CONCLUSION AND FUTURE WORK	95
7.1 Conclusion	95
7.2 Future Work	96
REFERENCES	97
WEEKLY LOG	99
POSTER	105
PLAGIARISM CHECK RESULT	106
FYP2 CHECKLIST	110

LIST OF FIGURES

Figure Number	Title	Page
Figure 2.1	Bitcoin transaction concepts architecture from [3]	5
Figure 2.2	Figure 2.2: Bitcoin timestamp server concepts structure from [3]	6
Figure 2.3	Bitcoin PoW notion from [3]	6
Figure 2.4	Solution of Bitcoin PoW flaw from [3]	7
Figure 2.5	Bitcoin transaction records structure from [3]	7
Figure 2.6	Comparison of Bitcoin and traditional privacy model from [3]	7
Figure 2.7	Ethereum State Transition Function from [5]	9
Figure 2.8	Ethereum Blockchain and Mining from [5]	10
Figure 2.9	Ethereum token system implemented basic codes from [5]	10
Figure 2.10	Ethereum basic contract on registration system from [5]	10
Figure 2.11	Healthcare Identity Management conceptions from [11]	15
Figure 2.12	Process of transactions record within blockchain, cloud storage, smart contract from [11]	15
Figure 2.13	Effect of sealers with block lost on blockchain performance from [11]	16
Figure 2.14	Effect of sealers with block propagation time on blockchain performance from [11]	16
Figure 2.15	Secure Authentication System Utilizing Blockchain Architecture from [12]	17
Figure 2.16	Bubbles of trust mechanism from [13]	18
Figure 2.17	Bubbles of trust blockchain environment from [13]	19
Figure 2.18	Voting process from [16]	21
Figure 2.19	Personal data management that using traditional client-server architecture from [17]	23
Figure 2.20	Personal data management using blockchain technology	24

	system architecture from [17]	
Figure 2.21	Performance vs scalability with high workload incurred from [17]	25
Figure 2.22	System Architecture of proposed solution from [18]	26
Figure 2.23	Certificate be added time taken from [18]	27
Figure 3.1.1	System Architecture Diagram of This Proposed Approach	33
Figure 3.1.2	Use Case Diagram of This Proposed Approach	34
Figure 3.1.3.1	Activity Diagram of Overall Website	39
Figure 3.1.3.2	Activity Diagram of Register an Account	40
Figure 3.1.3.3	Activity Diagram of Recover Account	42
Figure 3.1.3.4	Activity Diagram of Login to System	43
Figure 3.1.3.5	Activity Diagram of System Login Authentication Part using Email & Password	45
Figure 3.1.3.6	Activity Diagram of System Login Authentication Part using Google Account	47
Figure 3.1.3.7	Activity Diagram of System Login Verification through Email	49
Figure 3.1.3.8	Activity Diagram of Reset Password	50
Figure 4.1.1	System Block Diagram for Sign Up	51
Figure 4.1.2	System Block Diagram for Login	52
Figure 4.1.3	System Block Diagram for Reset Password	53
Figure 4.1.4	System Block Diagram for Recover Account	54
Figure 4.2	RAD overall phases conceptions	55
Figure 5.1.1.1	ProvenDB service running on Docker Desktop	57
Figure 5.1.1.2	Connection link on MongoDB compass	57
Figure 5.1.1.3	Connected to ProvenDB local service on MongoDB compass	58
Figure 5.1.1.4	Connected to ProvenDB local service on Command Prompt	58
Figure 5.1.1.5	Create ProvenDB cloud service account	59
Figure 5.1.1.6	Create ProvenDB cloud service	59
Figure 5.1.1.7	Filled in a unique service name	60

Figure 5.1.1.8	Filled in username and 2 password input	60
Figure 5.1.1.9	Choose free plan	61
Figure 5.1.1.10	Service be created	61
Figure 5.2.1.1	Start-up/Login page	62
Figure 5.2.2.1	Register/Signup with Email & Password	63
Figure 5.2.2.2	Login Page	63
Figure 5.2.2.3	User data status 'pending' in database	64
Figure 5.2.2.4	User data status 'pending' in blockchain	64
Figure 5.2.2.5	User data status 'confirmed' in database	64
Figure 5.2.2.6	User data status 'confirmed' in blockchain	65
Figure 5.2.2.7	User data status 'valid' in database	65
Figure 5.2.2.8	User data status 'valid' in blockchain	65
Figure 5.2.2.9	Register with Google Account	66
Figure 5.2.2.10	User data status 'pending' in database	66
Figure 5.2.2.11	User data status 'pending' in blockchain	67
Figure 5.2.2.12	User data status 'confirmed' in database	67
Figure 5.2.2.13	User data status 'confirmed' in blockchain	67
Figure 5.2.2.14	User data status 'valid' in database	68
Figure 5.2.2.15	User data status 'valid' in blockchain	68
Figure 5.2.3.1	Filled in email address and password in login form	68
Figure 5.2.3.2	Login with Google Account	69
Figure 5.2.3.3	User logged in to system	69
Figure 5.2.3.4	User logged in to system during blockchain validating data status: pending	70
Figure 5.2.3.5	User logged in to system during blockchain validating data status: confirmed	70
Figure 5.2.3.6	User enter wrong password to login	71
Figure 5.2.3.7	User login with not existed account	71
Figure 5.2.3.8	QR code page	72
Figure 5.2.3.9	Download QR code	72
Figure 5.2.3.10	QR code authentication be enabled	73
Figure 5.2.3.11	QR code authentication status: true	73
Figure 5.2.3.12	QR code authentication be disabled	73

Figure 5.2.3.13	QR code authentication status: false	74
Figure 5.2.3.14	System prompt user upload QR code	74
Figure 5.2.3.15	User browse QR code image to upload	75
Figure 5.2.3.16	System retrieved QR code result from uploaded image file	75
Figure 5.2.3.17	System retrieved QR code result from PC webcam scanned user QR code	76
Figure 5.2.3.18	QR Code authentication success and logged in to system	76
Figure 5.2.3.19	User first time logged in to system	77
Figure 5.2.3.20	User account information	77
Figure 5.2.3.21	Profile page prompt user verified through email	78
Figure 5.2.3.22	User received verification link on mail box	78
Figure 5.2.3.23	Profile page prompted verification message disappeared	79
Figure 5.2.3.24	Email verified status become true	79
Figure 5.2.4.1	User data	79
Figure 5.2.4.2	Enter email address to receive reset password link	80
Figure 5.2.4.3	Reset password link sent	80
Figure 5.2.4.4	User mail box received reset password link	80
Figure 5.2.4.5	Reset password page navigated by email received link	81
Figure 5.2.4.6	Fill in 2 consistent new passwords	81
Figure 5.2.4.7	Submit email address and hashed new password to blockchain	82
Figure 5.2.4.8	Password be updated	82
Figure 6.1.1.1	User account data	83
Figure 6.1.1.2	Enter email address to receive recover account link	83
Figure 6.1.1.3	Recover account link sent	84
Figure 6.1.1.4	User mail box received recover account link	84
Figure 6.1.1.5	Reset password page navigated by email received link	85
Figure 6.1.1.6	Fill in 2 consistent new passwords	85
Figure 6.1.1.7	User deleted account related record in blockchain	86
Figure 6.1.1.8	Account recovered and navigated back to login page	86
Figure 6.1.2.1	ProvenDB original collections	87
Figure 6.1.2.2	Submit a record to blockchain	87

Figure 6.1.2.3	Submitted data be validated in blockchain	88
Figure 6.1.2.4	Find and retrieve record in _provendb_versionProofs collection	88
Figure 6.1.2.5	Insert one record in _provendb_versionProofs collection	89
Figure 6.1.2.6	Find specific record and update it in _provendb_versionProofs collection	89
Figure 6.1.2.7	Delete specific record in _provendb_versionProofs collection	90
Figure 6.1.2.8	Insert one record in ‘users’ collection	90
Figure 6.1.2.9	Find and retrieve record in ‘users’ collection	91
Figure 6.1.2.10	Find specific record and update it in ‘users’ collection	91
Figure 6.1.2.11	Delete specific record in ‘users’ collection	91

LIST OF TABLES

Table Number	Title	Page
Table 2.1	Specifications of laptop	12
Table 2.2	Comparison between the existing applications and proposed application	28
Table 3.1	Use Case Description of Register Account	35

LIST OF ABBREVIATIONS

<i>SMS</i>	Short Message Service
<i>DDoS</i>	Distributed Denial-of-Service
<i>CA</i>	Certificate Authority
<i>WoT</i>	Web of Trust
<i>PKI</i>	Public Key Infrastructure
<i>SSI</i>	Self-Sovereign Identity
<i>DID</i>	Decentralized Identifier
<i>IDM</i>	Identity Management
<i>JSON</i>	JavaScript Object Notation
<i>JWT</i>	JSON Web Token
<i>ECIES</i>	Elliptic Curve Integrated Encryption Scheme
<i>Gsk</i>	Group private keys
<i>Gpk</i>	Group public keys
<i>MAC</i>	Message Authentication Codes
<i>PPG</i>	Public Parameter Generation
<i>KGen</i>	Key Generation
<i>Enc</i>	Encryption Algorithm
<i>Dec</i>	Decryption Algorithm
<i>PoW</i>	Proof of Work
<i>PoS</i>	Proof of Stake
<i>PoA</i>	Proof of Authority
<i>PoC</i>	Proof of Concept
<i>SCA</i>	Smart Contract Authentication
<i>OTP</i>	One-time password
<i>API</i>	Application Programming Interface
<i>IDE</i>	Integrated development environment
<i>GHOST</i>	Greedy Heaviest Observed Subtree
<i>UI</i>	User Interface
<i>GUI</i>	Graphical User Interface

Chapter 1

Introduction

In this chapter, the sections that be present are the problem statement and motivation of this research, project, objectives, impact, significance and contributions, and the background information.

1.1 Problem Statement and Motivation

First of all, this proposed project is a decentralized authentication system utilizing blockchain will be used to migrate the problems of normal authentication system that equipped with normal database. First, the authentication system is not secured enough when using normal unprotected database. Traditional way of the database to store the password and other information is not secured enough to protect user information. From the reality example, attacker able to use the brute force attack to guess out the possible passwords by using user's personal information. Furthermore, there are also many types of cybercriminal attacks can easily break through the authentication system by doing malicious actions in database. For example, using SQL injection to insert malicious query in the input then after the first query be execute finish then will run following malicious query. Another case in SQL injection is cybercriminal may use "or '1' = '1' " in the same input, or case will let database define the query is correct and run it. The second problem is normal centralized authentication system will assemble or pass through all data and privacy to a central point. Normally the central point is a server or a data server that communicate with the database through a, if server protection is not strong enough then it will become collapse or unfunctional well. For instance, Cloudflare [1] state that DDoS attack is to disrupt the target server service or traffic by using flood of internet traffic or overwhelming it. Attacker control botnet to send requests to the victim to cause the network or service traffic jam. Moreover, Cloudflare [1] mentioned the attack traffic is difficult segregate from general traffic because each bot is a legal internet device.

Authentication, database, decentralized and blockchain, these three technologies are widely been using in different kinds of field. Authentication must based on database to

Chapter 1

authenticate users by comparing the provided information and data records. Decentralized is to separate the risk and data to each system point or network node to ensure the minimal damage of system crumble and high availability because there are many nodes are working in same time. Blockchain will provide the protection and increase the security of the system by the complicated hashing algorithm in every block. In order to enhance a general system security, this motivate me to assemble and utilize these technologies to give a good protection and security to a system. Another motivation of this proposed project is due to normal authentication system that using general encryption and algorithm is not strong in security level part, some of hashing algorithm method like MD5 is not efficiency for handling password. Nowadays, the hashing algorithms is been upgraded and some published new version. In order to solve problems mentioned above, therefore I will conduct the development and improve the authentication system with decentralized-based by utilizing blockchain.

1.2 Objectives

The purpose of the thesis is to propose a decentralized authentication system by utilizing blockchain. The list of main objectives is:

- To develop a login database utilizing blockchain
- To ensure available of database as a public ledger
- To provide confidentiality of the login credentials in the database
- To design an authentication system utilizing the blockchain database

This project is main to achieve for the security level be increased to a certain level that higher than normal or some existed authentication system. The focus point of this project is mainly on the blockchain part and let the blockchain characteristics such as immutability, enhanced security, distributed ledgers, etc. be applied on the whole project. The project mainly covers the authentication and identity identify part, but no the authorized or other software function and content part.

1.3 Project Scope and Direction

In order to solve the drawbacks of general authentications system and improve it. The outcome will be a high security authentication system with decentralized based by using

Chapter 1

blockchain technology. The proposed authentication system will be applied in website based. The system able to hashed the user records with blockchain in database. Hence, this is to ensure the transaction record is immutable and confidentiality of the login credentials. Nowadays, there are various types of attack that can break through the authentication system then penetrate into the main internal system. In additional, there may some malicious people try to use the system loopholes or bugs to get access into the system. As the results, this project aims to tackle the problems faced by general authentication system and enhanced the security of the system that able to defend attacks. The coverage of this projects will include all levels of users because there must be authentication system in every fields of system. The works in this project is based on the assumption that all general system with have authentication part as well in website-based platform. The proposed system needs to link with database and be decentralized to carry out authentication process to identify user's identity.

1.4 Contributions

The contribution of this project is significant to all fields of people who wish to enhance or develop the authentication system. With the decentralized authentication system that utilizing blockchain, the system owner or organizer can be helped to filter out many those identity that are unidentified, and also countercheck cybercriminals that wish to break through the first security of system which is authentication part. The decentralized identity can be easily verified by requiring access the user's sessions or devices to prove the rightful owner of the provided identity information. The identity information and authority are always based on users, so there is low risk of mishandled data and comprised. The decentralized trait able to let the system handle all of the process and no need the verify or reply from the central organizer or administrators. This is to increase the efficiency of process with ensuring the accuracy of information at the same time. Due to the blockchain stored data and signatures is immutable and constant, the verified transaction records are impossible be changed so that can ensure the originality of user provided information is consistent as before. Moreover, user can decide where, when, how the information is shared to control over their properties. They no need to rely third-party to maintain assets and also the profile is transparent with the enhanced security. Assemble those mentioned features, system administrator no need to pay much attention on the system security and identity of client to check whether the user is validated or not. In the same time, user can control over their properties and play around with their own preferred ideas.

1.5 Report Organizations

This report was separated into seven chapters. Chapter 1 introduces the project, which explains the project statements, motivations, scope, et cetera. Chapter 2 reviewed some papers on the used technologies in this project and existing project that utilizing blockchain technology in security aspect. Next, chapter 3 show the system methodology using system architecture diagram, use case diagram and activity diagram. Chapter 4 described the system design like system block diagram and methodology model. Chapter 5 show the system implementation including software setup, setting, configuration and operation with providing screenshots. Chapter 6 discuss the system evaluation consist of system testing and testing results, project challenges, objectives evaluation and concluding remark. Lastly, chapter 7 concluded this proposed system implementation, project achievements, and the possible future work of this project.

Chapter 2

Literature Review

2.1 Review of used technologies for this project

2.1.1 Bitcoin Blockchain

2.1.1.1 Nakamoto [3]

According to Iredal [2], Bitcoin is a classic decentralized based digital currency that no required control of central bank. It had consisted of four main components: blockchain, digital signature, distributed network, mining. At first, blockchain is immutable and shared public ledger on entire Bitcoin network. Iredal [2] stated that the transactions record will be stored in blockchain and is immutable because of using hash algorithm in every block. Next, the digital signature is be produced by private key to locate the owner's bitcoin wallet address. Moreover, the distributed network ensures the reliability and immutability of the blockchain. Thus, every node can store the copy of blockchain and be dispersed globally to avoid failure. The mining is using the consensus mechanism in order to decide the transactions be recorded in which blockchain, it continuously doing the computing process and get reward after done the tasks that mentioned in Iredal [2]. According to Nakamoto [3], the transactions consist of digital signatures chain which is the electronic coin. When the owner moves the coin to other by using a previous transaction hash. The next owner's public key will be adding to the end of coin. The following figure is the transaction architecture:

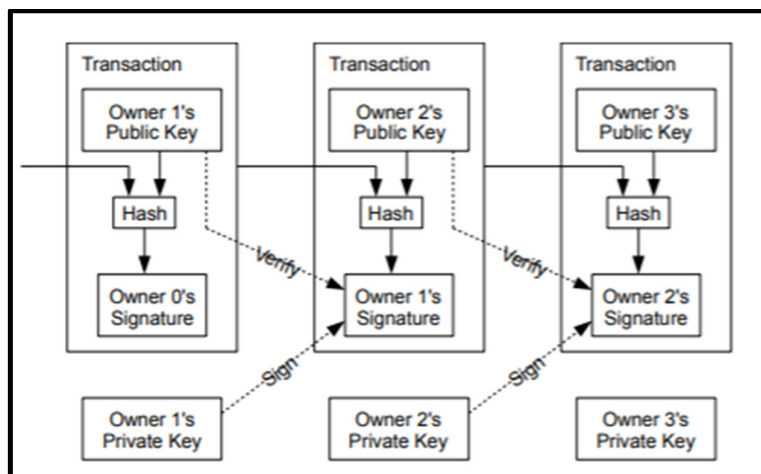


Figure 2.1: Bitcoin transaction concepts architecture from [3]

Nakamoto [3] stated there is a problem is payee cannot verify the owner did not spend double of times the coin or not. They purposed the timestamp server to solve the problem that proof each transaction time, most of the nodes already consent it was received by the first one. The timestamp server taking a hash of items block then timestamped it and publish the hash widely. The timestamp can be used to prove the data had existed on the time. Every timestamp includes previous hashed timestamp to form a chain. Then, the additional timestamp will reinforce the previous ones. The following figure shown the timestamp concept:

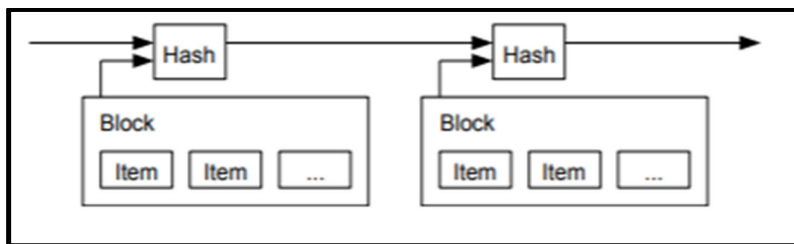


Figure 2.2: Bitcoin timestamp server concepts structure from [3]

In order to distribute the timestamp server on the basis of peer to peer, Nakamoto [3] introduced the Proof-of-Work (PoW) to solve the problem above and representation in majority decision making by the longest chain. If found that zero bits be required by block's hash, then PoW can increment a nonce in the block for the timestamp network. The block can't be changed without redo work when the CPU effort and power has been expended to satisfy PoW. The work needs to change the block when later blocks are chained then the blocks need to be redone after it. The following figure shown the PoW diagram:

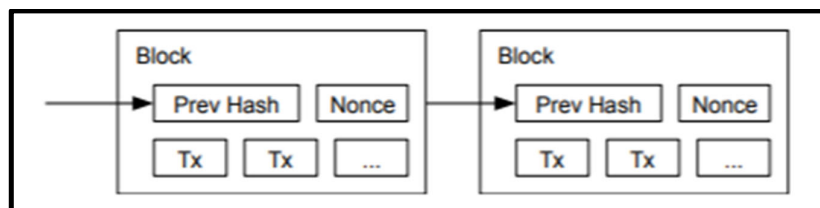


Figure 2.3: Bitcoin PoW notion from [3]

For the payment verification, Nakamoto [3] raised a solution that is simplified the process that is user only keep a replicate data of block headers from the largest PoW chain. When user convinced the longest chain then can get by network nodes query and get the linking transaction to timestamped blockchain. By the linked to chain location, user can see network node that accepted and added block after confirm network has already accepted it. The verification process is relied on the longest nodes. The network nodes are better giving

alerts when they detect invalid block when attacker can overpower the network because the network nodes can verify transactions by themselves due to the decentralized characteristics.

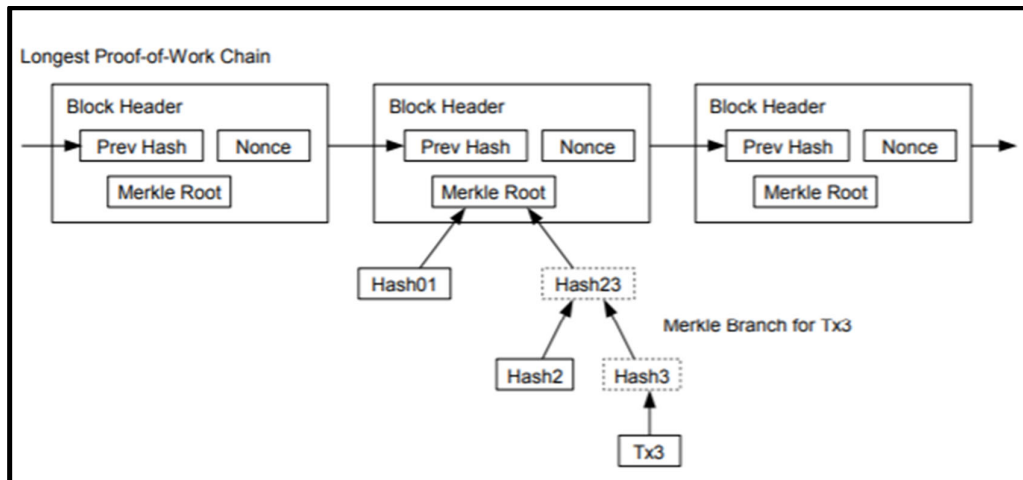


Figure 2.4: Solution of Bitcoin PoW flaw from [3]

In order to make sure the value is be split and combined, a transaction can include multiple inputs and output. The larger previous transaction or multiple inputs combine and formed amounts will provide single input. The output will be payment and return the change.

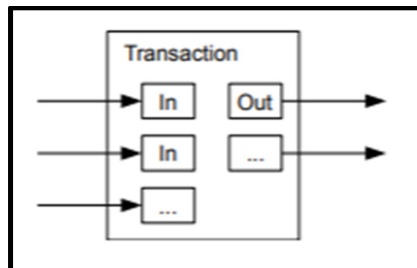


Figure 2.5: Bitcoin transaction records structure from [3]

For privacy, the proposed privacy model is different with traditional privacy model.

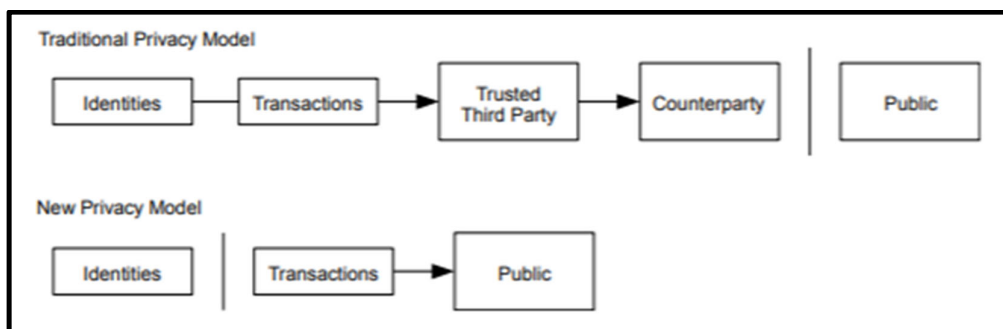


Figure 2.6: Comparison of Bitcoin and traditional privacy model from [3]

From the illustrated figure, the public is be limited to access the information which are identities, transactions, trusted third party and counterparty.

Chapter 2

According to M. Chand [4], the programming language that used to built Bitcoin blockchain is C++. C++ able to scale resource intensive applications and run the code smoothly. At the same time, E. Tarasenko [7] state that Simplicity is a programming language which higher level to develop in Bitcoin Script. Bitcoin Script is a smart contract that integrate with Bitcoin blockchain. Simplicity had integrated with Merklized Abstract Syntax Trees (MAST) then make the program into trees format and only utilize those necessary part to execute the program. In the meanwhile, the unused part will be removed to decrease the block space occupy and improve the confidentiality part.

2.1.2 Ethereum Blockchain

2.1.2.1 Wood [5]

According to Wood [5], Ethereum is an open source project, it provides blockchain solution to deploy the distribution application. Ethereum had also apply and link the smart contracts with blockchains. Basically, the general characteristic of Ethereum is inherited and same with Bitcoin but improve various types of aspects such ad stability, stability, run application on computer and others. In additional, Wood [5] had introduced smart contract in term in the patterns of contracts engineering: data feeds and random numbers. For data feeds, this type of contracts is to give access of the information from the external range within Ethereum. The timeless and accuracy of information is not guaranteed. The random numbers are be provided with a deterministic system that is impossible task naturally. But the data include block hash, timestamp, beneficiary address. In order to prevent malicious minor to connect the information and values, they used the BLOCKHASH that is hashes of previous 256 blocks as the pseudo-random numbers, and also apply a trivial solution to add several of constant values and hashed results. Through the Wood [5] introduced protocol, user can implement a node on the Ethereum network and join with other nodes in a decentralized secure operating system. The contracts can be set and suit by using autonomously enforce rules and algorithmically specify.

2.1.2.2 Buterin [6]

From Buterin [6] paper that had described some limitations and challenge of Bitcoin's script language. With the full Turing-completeness provide assist, Ethereum able to support many types of computation. Therefore, Ethereum can handle and support the state of transaction to do the improvements over the blockchain architecture. Moreover, an abstract

layer is provided that allow everyone create ownership regulation and rules, transactions formats and transition functions. By certain conditions are met, this is involved the execution of smart contracts and cryptographic rules. The Ethereum blockchain network is depend on improved GHOST protocol. It is born to solve the stale block issues in the network. The stale block is a miners group existed in a mining group that cause more overall compute power and energy than others then create centralization problems. The GHOST protocol will solve the problem by include the stales block in the longest chain. Thus, the stales block receives the 87.5% of reward, left 12.5% will left for stale block nephew, but the stales block can't become main part of blockchain anymore. The following figure is the Ethereum State Transition Function diagram:

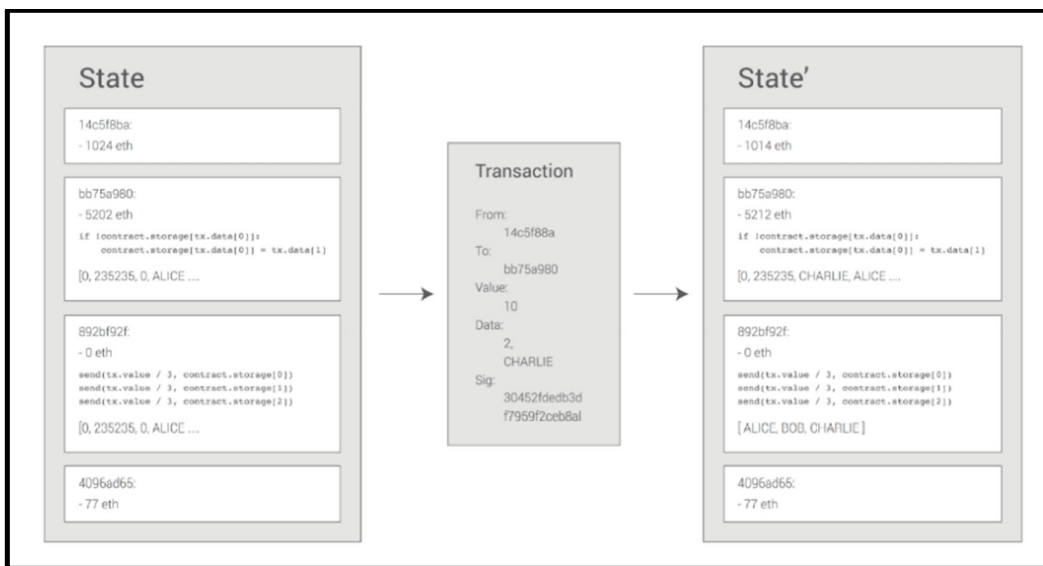


Figure 2.7: Ethereum State Transition Function from [5]

Next, Buterin [6] declared the difference of Ethereum blockchain with Bitcoin is block numbers, nonce, difficulty, transaction list, recent state and others. But for the transaction list, the previous state is applied to create a new state. The Ethereum blockchain included parent block's header Keccak 256-bit hash, roots state hashes, mining fee receipt address, transaction, block current gas limit, block transactions used total gas, nonce, timestamp in the verification. The main problem of Bitcoin network is ASIC mining eligibility. Ethereum solve the problems by using Ethash on the PoW algorithm that consists of heavy memory and less appropriate for ASIC mining. The Ethash is the alteration and further improvement from the Dagger-Hashimoto algorithm. The Ethereum network manage to process over 1 million unique transactions in 1 day and suppose to do the PoS mining paradigm based on Casper consensus algorithm. But the miner reward is based on the coin holdings amount not their

computations compare to Bitcoin. The following figure shown the Ethereum blockchain and mining structure:

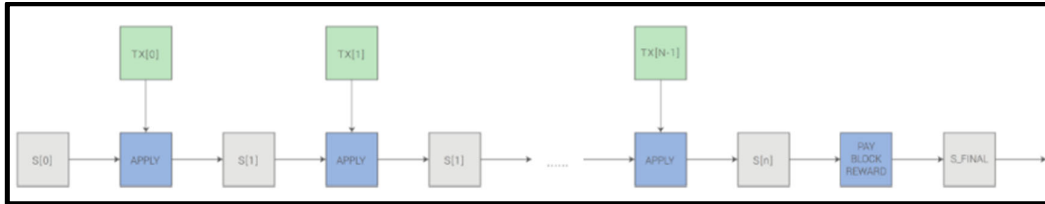


Figure 2.8: Ethereum Blockchain and Mining from [5]

Nowadays, Ethereum can be used in financial derivatives, token systems, cloud computing, identity management and etc. The token system is to implement the smart contract logic. The following figure demonstrated the token system implemented basic codes and registration system contract:

```

from = msg.sender
to = msg.data[0]
value = msg.data[1]

if contract.storage[from] >= value:
    contract.storage[from] = contract.storage[from] - value
    contract.storage[to] = contract.storage[to] + value
    
```

Figure 2.9: Ethereum token system implemented basic codes from [5]

```

if !contract.storage[tx.data[0]]:
    contract.storage[tx.data[0]] = tx.data[1]
    
```

Figure 2.10: Ethereum basic contract on registration system from [5]

E. Tarasenko [7] mentioned that Ethereum was written in Go, C++, Rust., Python, Java. The backbone of Ethereum is the runtime environment to execute the code and script that using JavaScript. Not only that, Ethereum has it own smart contract that also integrate with blockchain. The used language call Solidity that was based on JavaScript and have learn concept from Python and C++ as well. The Ethereum blockchain developer can use Solidity in high-level programming and compile it into low-level language which is machine language. Solidity usually work on Ethereum virtual machine (EVM) that developer create the smart contract by the Ethereum digital transactions.

2.1.3 Hyperledger Fabric Blockchain

2.1.3.1 Linux Foundation [8]

According to the Linux Foundation [8], Hyperledger is Linux Foundation project which handle technology frameworks such as Hyperledger Fabric, Sawtooth, Burrow and others. Hyperledger can fulfill major requirements of blockchain for business. The PoW and permissioned network is not involved in mining cryptographic problems. The mining is no reliance on cryptocurrencies when mining is not needed. It supports business the flexibility to conduct transactions transparency to specific organizations. The encryption key management and signature operations can be achieved on this blockchain. Hyperledger enable smart contract to business process, fine-grained access control and Byzantine fault tolerance (BFT) algorithm for the operation in permissioned node.

According to E. Tarasenko [7], Hyperledger Fabric blockchain was written in Java, JavaScript Golang, Python. The Golang is a language that developed by Google that is based on C programming language syntax, The Hyperledger Fabric smart contract can be using the Golang to achieve it. This language is easier to learn, user-friendly, flexibility, scalability, and fast so this can let developer to adopt in this language in a short time with same concept syntax as other programming language.

Characteristics	Bitcoin	Hyperledger	Ethereum
Founded	January 2009	July 2017	July 2015
Native currency	Yes, Bitcoin (BTC)	No	Yes, Ether (ETH)
Permission restrictions	Permission-less	Permissioned	Permission-less
Access to data	Public	Private	Public or private
Consensus	PoW	PBFT	PoW
Scalability	- High node scalability - Low performance scalability	- Low node scalability - High Performance scalability	- High node scalability - Low performance scalability
Anonymity	- Pseudonymity - No transaction data encryption	- Pseudonymity - No transaction data encryption	- Pseudonymity - No transaction data encryption
Scripting	- stark-based scripting	- High level language - Turing Complete - Scripting of chaincode	- High level language support - Turing complete virtual machine
Programming language	C++	Golang, Java	Golang, C++, Python, JavaScript, Java

Table 2.1: Comparison of Bitcoin, Hyperledger and Ethereum

2.2 Literature review about existing project that utilizing blockchain technology in security aspect

Nowadays, there are much more individuals, organizations, companies and government have done some researches, studies and development regarding to the blockchain technology. Before that, they already spend few of years to bring the blockchain technology become mature. Some applications and solutions are already suit well with this technology in the near term, and being used more and more as the technology evolves. But there are still some limitations on the authentication system such as lack of privacy and anonymity, lack of completeness and confidentiality. For the anonymity and lack of privacy, if user use a simple or general password format for the system then it can be easily predicted based on the daily activities or general info of users. For lack of completeness and confidentiality, some user's data are stored in local database. The malicious people and cybercriminal can easily comprise the database and modify the data. The popular method of database attacks has SQL injection,

privilege escalation, target unpatched database vulnerabilities and others. There are some related works that utilizing blockchain and decentralized-based:

2.2.1 Fromknecht, et.al. [9]

According to Fromknecht, et.al. [9], the developers and researcher introduce Certcoin which is public, decentralized authentication scheme that rely on NameCoin blockchain. The scheme can maintain domains public ledger and user public keys. The system consists of transparent CA and WoT, adopt the secret key as primary key that used to authenticate website. Moreover, Certcoin transaction records included the signed information which are two key pairs. First key pair is online and the secret key is used to authenticate messages that receive from or send to the server. The authentication required proof on knowledge of the key. Second pair key is offline and the secret key is be stored in secure offline place. The reason of offline is to revoke new keys if the key is comprised or security be break. Not only that, Certcoin also able to do key recovery and revocation. When user create key pair for domain, the system will require them to setup recovery system and the public key can be manually revoked with reason code. The domain transactions are executed with Bitcoin transactions to incentive minors to keep doing computing job. The Certcoin is able to self-sustaining and retrieve efficient key with the trusted key distribution mechanism to become better for performance conscious applications. After all, Certcoin had utilized public and private key pairs, resolves several inherent issues to current PKI. At the same time, it provides fault tolerance, transparency and redundancy. However, Certcoin still limit inaccessibility and require trusted third party to give external support for certain services. Not to mention that, the verification process needs large storage capacity for user device.

2.2.2 Stockburger et al. [10]

According to Stockburger et al. [10], the researchers and developers mentioned that the projects is focus on SSI and decentralized based blockchains. The proposed system is used to provide interoperable, convenience with eliminate travel card and unified identity management approach that can be applied by public transportation service providers. They derived key requirements and use Hyperledger Indy blockchain ad PoC to develop a less fidelity prototype. They utilized blockchain in mobility service, digital identities, SSI to verify the customers and other individuals' identity. The schemas and design are determined by stakeholders to generate credentials for clients. The credentials and information can be

Chapter 2

used in transactions. Furthermore, the decentralized identity management conceptual design had considered different stakeholder's requirement. The developer will develop a simple prototype to fulfill the requirements and demand of stakeholder's in making the credentials universal among them. Moreover, this proposed solution able to determine interchangeable between stakeholders, clients can customize the prototype based on requirement, eliminate the redundant information in each identity system and users have full identity information control. In reality, there have challenges when deploy and apply the system in limited internet connectivity area especially the user credentials verification process during ticket checking process. Not only that, there still have difficulty for adopting proposed systems in a multi-transport provider and stakeholder's environment.

2.2.3 Javed et al. [11]

Javed et al. [11] proposed a project which is a decentralized identity management blockchain-based for remote healthcare to distinguish and authenticate patients and healthcare providers transparently that among different eHealth domains. There are few roles in the healthcare system and database which are users, healthcare regulator, blockchain and cloud storage. The blockchain is used in security and identity management. The health ID is implemented by a consortium blockchain. Therefore, the transaction records, transaction per second, lost block number and block propagation time is to measure the blockchain efficiency and effectiveness. The decentralized IDM mostly be categorized into SSI and decentralized trusted identity, the main propose of IDM is to let the healthcare providers and patients to authenticate and identify each other.

The applications are used by patient, provider and regulator to register blockchain. The application stored public and private key pairs. Javed et al. [11] mentioned the password authentication method is using the stored private key in application of user device to distinguish users. The private key is to prove transactions and sign attestations then store in blockchain, while the public key is be applied to create an account in blockchain. Furthermore, the identification and authentication also use the health ID to carry out the methods. Healthcare providers use their practice license to prove their identity while patients can use public identification cards or documentations like national identity cards, passport, driving license and so forth to prove identity. The following below figure 2.11 is the architecture if the smart healthcare identity system:

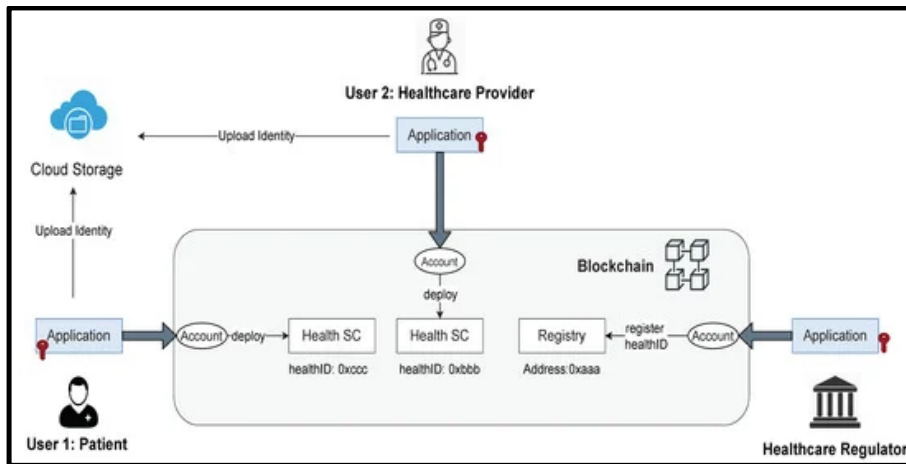


Figure 2.11: Healthcare Identity Management conceptions from [11]

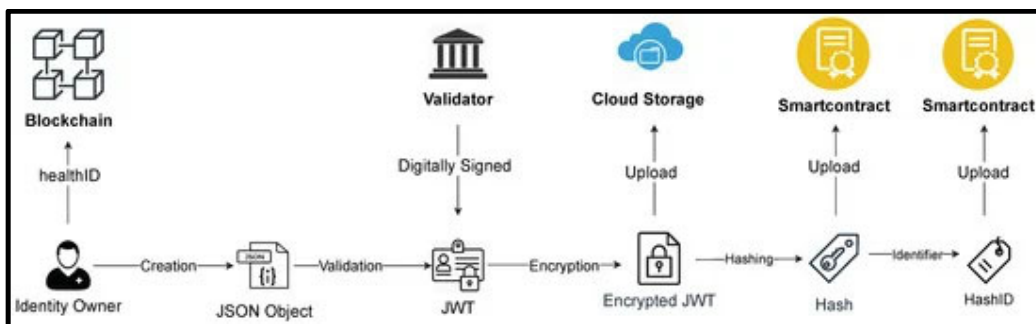


Figure 2.12: Process of transactions record within blockchain, cloud storage, smart contract from [11]

From the above figure 2.12 is the overview of the digital identity from Javed et al. [11]. The healthID create, store, manage the identity attributes and be stored in blockchain. They use JSON to store the identity attributes then sign by JWT which is standard to transmit information JSON object securely. Then owner can upload the JWT identity attributes which is encrypted over cloud service. The identity attribute will be hashed to ensure the data integrity. The unique random number will act as hash ID to identify the hash data. Thus, owner’s smart contract will upload the hash and its ID to the blockchain. In PoA, the consensus need agreement from most of the sealer (nodes that validated block) nodes. When Javed et al. perform blockchain performance with the effect of sealers, a high number of blocks lost were observed when five nodes selected more than three sealers. This situation shown as figure 2.13 below:

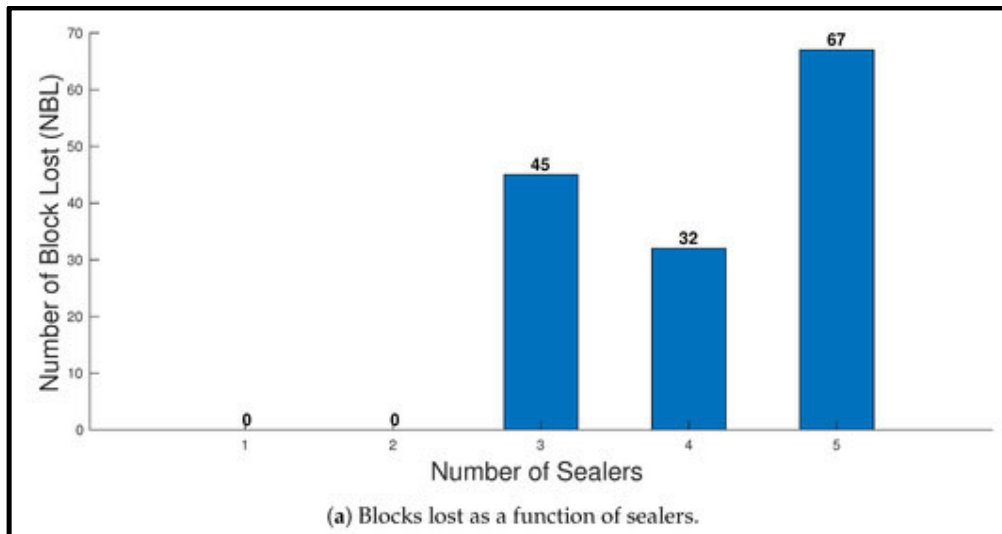


Figure 2.13: Effect of sealers with block lost on blockchain performance from [11]

At the same time for the effect of sealers on blockchain performance, the more sealers cause the more block propagation time. This means the synchronization and propagation delays will be observed and affect the performance. This is because more block needs to be released to most of the nodes within the network. This situation shown as figure 2.14 below:

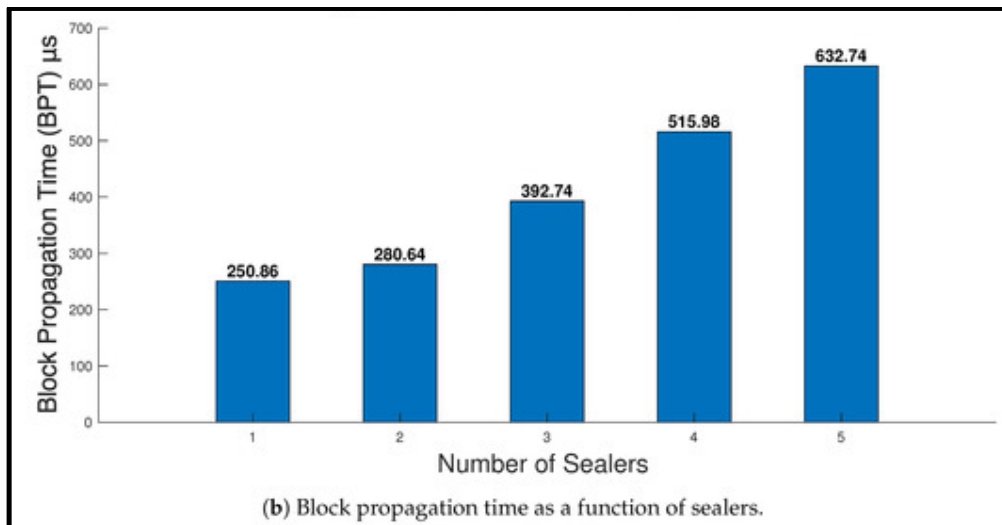


Figure 2.14: Effect of sealers with block propagation time on blockchain performance from [11]

2.2.4 Lin et al. [12]

According to Lin et al. [12], the researcher proposed a secure mutual authentication system that applied blockchain technology to be used in smart home. The proposed system able to provide privacy protection and traceability. They integrate blockchain and signatures of group to authenticate users and code in anonymous way. Due to the characteristic of

blockchain that ensure the records are immutable, the records are not easily to be modify. To achieve privacy protection of access policy, they adopt a list to revoke malicious users' authority but prevent to use the access control policy table. For the system setup, they assume family members are the group users on the system. Group manager obtain Gpk and Gsk, public parameters generator and ECIES to generate public paramaters. Group members will have own individual Gsk to sign the transactions. The home gateway will keep the Gpk to verify transaction and generate group members' key pair. They use MAC equal to MAC key multiple with message in the MAC generation function, the key is act as negotiated key and MAC is used for message authentication. The following figure is the architecture design of their system.

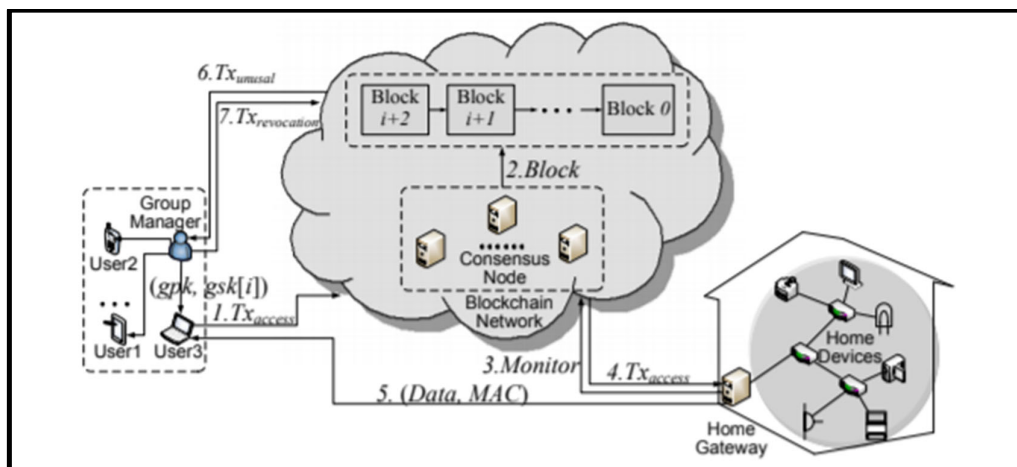


Figure 2.15: Secure Authentication System Utilizing Blockchain Architecture from [12]

From the above figure, the researchers explained that after manager of group registered, a user from the group can do requests into blockchain. The gateway monitors will request to record that stored in blockchain and show the data to users. Besides, the group manager able to trace identity from unusual transaction and revoke malicious user's authority. For the security requirements, Lin et al. [12] have described many types of demand that their proposed system achieved such as single registration, mutual authentication, resistance of hijack and resistance of various attack. The resistance of various attacks examples of they mentioned are DDoS attack, user impersonation attack, replay attack, etc. However, from the previously mentioned attack only able to resist DDoS and modification attack, other types of attack only able to detect but not resist. To show the authors' proposed system advancement, they compare it with the existing authentication protocols and stated that they achieve following features:

- Mutual authentication communication
- Anonymity
- Confidentiality
- Traceability.
- Privacy
- Revocation.

They mentioned that other blockchain-based authentication protocols have not achieved the access policy privacy and traceability features, but their system has achieved the goals.

2.2.5 Hammi et al. [13]

From Hammi et al. [13], the researchers proposed a decentralized authentication system that used blockchain technology for IoT. The main purpose of this system is to make a secure virtual zone named bubbles of trust in IoT environments. Only those devices that in same zone can communicate each other and treat other devices are malicious. This is a high-level protection for users that in same zone and inaccessible for non-member user. They use public blockchain which is Ethereum to implements smart contracts and open to any users. The communication that occurred in system is considered as the transactions and must be validate in the blockchain. For example, if A send message to B, the message will go through the blockchain first. If blockchain authenticate successful the message then B can read the messages. The following figure is the bubbles of trust mechanism:

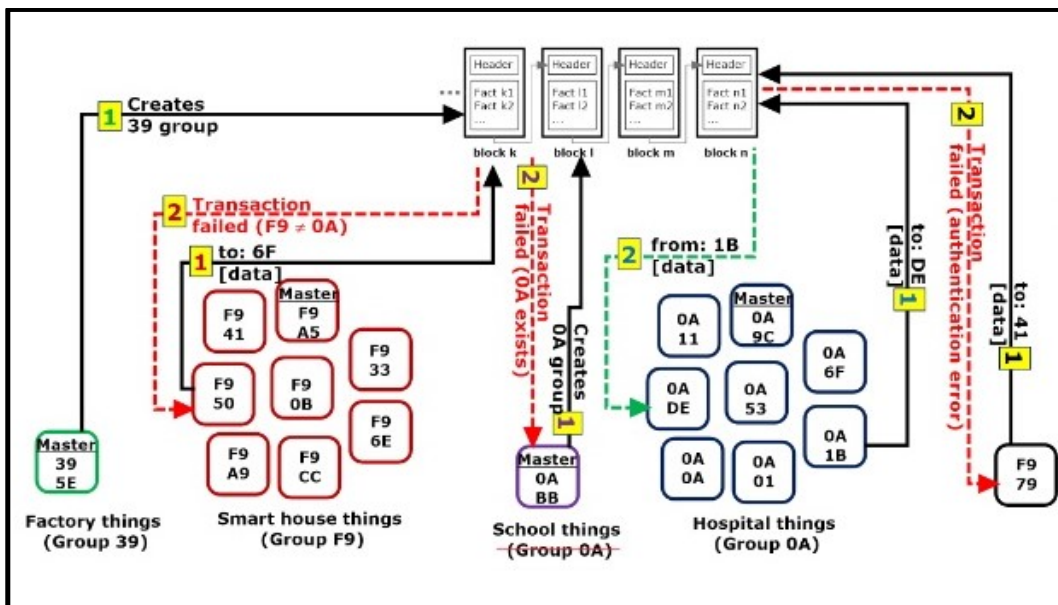


Figure 2.16: Bubbles of trust mechanism from [13]

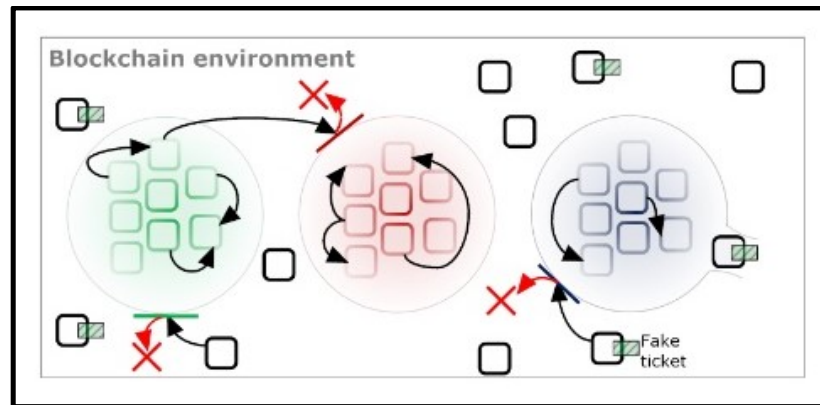


Figure 2.17: Bubbles of trust blockchain environment from [13]

Hammi et al. [13] mentioned the reason of relying public blockchain instead of private is more robust against alteration and falsification. Moreover, public blockchains are autonomous to perform their functioning and scalable to any user. The limitation that they faced are not adapt to real-time applications, needs an initialize phase and evolution of cryptocurrency rate. The system is relying on public blockchain and according to consensus protocol, the transactions need certain time (consensus process time) to be validate but IoT scenario does not much tolerate the waiting period. In order to solve this issue, they stated that can use private blockchain instead of public blockchain. For the need of initialize phase, the approach needs of service vendor. Next, the costs that are used by blockchain are depending on cryptocurrency, every second the cryptocurrency rate is changing so the costs is not constant. For the security requirements, they achieve following features:

- Mutual authentication and messages integrity
- Identification
- Scalability
- Spoofing attack protection
- Message replay protection
- Message substitution protection
- DoS/DDoS protection

At last, they mentioned that they plan to improve the system to control communication among the bubbles that be chosen. Besides, they want to perform revocation mechanism on comprised devices. They need to study and do more research to design a protocol that able to optimize the miners' number like how the selected miners can be placed in specific position.

2.2.6 Patel et al. [14]

Patel et al. [14] had proposed a decentralization web authentication system named DAAuth that use Ethereum based blockchain. The implementation of the system was using the Ethereum platform and Ethereum JavaScript API. They utilizing the blockchain advantages and clarify the blockchain as before mentioned to apply the technology in the system. Next, they use the Ethereum as decentralized network instead of using Bitcoin network. The reason of using Ethereum is this the platform can support Solidity programming language. The Turing-complete language can allow computer run the application and smart contract on the network. Moreover, the Ethereum can be a public ledger that can allow everyone to participated in network and verify the transactions.

DAAuth will require the users' Ethereum address from the network smart contract. Moreover, the user will be asked by backend to sign the produced message with AuthKey address. At frontend, user can sign the message by using the plugin metamask. While on backend, the signed message sent by user. After that, the backend compares two signatures, the frontend signatures will be stored in blockchain. As long as the signature be verified, then the verified signature will active user sessions. Patel et al. [14] had applied the smart contract that store in blockchain so the smart contract will inherit the blockchain traits. For security part, the Ethereum based 0blockchain and 160-bit hash AuthKey can prevent the cyberattacks such as Buffer Overflow, Data Tampering, Session Management Attacks, Dictionary attack, Denial of Service (DoS) attacks, etc.

2.2.7 Lim et al. [15]

According to Lim et al. [15], they purposed a decentralized blockchain-based authentication system named AuthChain. They apply the Ethereum blockchain in their mobile app authentication system. They realize that current digital authentication has few issues such as third party require significant trust, user must trust the website so cause the data be collected by malicious people. They feel that this is not secure and the data may be use in profiling, exploitation and data mining. They aim to utilize AuthChain to improve current system authentication with trustless authentication, one account, not using password for login, no storing for user credential data, immutable, flexible and resilient authentication. The user data will be hashed by SHA256 and transactions will keep in the blockchain. Moreover, the third-party user requires to scan QR code or through API online service to authenticate themselves. Their solution can authenticate user from public platform, perform

proof of user authentication and complete subsequent authentication for third-party. Not only that, they apply Ethereum smart contract by using Solidity programming language. This is used to identify validation party and allow them to authenticate themselves without accessing data itself.

2.2.8 Patidar et al. [16]

In 2019, Patidar et al. [16] proposed a decentralized e-voting portal using blockchain. Their goal is to aim for prevent data be theft by malicious people or cybercriminals. As equally important, the data will be stored in the public blockchain so anyone can have a copy for that, this is to maintain the consistent and ensure the immutability. The blockchain that be implemented in this project is Ethereum. For the registration process, voter and candidate registration must be completed ahead of time in order to complete the procedure. Before create the user accounts, the system needs to ensure the identity is verified. After the identification verification, eligible users should be identified and authenticated by authorized users by using the token. For the voting token, each user can only use the token to vote once. The token will be prevented to repeat use by blockchain's verification procedure. As a result, nobody can vote for other times and only able to vote only once. The elections are not overseen by a central authority because blockchain-based e-voting system is decentralized.

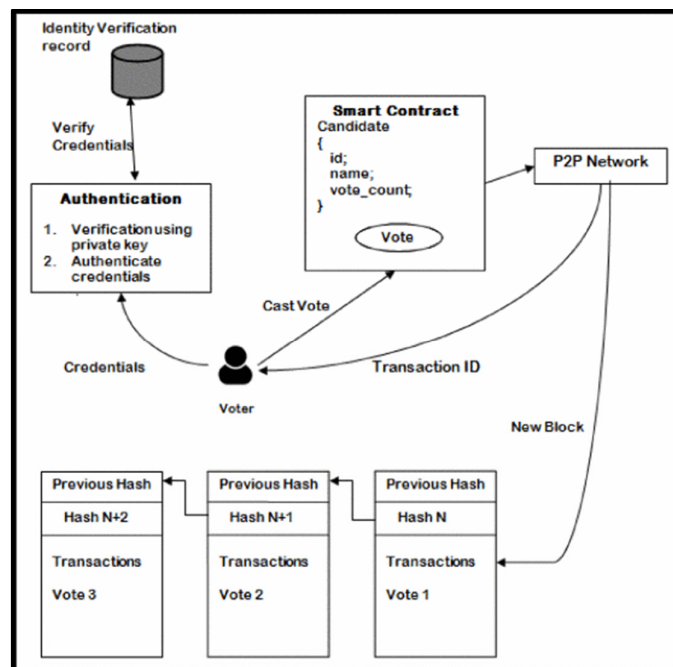


Figure 2.18: Voting process from [16]

Chapter 2

The voting process is depicted in the diagram above. To begin with, user needs an Ethereum account then use the credentials to login to the system. The system uses the user's provided credentials and private key to compare the data with the database in order to identify verification records. When the user casts a vote, the smart contract will store the user information and subsequently executes the transactions and agreements to the peer-to-peer network. After then, the network will provide the user with a transaction ID. Within a few minutes, the network's nodes will review and validate the user, and the status may show as submitted or pending. If the data is correct, it will be put in a new block and added to the blockchain; the block will have a unique hash and store the previous block hash.

According to Patidar et al. [16], the following are some of the benefits of this blockchain-based e-voting system:

- Votes are protected by cryptography.
- Votes are unchangeable and tamper-proof after they've been saved.
- It protects the privacy and anonymity of voters.
- It has the potential to increase efficiency and speed up outcomes.
- It improves the system's transparency and clarity.
- It avoids ambiguity caused by incorrect or confusing choices on paper ballots.
- The voting results are open to public scrutiny.
- The use of an electronic voting system may increase active voter engagement.

Besides with the benefits, there still have some challenge and difficulties of the system. The continuous broadband access and bandwidth is a challenge for the system. During the user's authentication vote's validation, the blockchain will require much of energy and also for the nodes of the network. For Ethereum, the intermediary token called gas which used to pay for computational effort performed in the execution of a smart contract or for particular transactions. Gas can be purchased by Ether which is denoted as "ETH" for cryptocurrency exchange. Due to the exchange rate is different in every second, so the consumed rate is different when applying the proposed system solution.

2.2.9 Truong et al. [17]

According to Truong et al. [17], they mentioned the traditional data management is implemented under a centralized client-server architecture as shown in Figure 2.19. This

architecture is leveraging a delegated authentication and authorization server. When data owner wishes to access third-party services, the data owner needs to be authenticated and grants the access. After that, the third-party will get the token as permission proof from the centralized authentication server then can access to the data from the resource server. This centralized strategy on this approach is reveals a lack of openness and trust. Due to the fact, the service provider has the complete control of the data. At the same time, the authentication server is the only point to authenticate incoming users and request. The system diagram is shown as figure below:

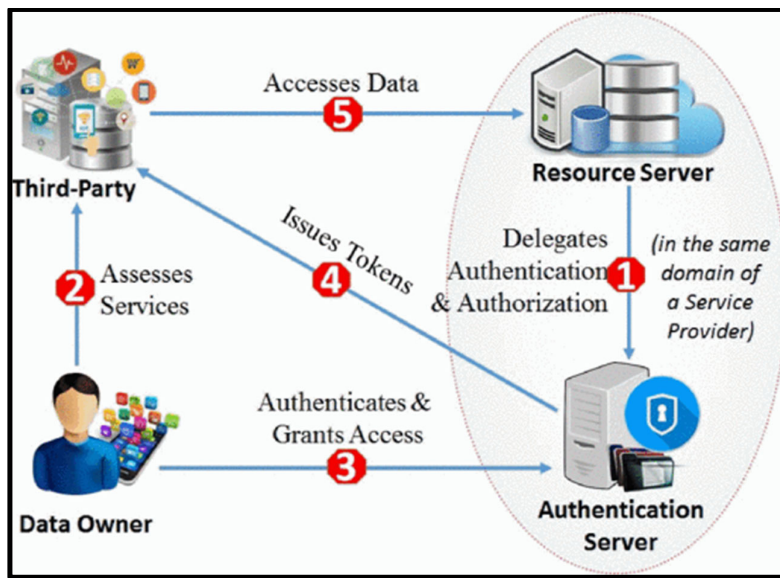


Figure 2.19: Personal data management that using traditional client-server architecture from [17]

In order to solve the problems of the centralized data management that mentioned above, they proposed a solution for to make the personal data management become decentralized. The suggested method relies on a blockchain serving two functions: (i) act as delegated server to handle authentication and authorization; and (ii) a logging system that is immutable. The blockchain provide an access token as "proof of permission" to indicate that the third-party already be authorized to access the resource server data. However, there still have similar point with traditional client-server systems which is the third-party uses the access token in API requests to access the data. The figure 2.20 below is the system architecture and procedure of this proposed solution:

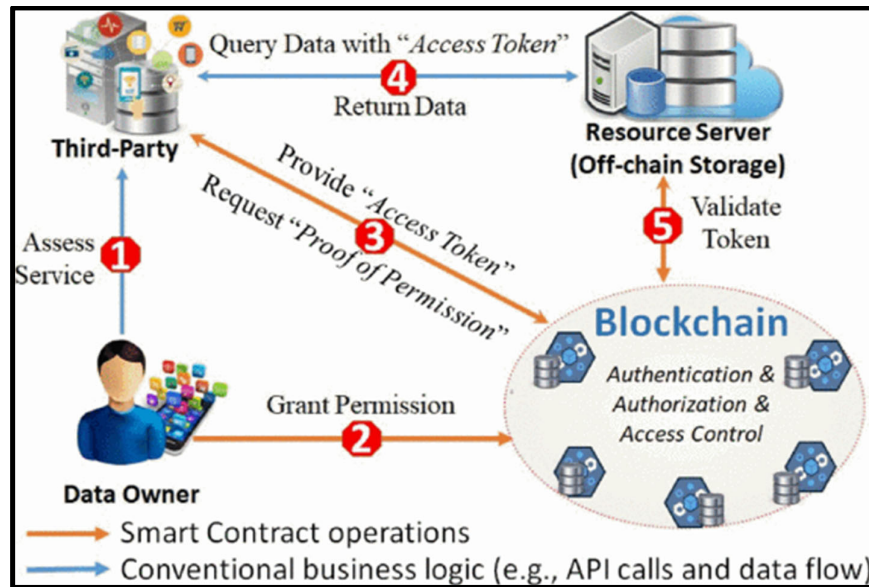


Figure 2.20: Personal data management using blockchain technology system architecture from [17]

As illustrated in figure 2.20 above, the procedure is start from when data owner request to access the third-party service (step-1) then the third-party request to access the data owner's information that be stored in resource server or database. After data owner grants the consent to third-party by bring an Access Control Transaction to the blockchain (step-2). Shortly, the blockchain authenticates and executes the Access Control Transaction, then modified the distributed ledger's access control policy to reflect the updated consent. The blockchain does not directly obtain off-chain data or data be delivered to the requester, in the other way it offers an access token as "proof of permission" (step-3) and allows the third-party to query data directly from the resource server using the access token that issued by blockchain (step-4). If the request is valid, resource server confirms the data query request by containing the access token (step-5) before returning the desired data. The truthfulness of the resource server assumption is critical in ensuring that the resource server actually validates an access token is authentic or not and being used by a matching authorized party.

Truong et al. [17] proposed solution able to solve the high expensive scheme due to they use the lesser complicated scheme encryption compare to existed approach. Their access token is hash of parties' ID sum up with the block hash. Subsequently, the access token able to be include with query data so can access the resource server by having a permission of proof. The token will be issued to blockchain network if it fulfills the policy of data usage that stored in ledger. Not only that, they also keep on the track to balance the access control

Bachelor of Information Technology (Honours) Communications and Networking
Faculty of Information and Communication Technology (Kampar Campus), UTAR

and the privacy also. This is to make sure everyone can access the information legally and also keep the user information in a protected layer as well. The feasibility of this proposed approach is successfully illustrated by the user case of trial clinical management. Nonetheless, the findings indicate that the system only appropriate for small size of services and business. This is because the system performance is primarily reliant on the underlying network of blockchain rather than top-built application. For this reason, the blockchain require much of energy source and the nodes need the reward as computational power, so the system performance and scalability can't be leveraged.

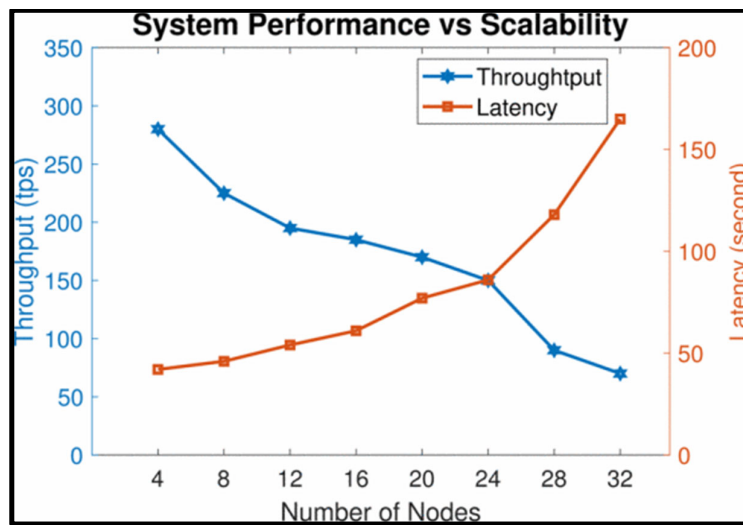


Figure 2.21: Performance vs scalability with high workload incurred from [17]

As the figure 2.21 shown above, the greater number of nodes, the higher latency but lower performance. This can prove that power and energy source are the concern and problem for this proposed solution.

2.2.10 Leka et al. [18]

In May 2022, Leka et al. [18] proposed IPFS (Interplanetary File System) - Blockchain based Authentication/Management System to enhance the ownership trustworthiness and integrity of certificates. They name their prototype as BCert and main function is attaching the certificate and related documents into blockchain network. Therefore, the certificates and documents can be checked and verified then stakeholders can access it without worrying the potential threat and damage occurs. The BCert main goal is to achieve fast, efficient and secure during the deploy and verification process. Simultaneously, the IPFS will store the encrypted certificates.

Next, there will be 3 roles: issuer institution, accreditor, employer in Leka et al. [18] proposed project during implementation. The main task of issuer institution is to issue certificates that including student’s information; The accreditor is in charge of insert/remove universities certificates validation; The employer is responsible to verify the certificate whether it is valid or not then issue verified and validated title/employment information. Eventually, a university can be accredited using the accreditation body's private key. At the meantime, the university issuer uses the private key to sign the certificate to make sure that it was granted by a reputable organization. The certificate’s authority will be checked by a prospective stakeholder using the distinctive serial number of the graduate.

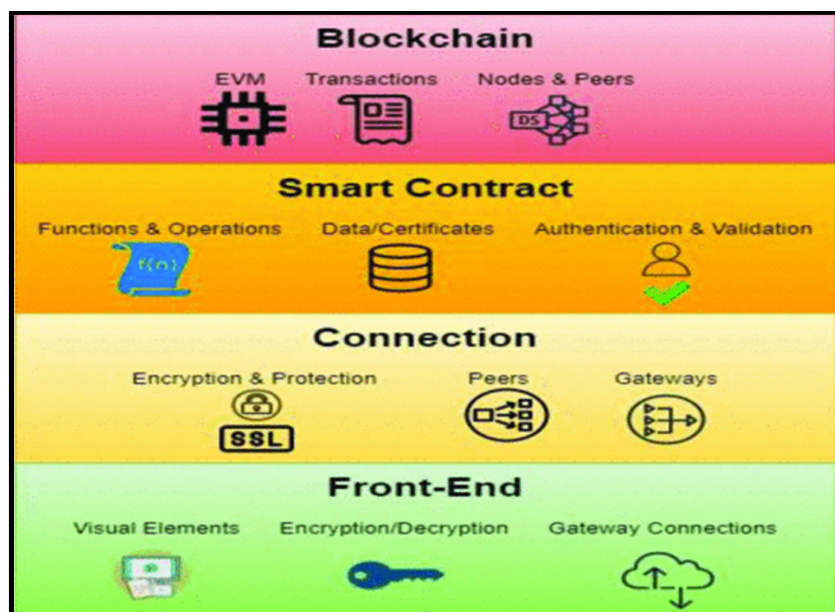


Figure 2.22: System Architecture of proposed solution from [18]

As figure 2. Shown that the data will be retrieved from front-end then send to IPFS and blockchain network through the secured connection and gateways by using encrypted private key. The gateways will separate the information to network nodes or peers. The system implementation platform is Ethereum and smart contract is using Solidity programming language then the certificates will be stored in IPFS.



Figure 2.23: Certificate be added time taken from [18]

Next, the researcher had performed BCert system performance testing. The testing will focus on Ethereum gas usage, timestamp for every transaction and evaluated by computational resources amount to mine intermediate transactions and write, upload, update, access required resources. From the figure 2. shown that the time taken needed for certificated be added. As the result, when more certificates be added in same time, the more time taken needed. Moreover, the researchers test out they can deploy 75 certificates in 15 seconds at the same time.

At the end, the researchers able to achieved several goals during BCert prototype development:

- Feasible user-blockchain interaction.
- Sustainable way of certificates deployment
- Overview on consumed resources
- Able to access certificates through multiple devices

2.3 Critical Remarks of previous works

Scheme	Proposed type	Blockchain	Network	Achievements	Difficulties/ Challenges
CertCoin: A NameCoin Based Decentralized Authentication System. (Fromknecht et.al. [9])	Open source	Hyperledger Indy	Permissioned	<ul style="list-style-type: none"> - Utilized public and private key pairs - Resolve Public Key Infrastructure issues - Fault tolerance, transparency and redundancy 	<ul style="list-style-type: none"> - Need trusted third party - Limit accessibility - Requires large storage capacity of user device to perform verification
Blockchain-Enabled Decentralized Identity Management: The Case of Self-Sovereign Identity in Public Transportation. (Stockburger et al. [10])	Government	Hyperledger Indy	Public Permissioned	<ul style="list-style-type: none"> - Utilizes self-sovereign identity building blocks - Low fidelity prototype - Determine interchangeable between stakeholders - Prototype can be customized based on clients - Eliminate the redundant information in each identity system - Users have full identity information control 	<ul style="list-style-type: none"> - Offline user validation during ticket checking process - Scalability - Adopting proposed systems in big amount transport provider and stakeholder's environment

Chapter 2

<p>Health-ID: A Blockchain-Based Decentralized Identity Management for Remote Healthcare. (Javed et al. [11])</p>	<p>Company</p>	<p>Ethereum</p>	<p>Private</p>	<ul style="list-style-type: none"> - One HealthID able to identify all identities - Health care provider – practice licence - Patients– IC, Driving Licence, Passport - Use any official information to proof identity - Ensured user’s information traceability 	<ul style="list-style-type: none"> - Performance - System propagation and synchronization delays - Lost block if over the sealers number limits
<p>HomeChain: A Blockchain-Based Secure Mutual Authentication System for Smart Homes. (Lin et al. [12])</p>	<p>Start-up</p>	<p>Bitcoin</p>	<p>Private</p>	<ul style="list-style-type: none"> - Only trusted members and within the same trust zone then can get access to main system - Able to detect malicious activities. - Achieved: Mutual authentication communication, Anonymity, Confidentiality, Traceability, Privacy, Revocation 	<ul style="list-style-type: none"> - Detect but can’t resist the cyberattacks: User impersonation attack, Man-in-the-middle attack, Replay attack - Only suitable in small network
<p>Bubbles of Trust: A Decentralized Blockchain-based Authentication System for IoT. (Hammi et al. [13])</p>	<p>Open source</p>	<p>Ethereum</p>	<p>Public Permissioned</p>	<ul style="list-style-type: none"> - Only same trust zone can communicate each other - Achieved: Mutual authentication and messages integrity, Identification, Non repudiation, Scalability, Cyberattacks 	<ul style="list-style-type: none"> - Not adapted on real time applications - Need an initialization phase - Alteration of cryptocurrency rate

Chapter 2

<p>DAuth: A Decentralized Web Authentication System using Ethereum based Blockchain. (Patel et al. [14])</p>	<p>Start-up</p>	<p>Ethereum</p>	<p>Public</p>	<ul style="list-style-type: none"> - Cost saving, user control for information, tamper resistant - Apply smart contract for the interface security - Able to against cyberattacks 	<ul style="list-style-type: none"> - Evolution and alteration of cryptocurrency rate - Acceptance of public especially in social media platform
<p>AuthChain: A Decentralized Blockchain-based Authentication System. (Lim et al. [15])</p>	<p>Start-up</p>	<p>Ethereum</p>	<p>Public Permissioned</p>	<ul style="list-style-type: none"> - Trustless authentication. - Not using password for login - Identify third party with Ethereum Smart. - Contract in terms of QR code or API services 	<ul style="list-style-type: none"> - Scalability - Performance
<p>Decentralized E-Voting Portal Using Blockchain. (Patidar et al. [16])</p>	<p>Start-up</p>	<p>Ethereum</p>	<p>Public Permissioned</p>	<ul style="list-style-type: none"> - Votes are protected by cryptography, unchangeable and tamper-proof - Improves the system's transparency and clarity - Prevent ambiguity caused by incorrect or confusing choices on paper ballots. - The voting results are open to public scrutiny 	<ul style="list-style-type: none"> - The continuous broadband access and bandwidth problems. - Blockchain and nodes require much of energy - Evolution and alteration of cryptocurrency rate

Chapter 2

Blockchain-based Personal Data Management: From Fiction to Solution. (Truong et al. [17])	Open source	Ethereum	Private	<ul style="list-style-type: none"> - Solve the high expensive scheme that using more complicated encryption - Traceable for access control and data - Within protected layer, more accessibility 	<ul style="list-style-type: none"> - Blockchain and nodes require much of energy - Performance - Only suitable for small network and business
Towards an IPFS-Blockchain based Authentication/Management System of Academic Certification in Western Balkans. (Leka et al. [18])	Start-up	Ethereum	Public	<ul style="list-style-type: none"> - Sustainable way of certificates deployment - Able to access certificates through multiple devices - real time online verification, confidentiality, authentication, and revocation 	<ul style="list-style-type: none"> - Blockchain and nodes require much of energy - Can't afford too much of certificates deployment in same time - Alteration of cryptocurrency rate
My Proposed solution	Start-up	Bitcoin	Public	<ul style="list-style-type: none"> - high data transparency - redundancy - immutability in data - able to recover back data 	<ul style="list-style-type: none"> - Performance - System robustness

Table 2.2: Comparison of Decentralized Blockchain Based Authentication Schemed with My Proposed Solution

As reviewed in section 2.1, the blockchain based able to provide several benefits is various type of proposed solution and aspect. On the other hand, the alteration of cryptocurrency rate and energy resource still is a challenge and inefficient enough to serve the public. But not to deny, the blockchain is a good option and modern technology to solve

Chapter 2

the security concern and issues. This become the main reason of people that proposed those approach by utilizing blockchain technology.

From the table 2.2 above, my proposed solution able to provide high data transparency, redundancy and immutability. This is thanks to blockchain technology offer the benefits. In addition, the blockchain characteristic is much suitable in the security aspect and able to against some of the cyberattack. However, the performance still is a challenge due to applied public blockchain. The reason is there are much many network nodes to conduct mining and validating transactions for the public blockchain. Therefore, the needed time is much more compare to private network blockchain. Besides that, the system robustness is not enough strong yet due to the system has just started developed. There are lots of part need to be improve, this may need some time more for the future work.

Chapter 3

System Methodology/Approach

3.1 System Design Diagram

3.1.1 System Architecture Diagram

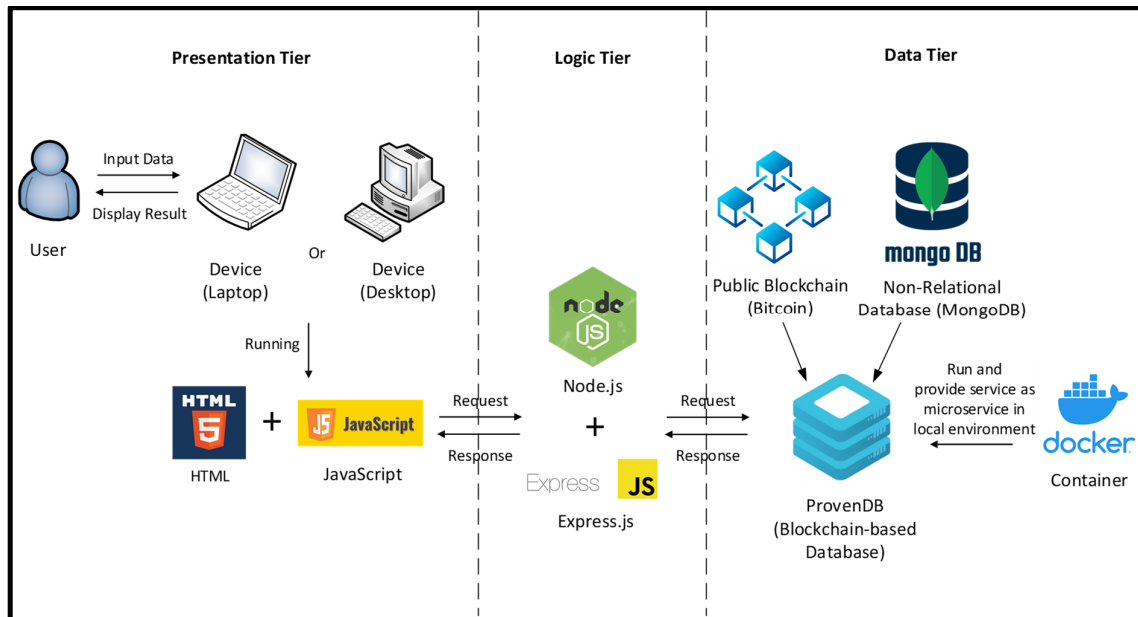


Figure 3.1.1: System Architecture Diagram of This Proposed Approach

The figure above shown that the system architecture has been separated in 3 tier architecture which are presentation, logic and data tier. In the presentation tier, the device is running the system which is the frontend is programming on HTML + JavaScript. Therefore, user can input data for the operation on device and the device able to display the related data and result. The data that has been input on the system will be sent to logic tier which is handle by Node.js + Express.js. The Express.js will handle the website session and the data route then passthrough to the Node.js to process the data to decide whether is signup, authentication, reset password or recover back the deleted account. The process will be based on CRUD (Create, Read, Update, Delete) operation and send the request to data tier. The data tier components include the ProvenDB which consist of MongoDB and public Bitcoin blockchain. The ProvenDB service can be run on the Docker container to host the microservice or use the ProvenDB cloud database service. After the request is sent to the

ProvenDB, the data will be process in MongoDB and blockchain then response back to logic tier, after that the data will be passed to presentation tier to display the results.

3.1.2 Use Case Diagram and Description

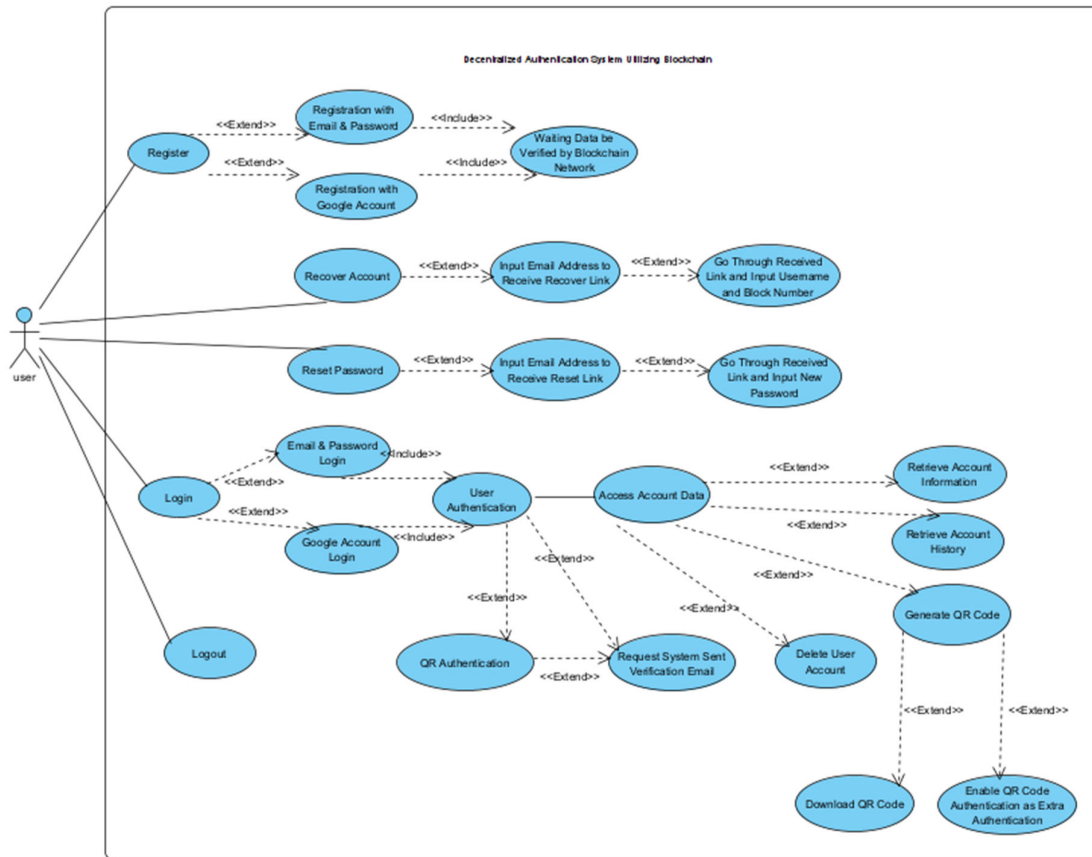


Figure 3.1.2: Use Case Diagram of This Proposed Approach

Use Case Description

Use Case Name	Basic Flow @ Happy Path	Alternate Flow @ Alternate Path	Exception Flow @ Exception Pathway
Register with email & password	- User key in the username, email address password to register an account.	- User need to ensure email and password are valid. - User need to wait the credential data (email address & password) be submitted to blockchain.	- User key in invalid data. - User 2 password input doesn't match.
Register with Google account	- User key logged in own Google account.	- User need to wait the credential data (email address & Google ID) be submitted to blockchain.	- User didn't have a Google account. - User didn't log in the Google account in browser.
Recover Account	- User filled in email address and request system send link with token to user mail box.	- User need to fill in username and block number as security code to recover back previously account.	- User input invalid email address. - User unable received the link with token. - User filled in invalid block number. - User 2 block number input not match.

Chapter 3

<p>Reset Password</p>	<p>- User filled in email address and request system send link with token to user mail box.</p>	<p>- User need to fill in new password to replace old password with new password.</p>	<p>- User input invalid email address.</p> <p>- User unable received the link with token.</p> <p>- User 2 block number input not match.</p>
<p>Login with email & password</p>	<p>- User filled in email & password to login.</p>	<p>- User need to ensure email and password are valid.</p> <p>- User need to ensure account is registered and the data is existed in database.</p> <p>- User need to ensure data proof is submitted to blockchain.</p> <p>- User need to ensure data proof is existed in blockchain.</p> <p>- User need to ensure the data proof status already be validated by blockchain network.</p>	<p>- User input invalid email address and password.</p> <p>- User account data not existed in database</p> <p>- User data proof not existed in blockchain</p> <p>- User data proof status is not validated.</p> <p>- User data proof status is failed after credential data submitted to blockchain network.</p>

<p>Login with Google account</p>	<ul style="list-style-type: none"> - User login with Gmail account that has logged in on browser. 	<ul style="list-style-type: none"> - User need to ensure Gmail account that has logged in on browser. - User need to ensure account is registered and the data is existed in database. - User need to ensure data proof is submitted to blockchain. - User need to ensure data proof is existed in blockchain. - User need to ensure the data proof status already be validated by blockchain network. 	<ul style="list-style-type: none"> - User Google account not logged in on browser. - User account data not existed in database - User data proof not existed in blockchain - User data proof status is not validated. - User data proof status is failed after credential data submitted to blockchain network.
<p>Verified account through user email</p>	<ul style="list-style-type: none"> - User request system send verification link with token to user email account mail box. 	<ul style="list-style-type: none"> - User need to ensure the email account is valid. - User need to check whether received the verification link or not. 	<ul style="list-style-type: none"> - User didn't request system send the verification link. - User email account is invalid. - User didn't receive verification link. - User didn't click the link.

Chapter 3

Retrieve account information	- User logged in to the account.	- User can delete whole account by simple click to the delete button.	- User didn't log in to the account
Retrieve account history	- User logged in to the account.	-	- User didn't log in to the account.
Download QR Code	- User click generate QR Code	- User can download the QR Code. - User can enable the extra authentication level with the QR Code.	- User didn't log in to the account.
Extra authentication level (QR Code)	- User input QR Code and compare the proof ID result with blockchain result proof ID.	- User need to input valid QR Code. - User need to ensure the credential data existed in blockchain.	- User input invalid QR Code. - User data proof not existed in blockchain.
Logout	- User logout from the system.	-	-

3.1.3 Activity Diagram

Bachelor of Information Technology (Honours) Communications and Networking
Faculty of Information and Communication Technology (Kampar Campus), UTAR

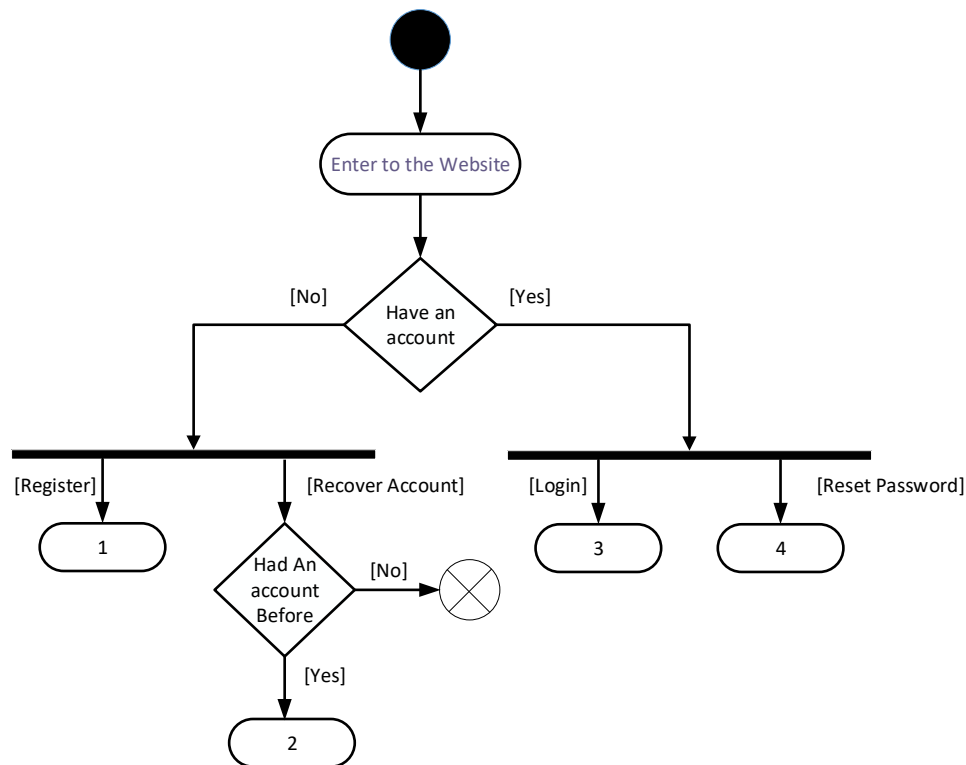


Figure 3.1.3.1: Activity Diagram of Overall Website

From the figure above shown that the process is start from when user enter the website, user have 4 options which are register (activity 1), recover account (activity 2), login (activity 3), reset password (activity 4) to choose and decide the further action. First of all, used need to base on whether already have an account or not. If user currently not have an account then can register an account. However, if user already had an account before then can try to recover back the account. On the other hand, if user have an account then user can login to the system. If user want to change account password or already forgot account password, then user can choose to reset the password.

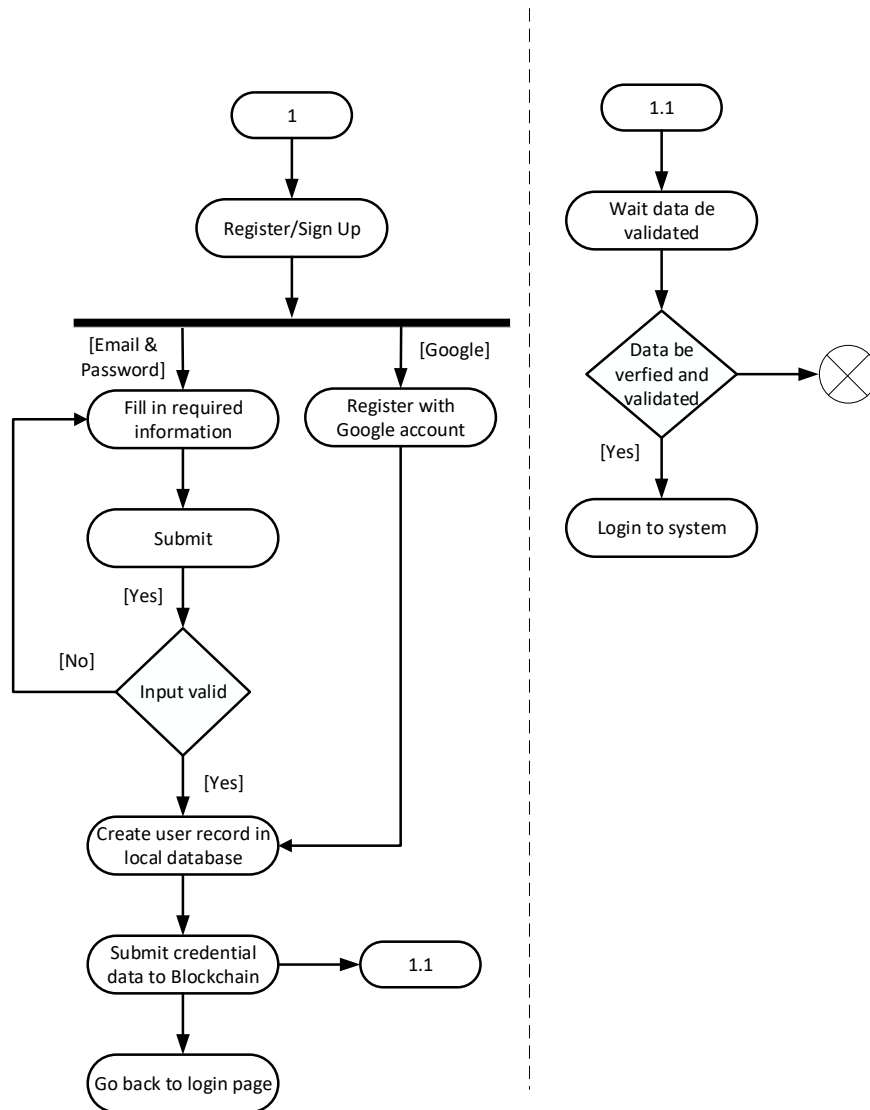


Figure 3.1.3.2: Activity Diagram of Register an Account

The figure above shown that the activity 1 which is register activity for user. User can register an account with 2 methods. One is using email & password and another is using Google email account. If user choose using email & password, user need to fill in the required information such as username, email address, 2 times of same password. After submit, the system will check the input is valid or not especially both password input must be identical. If all input is valid then will proceed to next step while invalid input then system will require user check input and make the input correct. The progress will go to the system create the user record in local database after input is valid then follow that the credential data which is email address and password will submit to blockchain. This is to let the blockchain

Chapter 3

network to validate the data then store the data in a block transaction. Then, the display will be redirected to login page, the activity 1.1 will be running at the same time for waiting the data proof status become validated. The status will start from 'pending' → 'submitted' → 'confirmed', at the end the proof status will be 'valid' or 'failed'. During the transformation process of status, the duration will be around 3 minutes so the user need spend few minutes to wait the blockchain network to validate the submitted data. If the data doesn't be validated then the activity will be ended immediately. Different from the status is not validated or failed, if the data is done to be validated and data status is 'valid' then user able to login to system.

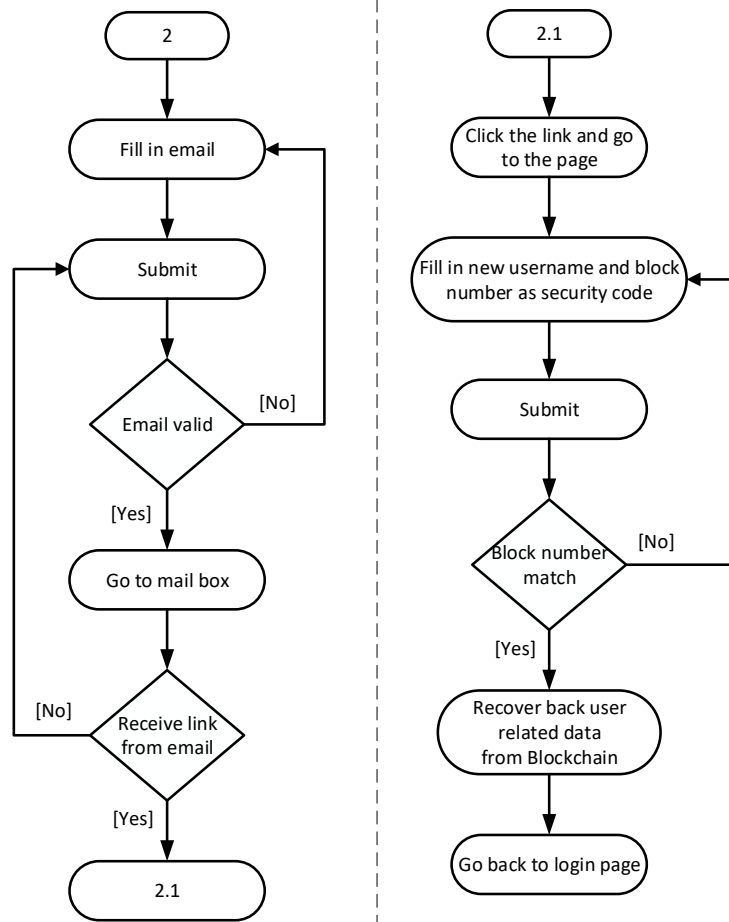


Figure 3.1.3.3: Activity Diagram of Recover Account

For the activity 2, if user currently don't have an account but had an account previously, user wish to recover back the old account at the same time. User can choose the recover account link on login page to redirect to fill in email address on a form then submit to system. The system will email the link with token and user need to check whether received the link or not. If no then back to fill in email address again and submit, if yes then process to activity 2.1. Continue from activity 2.1, user need to click the link then be redirected to fill in new username and block number as security code. After user submit, system will check the 2 times input of block number. If the input is valid then will search the block number from blockchain then retrieve the related data. Next, if system successful retrieved the data from blockchain then will create a list of record that same as previous one to recover back the account data. At last, the activity is done and system will redirect user to login page.

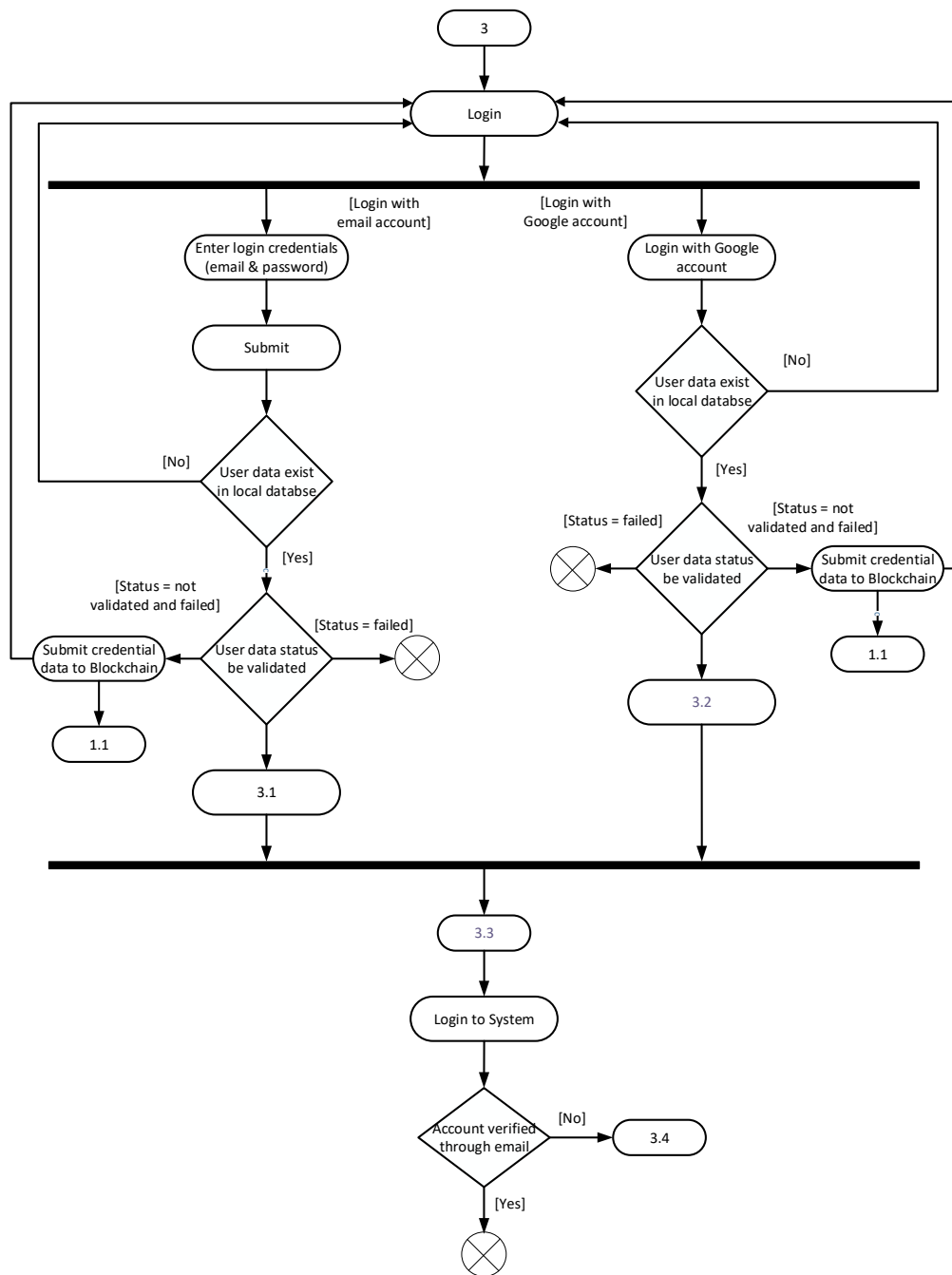


Figure 3.1.3.4: Activity Diagram of Login to System

The figure above presented the activity 3 which is login activity, user have 2 options to login to system which are login with email & password and login with Google account. If user filled in email & password and submit the credentials to system, the system will check the user data exist or not. If the user data not existed in local database then user will be redirected back to login page; if the data is existed then system will check user data status. If the status is not validated yet and not failed then will submit the credential data (email &

Chapter 3

password) to blockchain, the display will be redirected back to login page and the activity 1.1 will executing at the same time. Moreover, if the status is failed then the activity will be ended at all. If user data status is validated then will proceed to activity 3.1.

If login with Google account, the system will check the user data exist or not. If the user data not existed in local database then user will be redirected back to login page; if the data is existed then system will check user data status. If the status is not validated yet and not failed then will submit the Google account ID to blockchain, the display will be redirected back to login page and the activity 1.1 will executing at the same time. Moreover, if the status is failed then the activity will be ended at all. If user data status is validated then will proceed to activity 3.2.

The following activity 3.3 and 3.4 will be explained after activity 3.1 and 3.2.

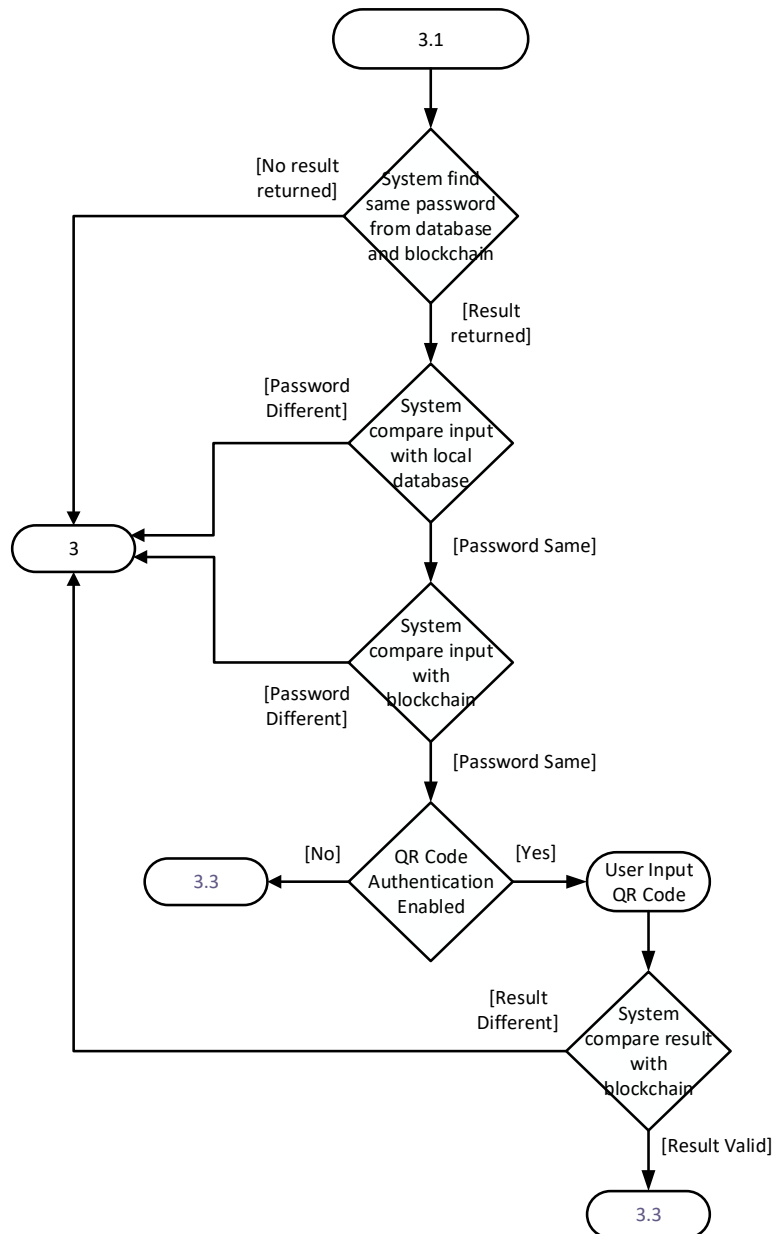


Figure 3.1.3.5: Activity Diagram of System Login Authentication Part using Email & Password

In this activity 3.1, system will find the same password from database and blockchain based on user input previously. If the system didn't get any result then will go back to activity 3 which is start of login; if the system get valid result then will proceed to next step. The next process is system compare password input with database first, if different then will redirect to login page which is activity 3; if the input is still same as result that retrieved from blockchain then the basic authentication is passed and go to activity 3.3. However, if user had enabled the QR Code authentication, then user need to input QR Code and system will

Chapter 3

compare the result which should be a hash proof ID with blockchain proof ID. If result valid then will go to activity 3.3 as well. If user input QR Code result proof ID is different with proof ID from blockchain then the whole authentication considers as fail.

Next, the activity 3.3 is login to system then system will check the account had verified through email account. If yes then nothing else and end the activity flow; if not then will prompt user to activity 3.4.

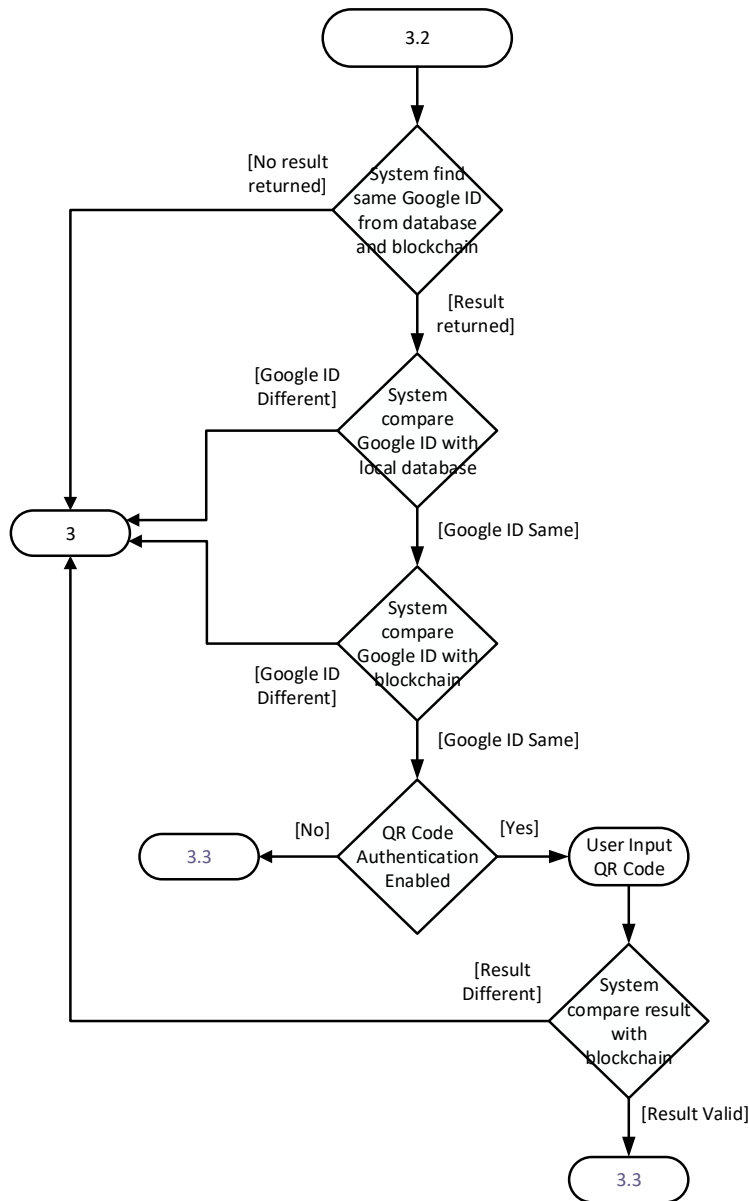


Figure 3.1.3.6: Activity Diagram of System Login Authentication Part using Google Account

In this activity 3.1, system will find the same Google ID from database and blockchain based on user input previously. If the system didn't get any result then will go back to activity 3 which is start of login; if the system get valid result then will proceed to next step. The next process is system compare Google ID with database first, if different then will redirect to login page which is activity 3; if the input is still same as result that retrieved from blockchain then the basic authentication is passed and go to activity 3.3. However, if user had enabled the QR Code authentication, then user need to input QR Code and system will compare the result which should be a hash proof ID with blockchain proof ID. If result valid

Chapter 3

then will go to activity 3.3 as well. If user input QR Code result proof ID is different with proof ID from blockchain then the whole authentication considers as fail.

Next, the activity 3.3 is login to system then system will check the account had verified through email account. If yes then nothing else and end the activity flow; if not then will prompt user to activity 3.4.

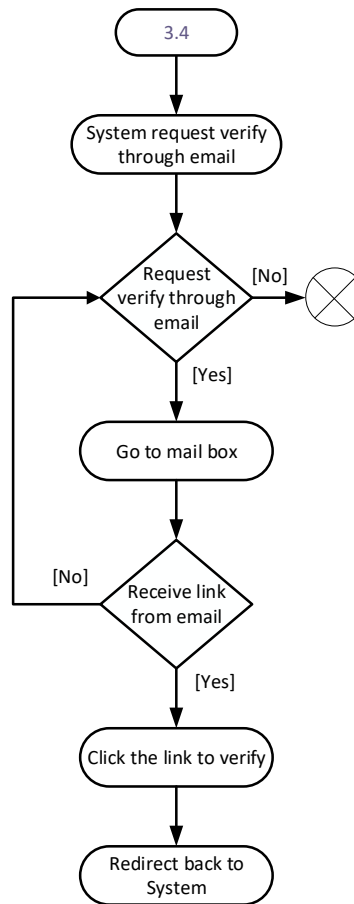


Figure 3.1.3.7: Activity Diagram of System Login Verification through Email

From figure above shown that the activity 3.4 is continue from activity 3.3 which is check account haven't verified through email. In this activity, the system will request user to verify account through email by prompting user to request system send a verification link to user email account. If user clicked, then the system will send the link to mail box and user should receive the link. If user wish to not click to sent mail then the activity flow is ended and the account still on not verified status. Continue from user should receive a link on mail box, if user didn't receive yet, user can request the system to send the link again; if already received then can click the link to verify account then the system will redirect user into the system profile page.

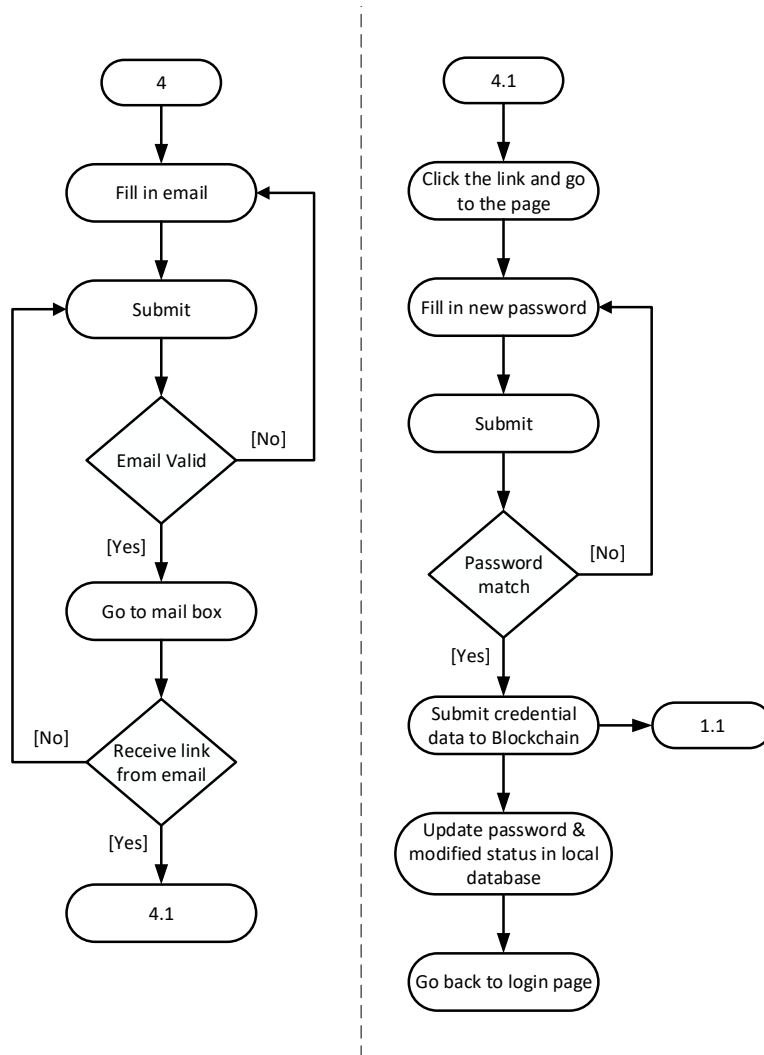


Figure 3.1.3.8: Activity Diagram of Reset Password

For the activity 4, if user already have an account but wish to reset password. User can choose the reset password link on login page to redirect to fill in email address on a form then submit to system. The system will email the link with token and user need to check whether received the link or not. If no then back to fill in email address again and submit, if yes then process to activity 4.1. Continue from activity 4.1, user need to click the link then be redirected to fill in new username and new password. After user submit, system will check the 2 times input of password. If the input is valid and both 2 password input are same then change the old hash password to new hash password. Next, the new password and email will submit to blockchain as credential data. The password and ‘document is modified’ status will be updated in database. In the meanwhile, the activity 1.1 will also be executed. Lastly, the activity is done and system will redirect user back to login page.

Chapter 4

System Design

4.1 System Block Diagram

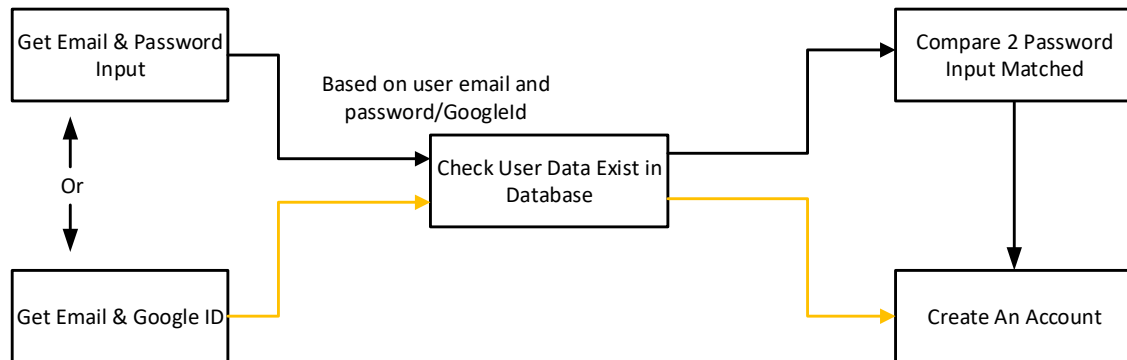


Figure 4.1.1: System Block Diagram for Sign Up

From the above figure 4.1.1 displayed the system block diagram during signup process. At first, system will get user email & password input if user login with email and password or system will get email & Google ID if user login with Google account. Then, user will based on user email and password/Google ID to check the particular data exist in database or not. If not exist in database then system will compare 2 password input matched or not if user choose sign up with email and password, if 2 password input consistent then system will create the user account. If user sign up with Google account after the system checked the user data not exist in database based on email address then straight create an account. If one of the stage fail then whole process will fail then redirect back to login page.

Chapter 4

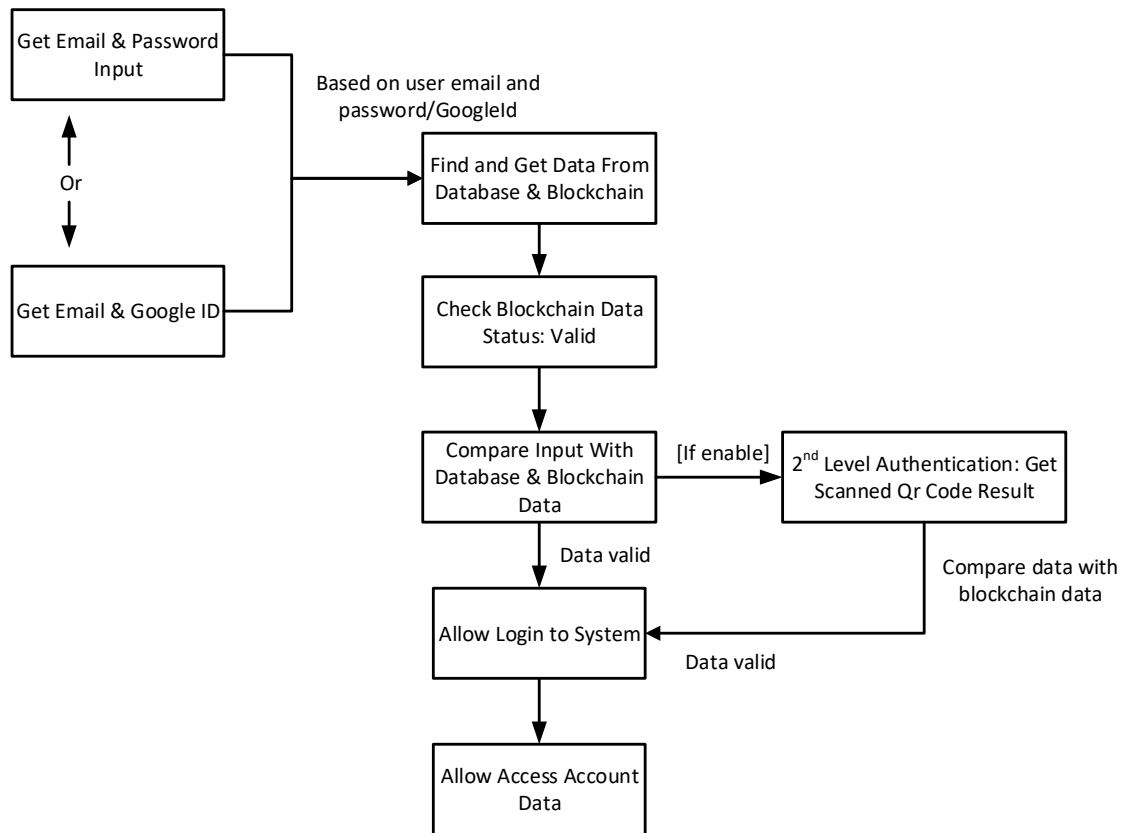


Figure 4.1.2: System Block Diagram for Login

From the figure 4.1.2, the diagram has shown that the system will get user input based on user login method is email & password or Google account ID. After that, system will find and get the specified user data from database and blockchain. Following on this step, system check the blockchain data status is valid or not, if valid then continue next step; else the whole login process will be ended and back to login page. After the status of data is valid then system will compare input with database and blockchain data to make sure 3 side of information is consistent then allow user to proceed to login to system and access account data. In this situation, if user had enabled the 2nd level authentication method which is QR Code then user need to input QR Code and scan the result to compare the result with blockchain data proof ID. If result then the data is valid then will let user login to system and access account data. If one of the stage fail then whole process will fail then redirect back to login page.

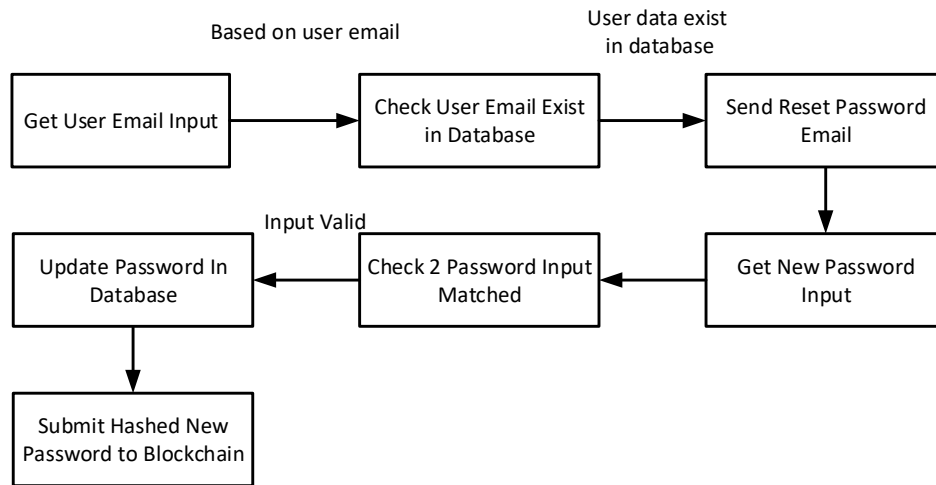


Figure 4.1.3: System Block Diagram for Reset Password

From the figure 4.1.3, system get user email input and check user related data exist in database or not based on email. If yes then send reset password email to user mail box. After user fill in the new password input and submit to system, system will check 2 password input matched or not. If the input is valid then update the password in database and submit the hashed new password to blockchain at the same time. If one of the stage fail then whole process will fail then redirect back to login page.

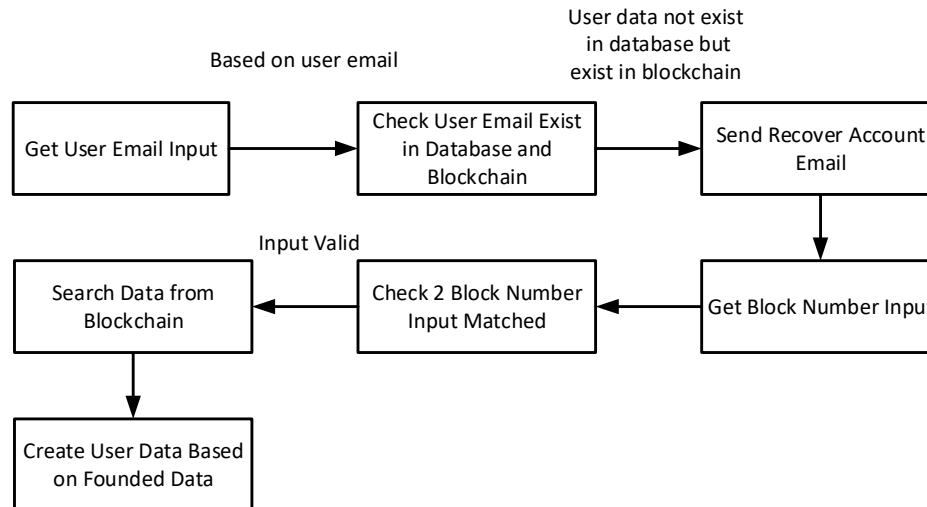


Figure 4.1.4: System Block Diagram for Recover Account

From the figure 4.1.4, system get user email input and check user related data exist in database and blockchain or not based on email. If the user related data not exist in database currently but exist in blockchain then send recover account email to user mail box. After user fill in the block number input as security code and submit to system, system will check 2 block number input matched or not. If the input is valid then search the data on blockchain that based on the block number then create a user account based on founded data. At last, the user old account has been recovered. If one of the stage fail then whole process will fail then redirect back to login page.

4.2 Methodology Model

In this section, the methodology that I used in developing the decentralized authentication system is Rapid application development (RAD). RAD is a methodology that more focus ongoing projects and user feedbacks, and also less to follow strict plan. Chien [19] state that the whole process and the prototyping are be developed rapidly through iterations and get feedback frequently. Nowadays, the RAD is a popular development in business globally. The RAD cycle consists of four phases which are:

1. Define project requirements;
2. Prototype;
3. Rapid construction & feedback gathering;
4. Finalize product / implementation.

Chien [19] also mentioned the main key principle of this methodology process is focus on high iterative design and construction process but reduction in the planning. The prototyping and constructions phase will be repeated until client and users are satisfied the prototype and the system is meet the project requirements. The following figure shown the RAD process concept:

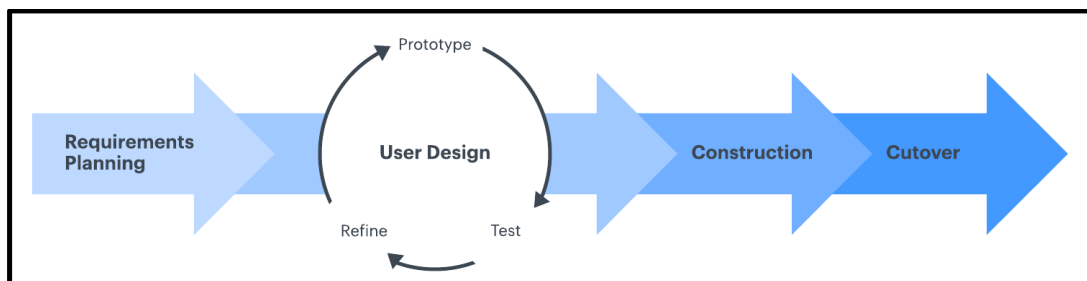


Figure 4.2: RAD overall phases conceptions

The benefits of the RAD is the various types of aspects on the project can be process fast. The RAD programming can reduce the extra risk and costs. Moreover, the feature requirements can be changed at any time this is more efficiency when project involved member is less. Therefore, this methodology is suitable and flexible for me. When I finish to develop a new feature, then I will test out by myself and present to my supervisor which is Ts Dr Gan Ming Lee. If the developed feature achievement is suitable in this project then will proceed to develop next feature, if the result needs some correction and improvement then will refine the part and testing again.

Chapter 5

System Implementation

5.1 Software Setup, Setting and Configuration

In this project, the software that I used to develop the system is Microsoft Visual Studio Code, Docker Desktop, MongoDB Compass, dbKoda and GitHub Desktop. For setup part, the Microsoft Visual Studio Code, MongoDB Compass, dbKoda and GitHub Desktop are installed and setup by common method and steps which is run installation file in .exe extension then follow the instruction to install the software.

While for the Docker Desktop, this software is complex to setup and it is the important part that need to be setup for the ProvenDB blockchain database service. This is used for act as a container hosting the microservice and run the ProvenDB database to provide the database service locally. The required components of Docker Components are Windows Subsystem Linux (WSL), Ubuntu 18.04 LTS. The WSL can be installed through ‘wsl – install’ command in Windows Command Prompt (CMD) or PowerShell. The Ubuntu 18.04 can be installed through Microsoft Store. After install these components, the WSL need to be ensured and can be used by Docker Desktop for first configuration setup.

After Docker Desktop been setup then need to setup the ProvenDB service on docker. I download the Docker-Compose deployment of ProvenDB project file from the official ProvenDB GitHub and the link is: “<https://github.com/SouthbankSoftware/provendb-docker-compose>”. Furthermore, I follow the official tutorial video at YouTube: “<https://www.youtube.com/watch?v=7yGjJ9Bfb44>” for ProvenDB docker compose setup. After the service has been setup in Docker Desktop, the ProvenDB service will automatically start when running the Docker Desktop as below figure 5.1.1.1 shown.

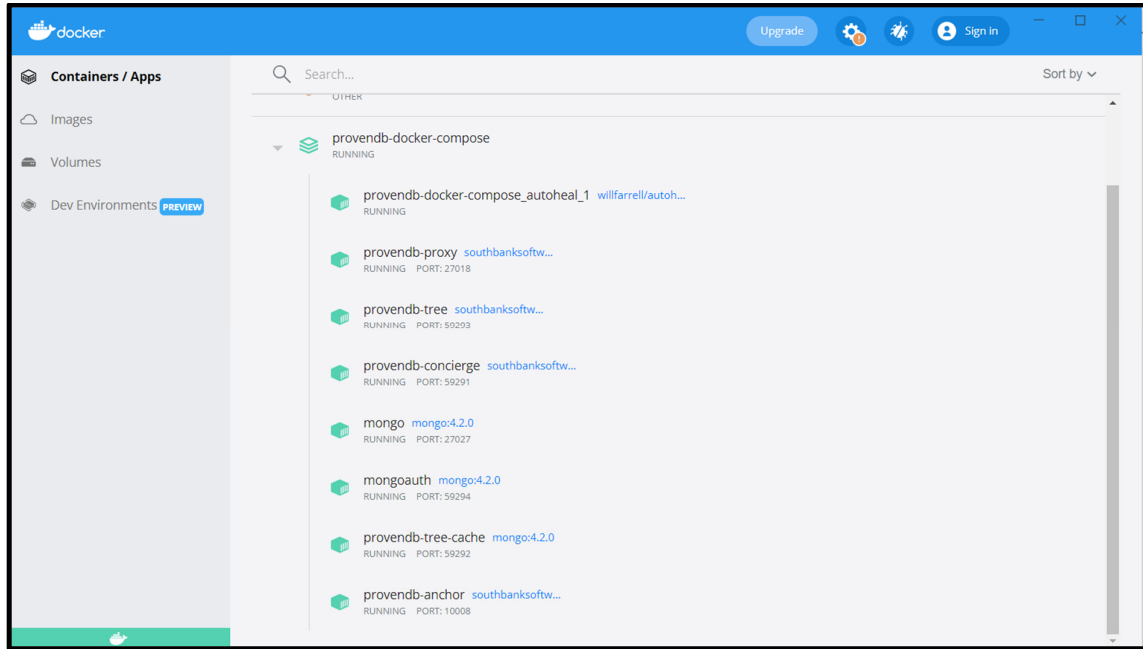


Figure 5.1.1.1: ProvenDB service running on Docker Desktop

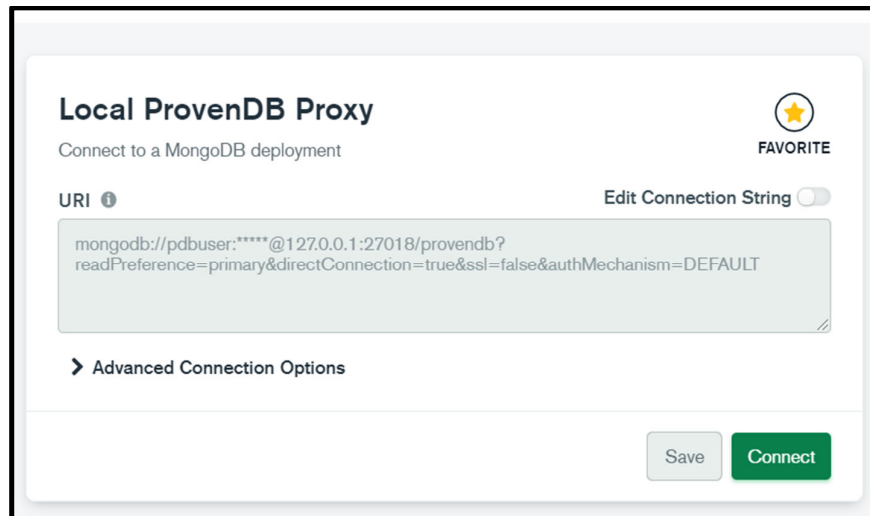


Figure 5.1.1.2: Connection link on MongoDB compass

At last, I can connect to database from this proposed system by using ‘mongodb://pdbuser:click123@127.0.0.1:27018/provendb?’ as above figure 5.1.1.2 displayed. With the same connection link, I can use it in MongoDB compass as below figure 5.1.1.3 demonstrated and Command Prompt as below figure 5.1.1.4 shown to connect, view, analyze, query the data in Graphical User Interface (GUI) method.

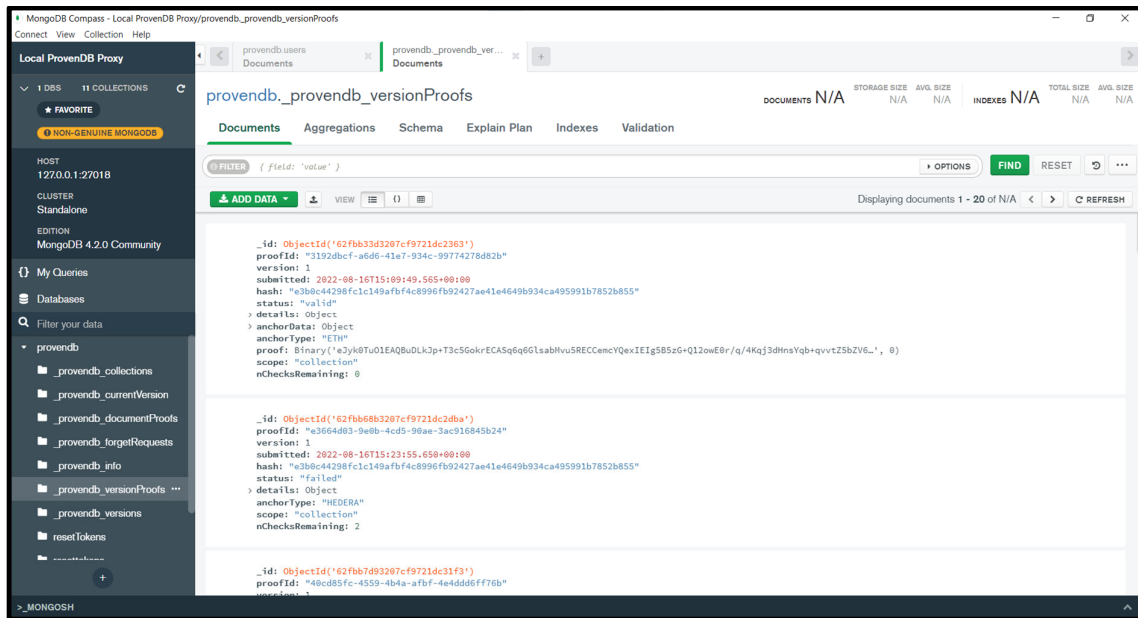


Figure 5.1.1.3: Connected to ProvenDB local service on MongoDB compass

```
C:\Users\chunm>mongo mongodb://pdbuser:click123@127.0.0.1:27018/provenadb?readPreference=primary&
directConnection=true&ssl=false&authMechanism=DEFAULT
MongoDB shell version v5.0.6
connecting to: mongodb://127.0.0.1:27018/provenadb?compressors=disabled&gssapiServiceName=mongodb
&readPreference=primary
Implicit session: session { "id" : UUID("d23da1ef-466e-4e04-bbc3-11b1ef9dce36") }
MongoDB server version: 4.2.0
WARNING: shell and server versions do not match
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will
be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
Warning: Non-Genuine MongoDB Detected

This server or service appears to be an emulation of MongoDB rather than an official MongoDB pro
duct.

Some documented MongoDB features may work differently, be entirely missing or incomplete, or hav
e unexpected performance characteristics.

To learn more please visit: https://docthub.mongodb.org/core/non-genuine-mongodb-server-warning.
> show dbs
provenadb 0.001GB
> show collections
  _provenadb_collections
  _provenadb_currentVersion
  _provenadb_documentProofs
  _provenadb_forgetRequests
  _provenadb_info
  _provenadb_versionProofs
  _provenadb_versions
  resetTokens
  resettokens
  user
  users
>
```

Figure 5.1.1.4: Connected to ProvenDB local service on Command Prompt

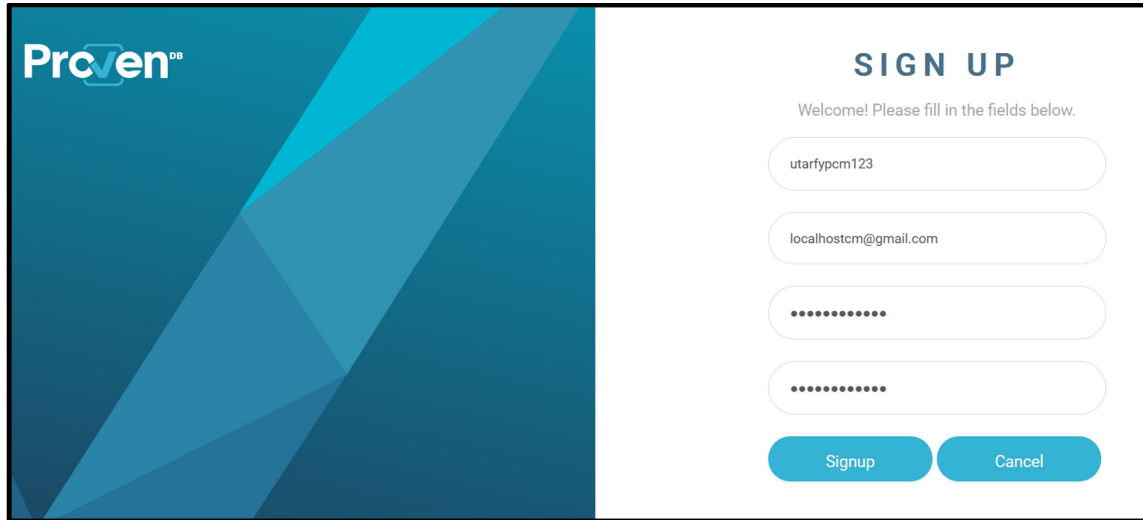


Figure 5.1.1.5: Create ProvenDB cloud service account

In order to use ProvenDB service, the process will start from create ProvenDB cloud service account as figure 5.1.1.5 shown then login to the account as figure 5.1.1.6 displayed.

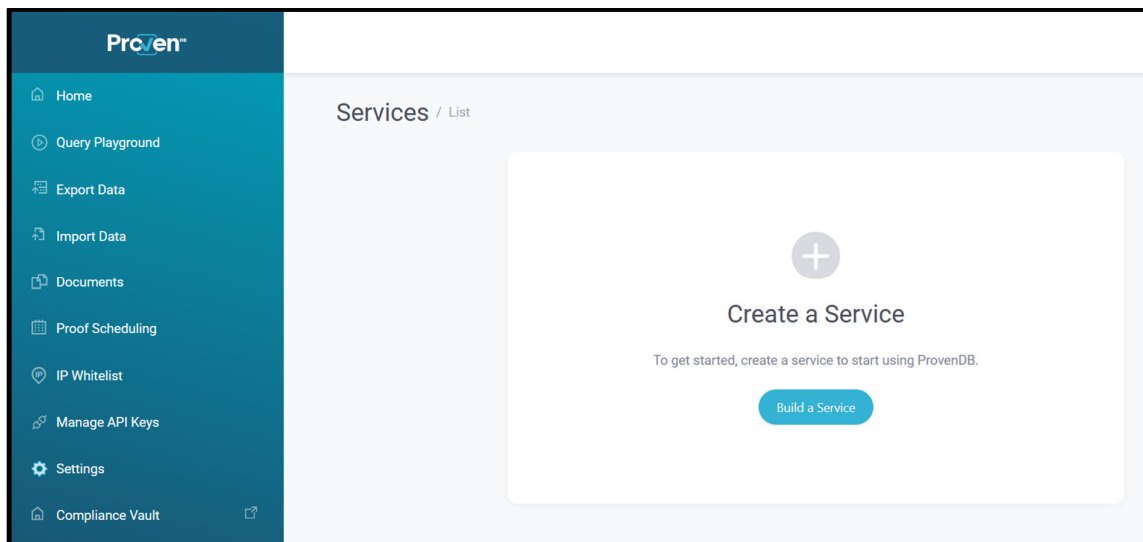


Figure 5.1.1.6: Create ProvenDB cloud service

The screenshot shows the 'New Service' page with the 'Create Service' step active. The 'Service Name' field contains the text 'utarfypcm123'. A note below the field states: 'Please note: once your cluster is created, you wont be able to change its name.' To the right, there is a 'Help' button and the text 'Need help creating a service?'. At the bottom of the form, there is a blue 'Continue' button. Below the main form, there are two steps listed: '2 Authentication' and '3 Plan Selection', both with right-pointing arrows.

Figure 5.1.1.7: Filled in a unique service name

To create a new service, user need to filled in a service name that is unique globally as figure 5.1.1.7 shown. Then, filled in username and password as figure 5.1.1.8 demonstrated for future connect to database with the link that include username and password.

The screenshot shows the 'New Service' page with the 'Authentication' step active. The 'Username' field is filled with 'utarfypcm123'. Below it, a note says: 'Enter the username for logging in to the ProvenDB server.' There are two password fields, both masked with dots. The first is labeled '* Password:' and the second is labeled '* Confirm Password:'. A blue 'Continue' button is located to the right of the password fields. At the bottom, a note says: 'Please keep a safe copy of this password.'

Figure 5.1.1.8: Filled in username and 2 password input

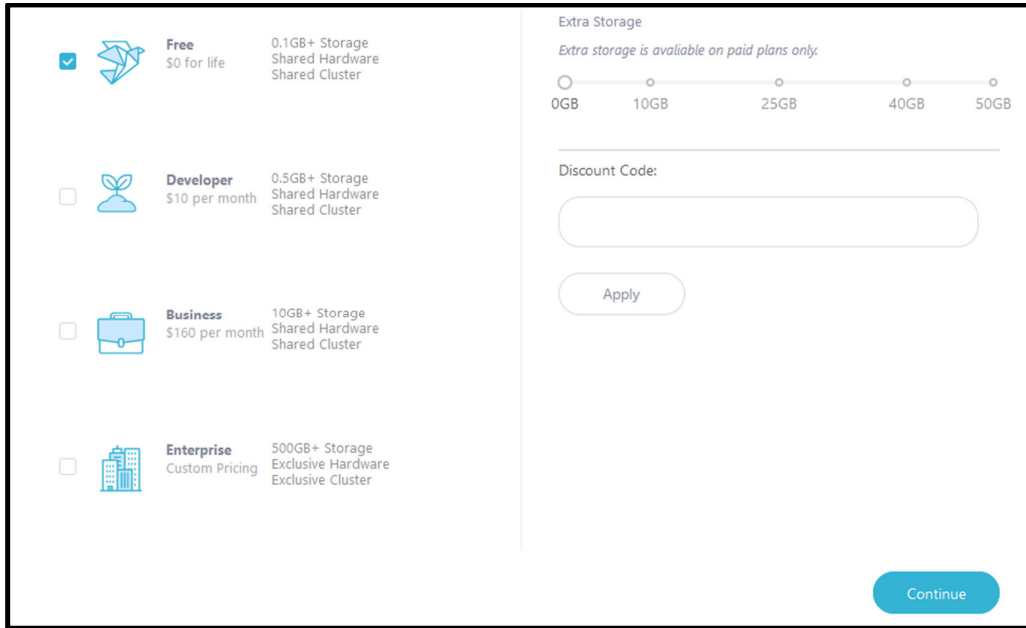


Figure 5.1.1.9: Choose free plan

Last, choose a plan for the service then click ‘Continue’ to create the service as figure 5.1.1.9 shown. In this proposed project, I had chosen free plan that only provide 100MB database capacity which is sufficient to apply in this proposed authentication system. After that, the service has been created as below figure 5.1.1.9 presented. Same as previous mentioned, user can use the URI to connect to the database in MongoDB compass, dbKoda and Command Prompt

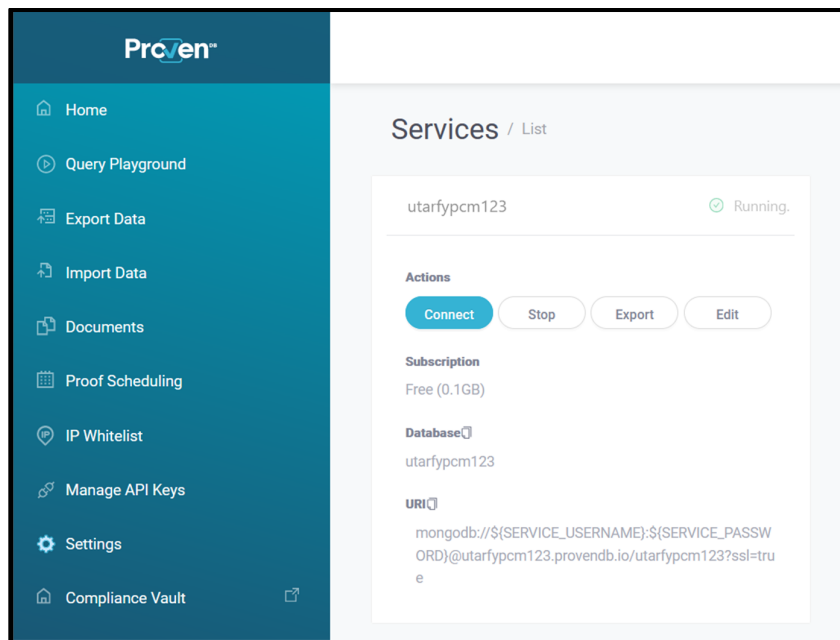
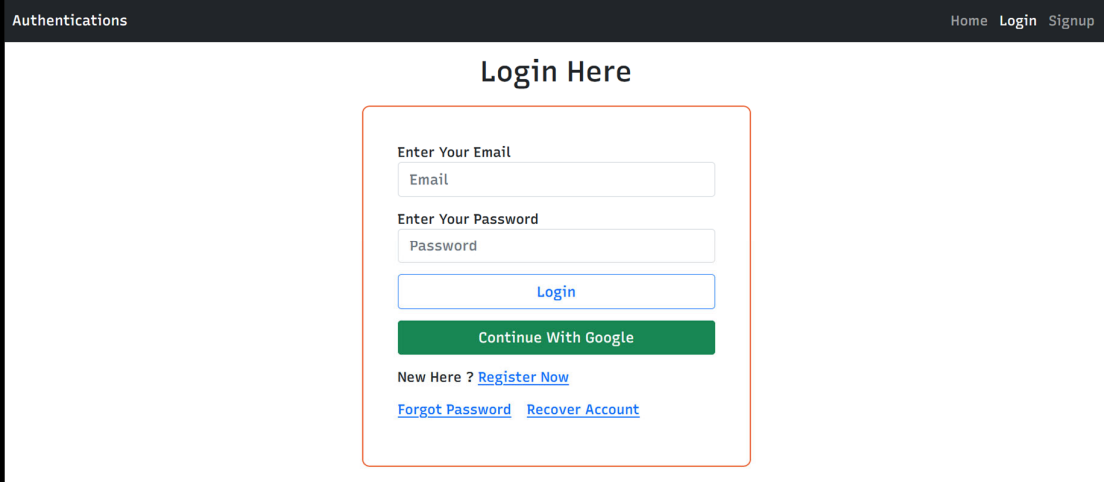


Figure 5.1.1.10: Service be created

5.2 System Operation (with Screenshot)

5.2.1 Start-up Page

In this section, there will be some screenshots of the proposed system prototype as completed work. At first, user will be leaded to the login page as figure 5.2.1.1 below shown when user first contact with the authentication system:



The screenshot shows a web page titled "Authentications" in the top left corner and "Home Login Signup" in the top right corner. The main heading is "Login Here". Below the heading is a form with the following elements: a label "Enter Your Email" above an input field with the placeholder "Email"; a label "Enter Your Password" above an input field with the placeholder "Password"; a blue "Login" button; a green "Continue With Google" button; and three links: "New Here ? Register Now", "Forgot Password", and "Recover Account".

Figure 5.2.1.1: Start-up/Login page

5.2.2 Register an account / Signup

If user doesn't have an account then need to go to the register page that shown as figure 5.2.2.1 below in order to register an account first. At this page, user have 2 options which are filled up require information in register form and click 'Continue with Google' using Google account to register an account. For the first register method, the register form is demonstrated as figure below. User need to filled up the required information in the register form then can submit the form to register an account. If the password is not match with each other in both fields, the system will prompt user to check and re-type again.

5.2.2.1 Register with Email & Password

Authentications Home Login Signup

Register Yourself

Enter Your Email
chun.ming107@1utar.my

Choose A Username
Chun Ming

Choose A Password

Confirm Your Password

Register

Continue With Google

Already Registered ? [LOGIN Here](#)

Figure 5.2.2.1: Register/Signup with Email & Password

Authentications Home Login Signup

Login Here

Message: Please wait for a while to submit data to Blockchain. X

Enter Your Email
Email

Enter Your Password
Password

Login

Continue With Google

New Here ? [Register Now](#)

[Forgot Password](#) [Recover Account](#)

Figure 5.2.2.2: Login page

From the above figure 5.2.2.2 shown that after user registered an account. The user related credential data such as email address and password will be submitted to blockchain. This is to let blockchain network validate the submitted data then store in the blockchain. User need to wait for approximately 2 to 3 minutes to let blockchain verify the data and the particular transaction process. The status of data will start from 'pending' to 'submitted' to 'confirmed' to 'valid' / 'failed'. The process from 'pending' to 'confirmed' is fast, most of

the time is from 'confirmed' to 'valid' / 'failed'. The figures below are the screenshots of the process from 'pending' to 'confirmed' to 'valid' in database and blockchain.

```

_id: ObjectId('6314d010fed5183c2c6b63bd')
isVerified: false
DocIsModified: false
qrcodeAuth: false
username: "Chun Ming"
email: "chun.ming107@utar.my"
password: "$2a$12$0eFQhF8u6WblJMDXH.0DjuD..qMsx/vuroi05rUc6Eth28RIedSW2"
googleId: null
provider: "email"
theStatus: "Pending"
proofId: "9cb0f829-06da-448b-9ea3-26f32fdd5e87"
submittedTime: 2022-09-04T16:19:34.962+00:00
theVersion: 349
blockNum: null
__v: 0

```

Figure 5.2.2.3: User data status 'pending' in database

```

_id: ObjectId('6314d0161c07f7b2251db727')
proofId: "9cb0f829-06da-448b-9ea3-26f32fdd5e87"
version: 349
submitted: 2022-09-04T16:19:34.962+00:00
hash: "cbc80ed0da20e503423ce9adcea819307c4251d903398c1b1b4aaa1b9ca00a87"
status: "pending"
details: Object
  protocol: Object
  collections: Array
    filter: "{ \"email\" : \"chun.ming107@utar.my\", \"password\" : \"$2a$12$0eFQhF8u6Wbl...\"
    batchId: "8iKvKtA8bmvK840mNyNv2"
    format: "CHP_PATH"
  anchorType: "ETH"
  scope: "collection"
  nChecksRemaining: 3

```

Figure 5.2.2.4: User data status 'pending' in blockchain

```

_id: ObjectId('6314d010fed5183c2c6b63bd')
isVerified: false
DocIsModified: false
qrcodeAuth: false
username: "Chun Ming"
email: "chun.ming107@utar.my"
password: "$2a$12$0eFQhF8u6WblJMDXH.0DjuD..qMsx/vuroi05rUc6Eth28RIedSW2"
googleId: null
provider: "email"
theStatus: "Confirmed"
proofId: "9cb0f829-06da-448b-9ea3-26f32fdd5e87"
submittedTime: 2022-09-04T16:19:34.962+00:00
theVersion: 349
blockNum: 11322603
__v: 0

```

Figure 5.2.2.5: User data status 'confirmed' in database

```

_id: ObjectId('6314d0161c07f7b2251db727')
proofId: "9cb0f829-06da-448b-9ea3-26f32fdd5e87"
version: 349
submitted: 2022-09-04T16:19:34.962+00:00
hash: "cbc80ed0da20e503423ce9adcea819307c4251d903398c1b1b4aaa1b9ca00a87"
status: "confirmed"
details: Object
  > protocol: Object
  > collections: Array
    filter: "{ \"email\" : \"chun.ming107@lutar.my\", \"password\" : \"$2a$12$0eFQhF8u6WbL...\"
    batchId: "8iKvKtA8bmvK840mNyNv2"
    format: "CHP_PATH"
  > anchorData: Object
  anchorType: "ETH"
proof: Binary('eJyKkb201DAQgF+GNpvxXxJvtRKvQEUTzYzHF0tLHCW+gyuBhnaf4RZxIEqg5D3yNmgvsB0NtJ/9fSPNfHg8cB6LvCk/h1KmZV/X...', 0)
scope: "collection"
nChecksRemaining: 2

```

Figure 5.2.2.6: User data status 'confirmed' in blockchain

```

_id: ObjectId('6314d010fed5183c2c6b63bd')
isVerified: false
DocIsModified: false
qrCodeAuth: false
username: "Chun Ming"
email: "chun.ming107@lutar.my"
password: "$2a$12$0eFQhF8u6WbLJMDXH.0DjuD..qMsx/vuroi05rUc6Eth28RIedSW2"
googleId: null
provider: "email"
theStatus: "Valid"
proofId: "9cb0f829-06da-448b-9ea3-26f32fdd5e87"
submittedTime: 2022-09-04T16:19:34.962+00:00
theVersion: 349
blockNum: 11322603
__v: 0

```

Figure 5.2.2.7: User data status 'valid' in database

```

_id: ObjectId('6314d0161c07f7b2251db727')
proofId: "9cb0f829-06da-448b-9ea3-26f32fdd5e87"
version: 349
submitted: 2022-09-04T16:19:34.962+00:00
hash: "cbc80ed0da20e503423ce9adcea819307c4251d903398c1b1b4aaa1b9ca00a87"
status: "valid"
details: Object
  > protocol: Object
  > collections: Array
    filter: "{ \"email\" : \"chun.ming107@lutar.my\", \"password\" : \"$2a$12$0eFQhF8u6WbL...\"
    batchId: "8iKvKtA8bmvK840mNyNv2"
    format: "CHP_PATH"
  > anchorData: Object
  anchorType: "ETH"
proof: Binary('eJyKkb201DAQgF+GNpvxXxJvtRKvQEUTzYzHF0tLHCW+gyuBhnaf4RZxIEqg5D3yNmgvsB0NtJ/9fSPNfHg8cB6LvCk/h1KmZV/X...', 0)
scope: "collection"
nChecksRemaining: 0

```

Figure 5.2.2.8: User data status 'valid' in blockchain

5.2.2.2 Register with Google Account

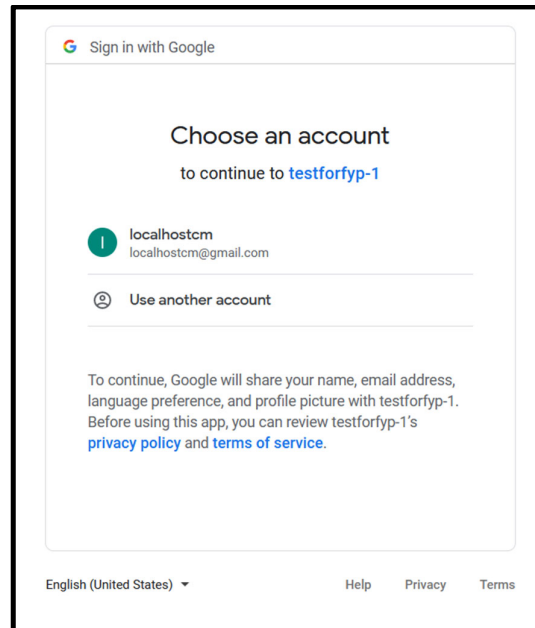


Figure 5.2.2.9: Register with Google Account

From the above figure 5.2.2.9 shown that user can register an account by using own Google account. The user related credential data such as email address and Google ID will be submitted to blockchain. This is to let blockchain network validate the submitted data then store in the blockchain. User need to wait for approximately 2 to 3 minutes to let blockchain verify the data and the particular transaction process. The status of data will start from 'pending' to 'submitted' to 'confirmed' to 'valid' / 'failed'. The process from 'pending' to 'confirmed' is fast, most of the time is from 'confirmed' to 'valid' / 'failed'. The figures below are the screenshots of the process from 'pending' to 'confirmed' to 'valid' in database and blockchain.

```

_id: ObjectId('6314daa0a960b659344ff002')
isVerified: false
DocIsModified: false
qrCodeAuth: false
username: "localhostcm"
email: "localhostcm@gmail.com"
googleId: "106426013287540039625"
password: null
provider: "google"
theStatus: "Pending"
proofId: "18b102d1-fef8-40e4-9001-7d6002ab46c2"
submittedTime: 2022-09-04T17:04:39.404+00:00
theVersion: 355
blockNum: null
__v: 0

```

Figure 5.2.2.10: User data status 'pending' in database

```

_id: ObjectId('6314daa71c07f7b225205928')
proofId: "18b102d1-fef8-40e4-9001-7d6002ab46c2"
version: 355
submitted: 2022-09-04T17:04:39.404+00:00
hash: "bb3184ad14ef81b4dcfa50e77db200587984a7f4390c7b506bf7d3c9e17ea385"
status: "pending"
  details: Object
    protocol: Object
    collections: Array
      filter: "{ \"email\" : \"localhostcm@gmail.com\", \"googleId\" : \"1064260132875400396...\"
      batchId: "2yB20oG-7B7kxAYeFL7w1"
      format: "CHP_PATH"
    anchorType: "ETH"
    scope: "collection"
    nChecksRemaining: 3

```

Figure 5.2.2.11: User data status 'pending' in blockchain

```

_id: ObjectId('6314daa0a960b659344ff002')
isVerified: false
DocIsModified: false
qrCodeAuth: false
username: "localhostcm"
email: "localhostcm@gmail.com"
googleId: "106426013287540039625"
password: null
provider: "google"
theStatus: "Confirmed"
proofId: "18b102d1-fef8-40e4-9001-7d6002ab46c2"
submittedTime: 2022-09-04T17:04:39.404+00:00
theVersion: 355
blockNum: 11322783
__v: 0

```

Figure 5.2.2.12: User data status 'confirmed' in database

```

_id: ObjectId('6314daa71c07f7b225205928')
proofId: "18b102d1-fef8-40e4-9001-7d6002ab46c2"
version: 355
submitted: 2022-09-04T17:04:39.404+00:00
hash: "bb3184ad14ef81b4dcfa50e77db200587984a7f4390c7b506bf7d3c9e17ea385"
status: "confirmed"
  details: Object
    protocol: Object
    collections: Array
      filter: "{ \"email\" : \"localhostcm@gmail.com\", \"googleId\" : \"1064260132875400396...\"
      batchId: "2yB20oG-7B7kxAYeFL7w1"
      format: "CHP_PATH"
    anchorData: Object
    anchorType: "ETH"
    proof: Binary('eJykkT2u1DAQgC9Dm41/Y3ur1bgCFU00Ho9fLIU4SvwebAk0tDnDLMJBLEDJPXIbtBvYjgbaz/6+kWY+XA6Yh0Jvys+uLHHe1/Vr...', 0)
    scope: "collection"
    nChecksRemaining: 2

```

Figure 5.2.2.13: User data status 'confirmed' in blockchain

```
_id: ObjectId('6314daa0a960b659344ff002')
isVerified: false
DocIsModified: false
qrcodeAuth: false
username: "localhostcm"
email: "localhostcm@gmail.com"
googleId: "106426013287540039625"
password: null
provider: "google"
theStatus: "Valid"
proofId: "18b102d1-fef8-40e4-9001-7d6002ab46c2"
submittedTime: 2022-09-04T17:04:39.404+00:00
theVersion: 355
blockNum: 11322783
__v: 0
```

Figure 5.2.2.14: User data status 'valid' in database

```
_id: ObjectId('6314daa71c07f7b225205928')
proofId: "18b102d1-fef8-40e4-9001-7d6002ab46c2"
version: 355
submitted: 2022-09-04T17:04:39.404+00:00
hash: "bb3184ad14ef81b4dcfa50e77db200587984a7f4390c7b506bf7d3c9e17ea385"
status: "valid"
details: Object
  protocol: Object
  collections: Array
    filter: {"email": "localhostcm@gmail.com", "googleId": "1064260132875400396..."}
    batchId: "2yB20oG-7B7kxAYeFL7w1"
    format: "CHP_PATH"
  anchorData: Object
  anchorType: "ETH"
proof: Binary('eJykkT2u1DAQgC9Dm41/Y3urlbgCFU00Ho9fLIU4SwebAk0tDnDlMJBLEDJPXIbtBvYjgbaz/6+kWY+XA6Yh0Jvys+uLHHe1/Vr...', 0)
scope: "collection"
nChecksRemaining: 0
```

Figure 5.2.2.15: User data status 'valid' in blockchain

5.2.3 Login

5.2.3.1 Login Email & Password / Google Account

Login Here

Enter Your Email
chun.ming107@utar.my

Enter Your Password
.....

Login

Continue With Google

New Here ? [Register Now](#)

[Forgot Password](#) [Recover Account](#)

Figure 5.2.3.1: Filled in email address and password in login form

Chapter 5

The figure 5.2.3.1 shown that the user filled in email address and password in login form to login in to the system as below figure 5.2.3.2:

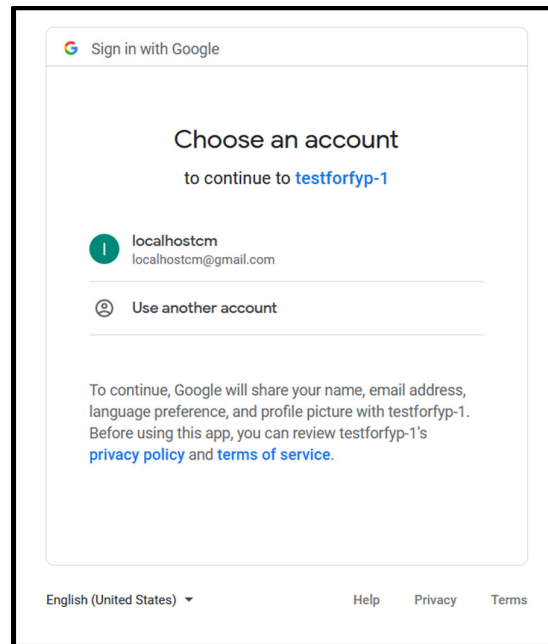


Figure 5.2.3.2: Login with Google Account

From the figure 5.2.3.2 shown that user can login with Google Account that logged in browser. After user choose the Google account then will straight login to system as below figure 5.2.3.3:

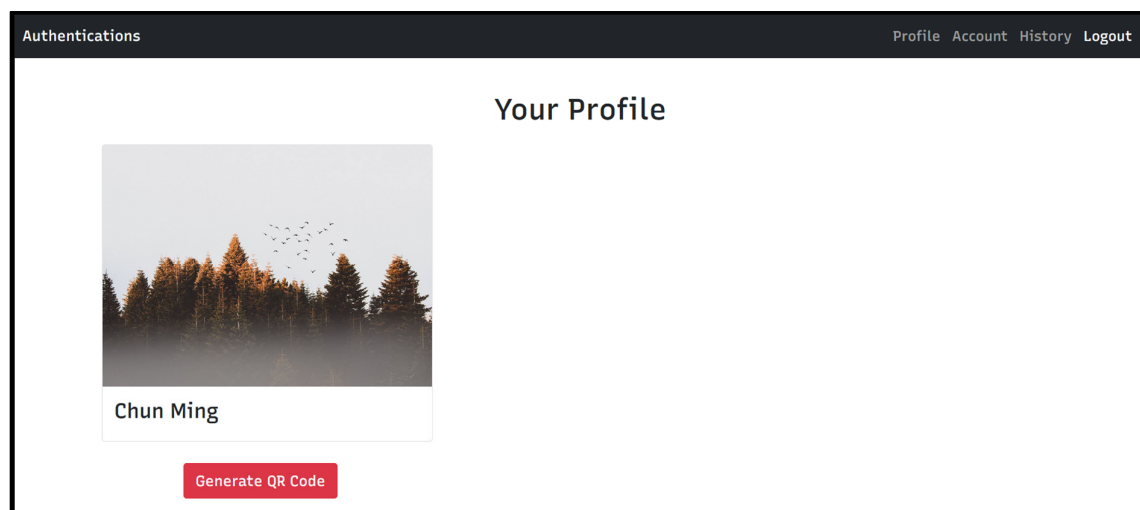


Figure 5.2.3.3: User logged in to system

The screenshot shows a login interface with the following elements:

- Title:** Login Here
- Error Message:** A red box at the top contains the text "Error: The blockchain proof status is: pending" with a close icon (X).
- Form Fields:** Two input fields labeled "Enter Your Email" and "Enter Your Password".
- Buttons:** A blue "Login" button and a green "Continue With Google" button.
- Links:** "New Here ? Register Now", "Forgot Password", and "Recover Account".

Figure 5.2.3.4: User logged in to system during blockchain validating data status: pending

The screenshot shows a login interface with the following elements:

- Title:** Login Here
- Error Message:** A red box at the top contains the text "Error: The blockchain proof status is: confirmed" with a close icon (X).
- Form Fields:** Two input fields labeled "Enter Your Email" and "Enter Your Password".
- Buttons:** A blue "Login" button and a green "Continue With Google" button.
- Links:** "New Here ? Register Now", "Forgot Password", and "Recover Account".

Figure 5.2.3.5: User logged in to system during blockchain validating data status: confirmed

If user try to login during the blockchain validating data then display will show as figure 5.2.3.4 if status is pending or figure 5.2.3.5 if status is confirmed. While if user data already validated the user data status become valid but user enter wrong password, the display will present as figure 5.2.3.6.

The screenshot shows a login form titled "Login Here". At the top, there is a red error message box that says "Error: Password Compare with Local Database Doesn't match !". Below the error message, there are two input fields: "Enter Your Email" with a placeholder "Email" and "Enter Your Password" with a placeholder "Password". There are two buttons: a blue "Login" button and a green "Continue With Google" button. At the bottom, there are three links: "New Here ? Register Now", "Forgot Password", and "Recover Account".

Figure 5.2.3.6: User enter wrong password to login

The screenshot shows a login form titled "Login Here". At the top, there is a red message box that says "Message: User Doesn't Exist !". Below the message box, there are two input fields: "Enter Your Email" with a placeholder "Email" and "Enter Your Password" with a placeholder "Password". There are two buttons: a blue "Login" button and a green "Continue With Google" button. At the bottom, there are three links: "New Here ? Register Now", "Forgot Password", and "Recover Account".

Figure 5.2.3.7: User login with not existed account

The figure 5.2.3.7 demonstrated that user login with not existed or registered account.

5.2.3.2 QR Code as Extra Authentication

The figure 5.2.3.3 displayed that a button 'Generate QR Code' can lead user to a page as shown as below figure 5.2.3.8 to generate QR Code.



Figure 5.2.3.8: QR code page

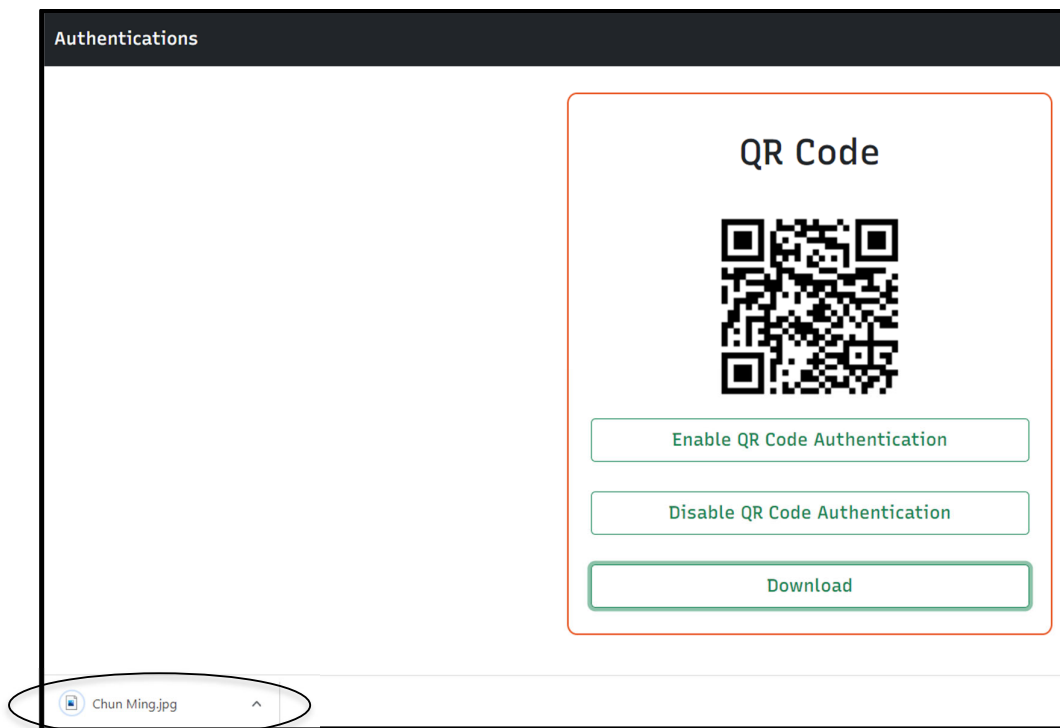


Figure 5.2.3.9: Download QR code

The figure 5.2.3.9 show that the QR Code picture has been downloaded after click the 'Download' button and the file name is based on account username. Then if click the 'Enable QR Code Authentication' then will the display will present as figure 5.2.3.10 the QR Code

Authentication status will be true as figure 5.2.3.11 shown. On the other hand, if click the 'Disable QR Code Authentication' then will the display will present as figure 5.2.3.12 the QR Code Authentication status will be false as figure 5.2.3.13 shown.

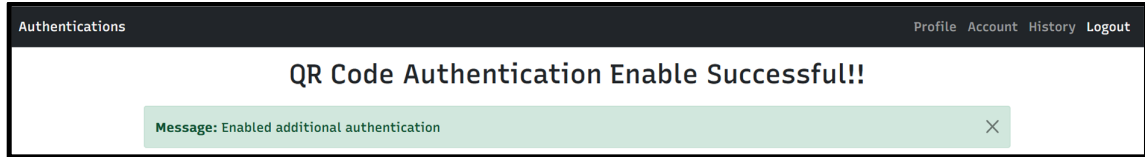


Figure 5.2.3.10: QR code authentication be enabled

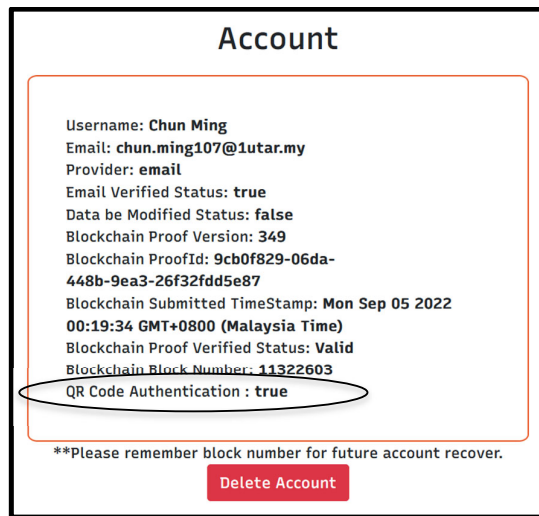


Figure 5.2.3.11: QR code authentication status: true

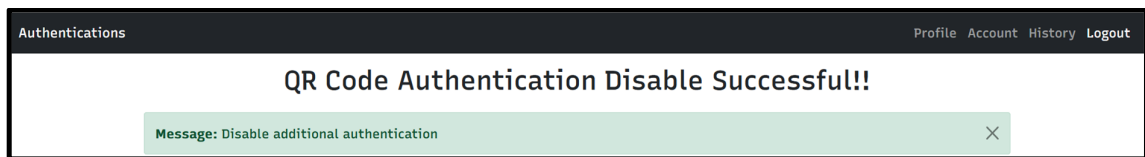


Figure 5.2.3.12: QR code authentication be disabled

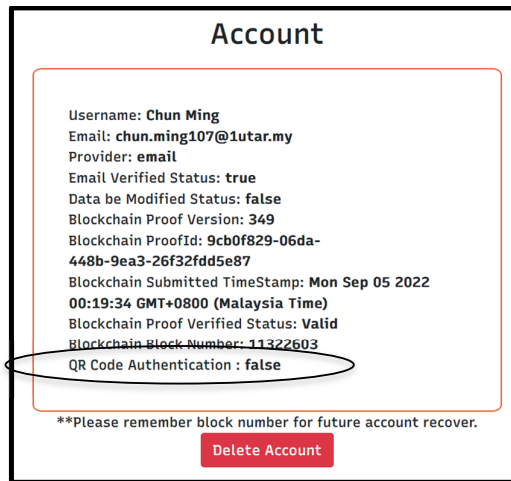


Figure 5.2.3.13: QR code authentication status: false

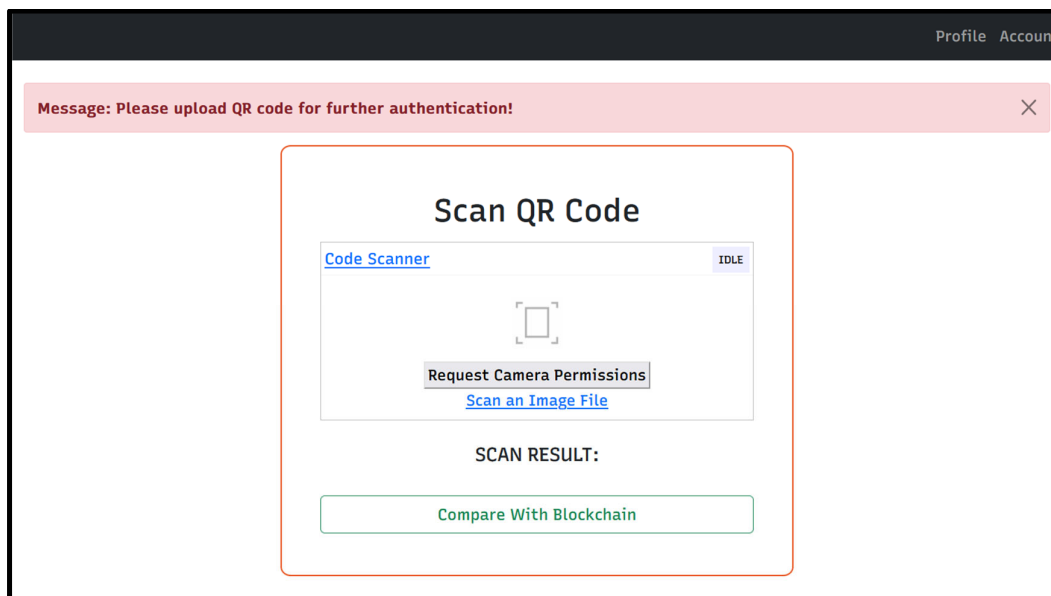


Figure 5.2.3.14: System prompt user upload QR code

If user had enabled the QR code authentication then after login with email address & password or Google account then will redirect user to prompt user upload QR code as figure 5.2.3.14 to let system scan the QR code result.

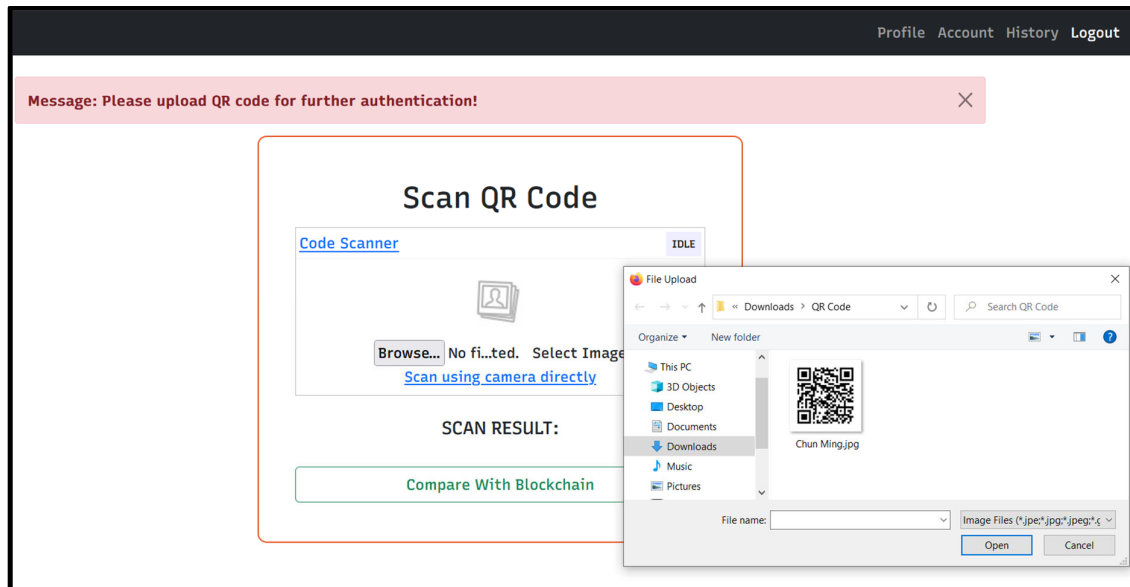


Figure 5.2.3.15: User browse QR code image to upload

User either can choose scan the result by using PC webcam as below figure 5.2.3.17 or upload image file. The figure 5.2.3.15 demonstrated that user upload image file by browsing file explorer folder. Then the following result will be retrieved by system as below figure 5.2.3.16 shown.

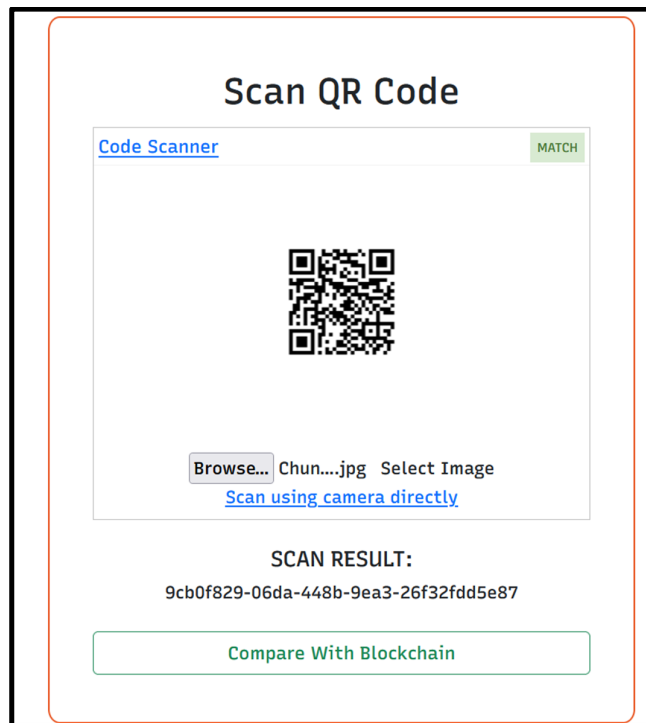


Figure 5.2.3.16: System retrieved QR code result from uploaded image file

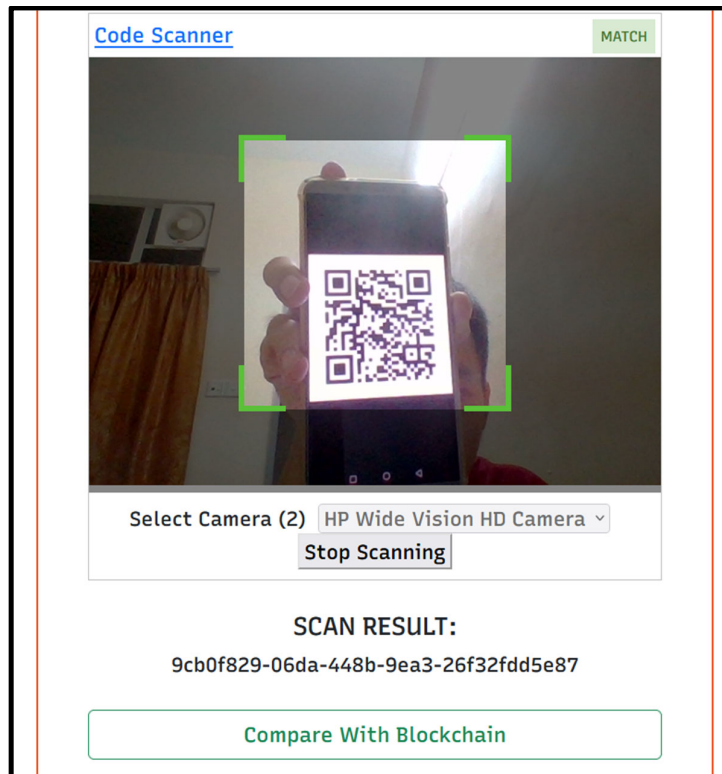


Figure 5.2.3.17: System retrieved QR code result from PC webcam scanned user QR code

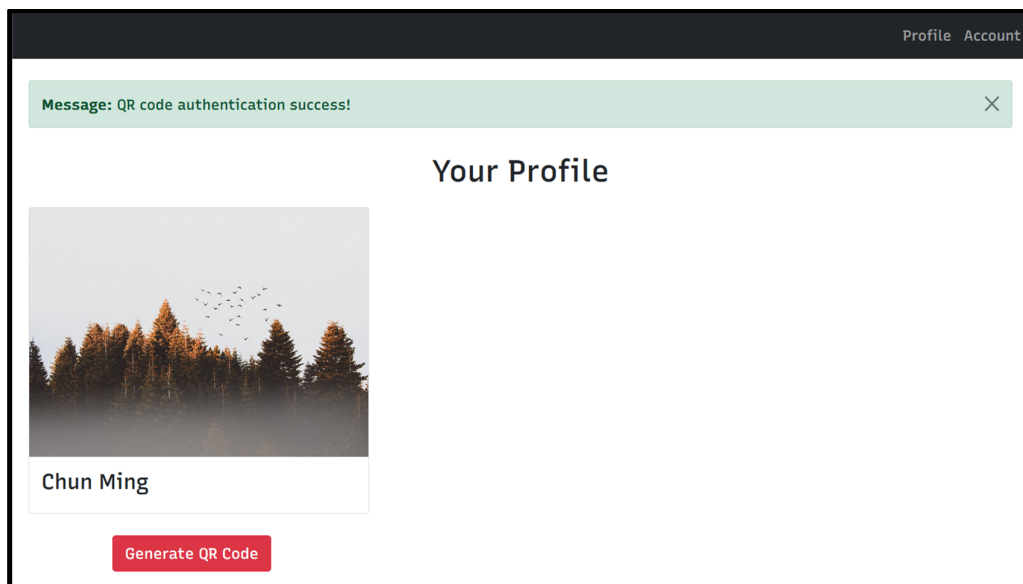


Figure 5.2.3.18: QR Code authentication success and logged in to system

After system get the QR code result and user click 'Compare with Blockchain'. The system will find the related data by using QR code result which is proof ID. If system found

related data then allow user login to system as figure 5.2.3.18 presented; if not then redirect back to login page.

5.2.3.3 Account Verification Through Email

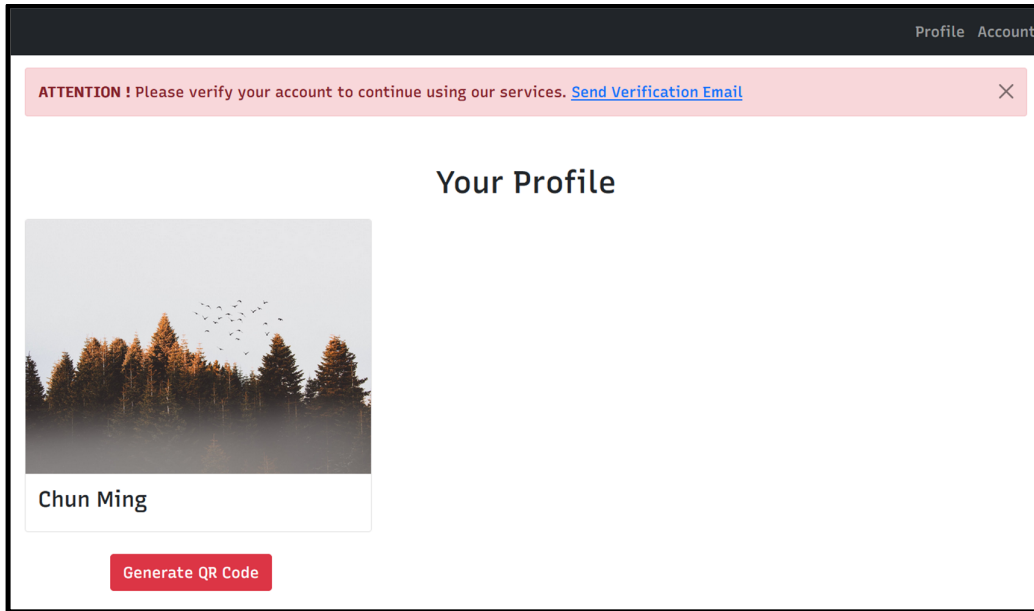


Figure 5.2.3.19: User first time logged in to system

After user first logged into the system then system will prompt user to verified the account through sent verification link to user mail box as figure 5.2.3.19 above. The reason is user account email verified status is 'false' as figure 5.2.3.20 below:

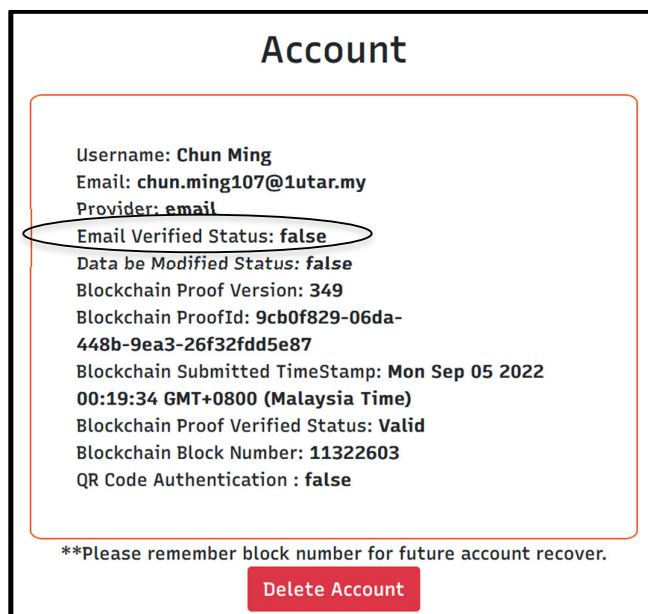


Figure 5.2.3.20: User account information

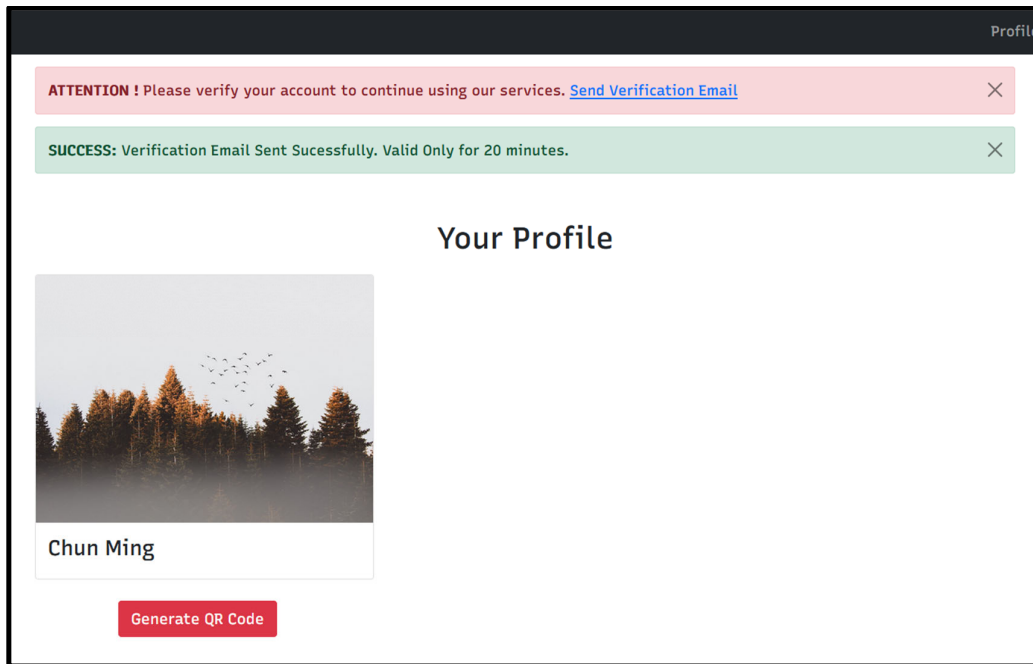


Figure 5.2.3.21: Profile page prompt user verified through email

After user click 'Send Verification Email' as figure 5.2.3.21 then system will send the verification email with link to user mail box that shown as below figure 5.2.3.22.

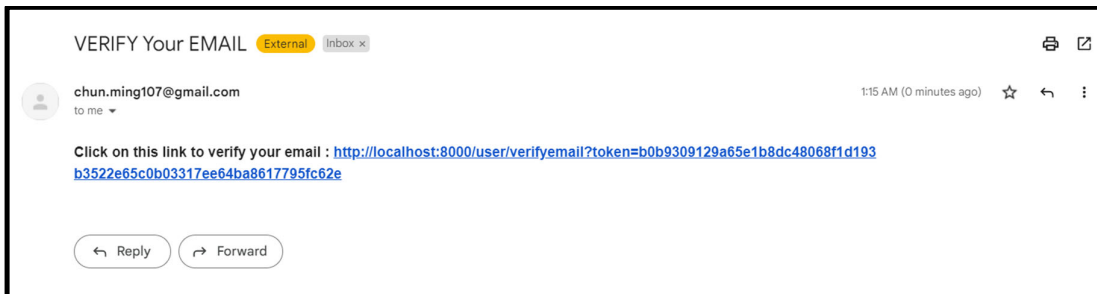


Figure 5.2.3.22: User received verification link on mail box

After user click the link then will go back to profile page and the prompted verification message disappeared as below figure 5.2.3.23 shown. Moreover, the email verified status become true as figure 5.2.3.24 shown.

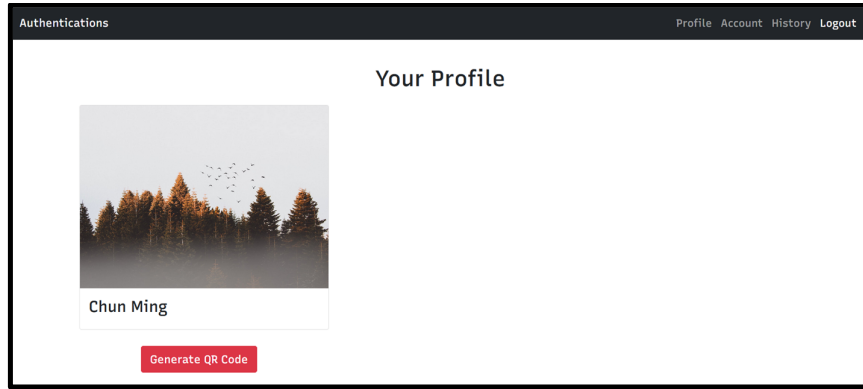


Figure 5.2.3.23: Profile page prompted verification message disappeared

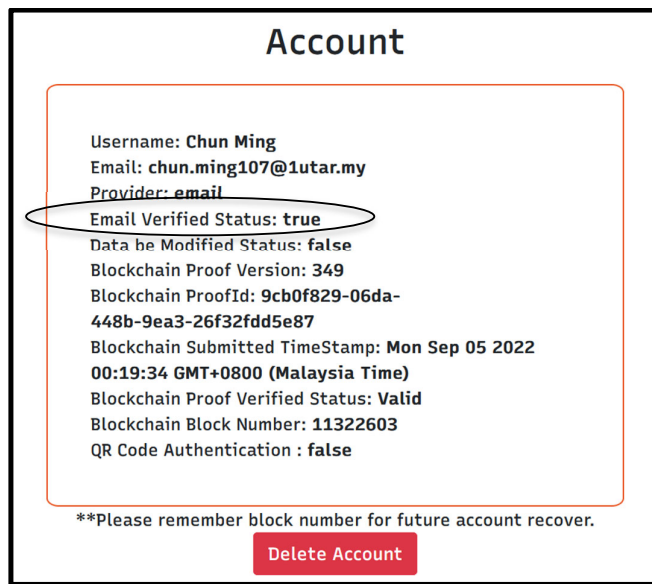


Figure 5.2.3.24: Email verified status become true

5.2.4 Reset Password



Figure 5.2.4.1: User data

From the figure 5.2.4.1 shown that the password is hashed: '\$2a\$12\$2sBUbjseJUMV.fkvFhOyyOM0PO6YJCJwiR5cCYfe0bLbkm2cpT6SpS'. The original password is: 'qwerty123'.

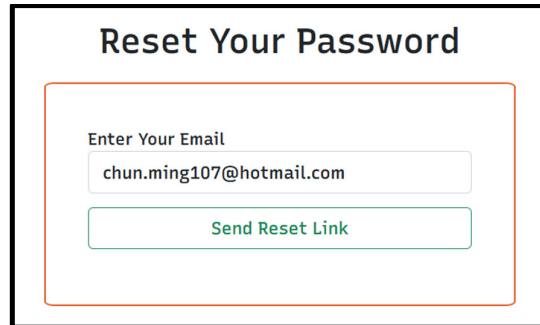


Figure 5.2.4.2: Enter email address to receive reset password link

From figure 5.2.4.2 show that user need to fill in the valid email address that used to registered account before to receive the reset link. Next, the figure 5.2.4.3 shown the email sent. Then, user able to receive the reset link at mail box that presented as figure 5.2.4.4.

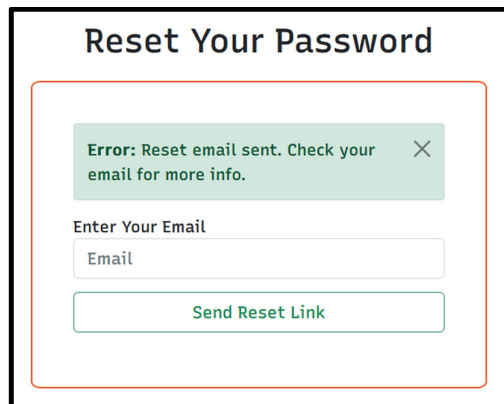
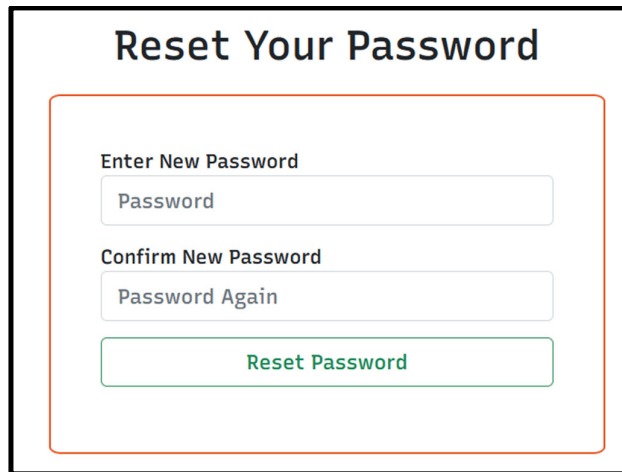


Figure 5.2.4.3: Reset password link sent



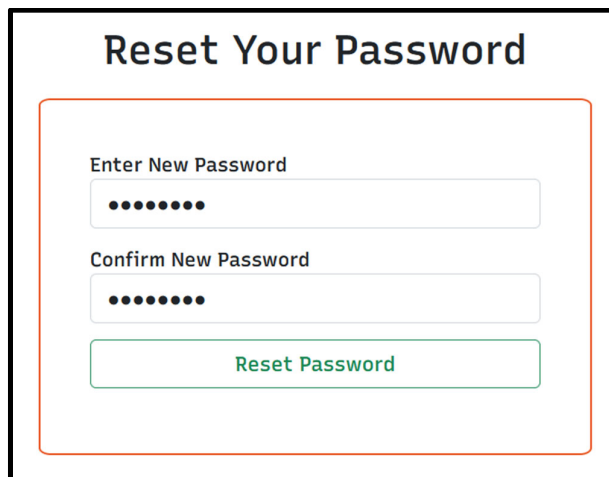
Figure 5.2.4.4: User mail box received reset password link



The screenshot shows a web form titled "Reset Your Password". It contains two input fields: "Enter New Password" with the placeholder text "Password" and "Confirm New Password" with the placeholder text "Password Again". Below the input fields is a green button labeled "Reset Password". The entire form is enclosed in a black border with a red inner border.

Figure 5.2.4.5: Reset password page navigated by email received link

After click the reset link, the display will be navigated to reset password form as figure 5.2.4.5 shown to fill in 2 new passwords that both must be consistent. The figure 5.2.4.6 demonstrated that filled in new password which both are same.



The screenshot shows the same "Reset Your Password" form as in Figure 5.2.4.5, but the input fields are now filled with black dots, indicating that the user has entered a password in both the "Enter New Password" and "Confirm New Password" fields. The "Reset Password" button remains visible below the fields.

Figure 5.2.4.6: Fill in 2 consistent new passwords

Figure 5.2.4.7: Submit email address and hashed new password to blockchain

After user submit the new password from the reset password form, the system will submit hashed new password with email address to blockchain as figure 5.2.4.7 shown. The following step will be same as signup process. User need to wait for approximately 2 to 3 minutes to let blockchain verify the data and the particular transaction process. The status of data will start from 'pending' to 'submitted' to 'confirmed' to 'valid' / 'failed'. The process from 'pending' to 'confirmed' is fast, most of the time is from 'confirmed' to 'valid' / 'failed'. The process from 'pending' to 'confirmed' to 'valid' for data status in database and blockchain will same as figure 5.2.2.3 to figure 5.2.2.8. Then after data be validated as 'valid', the password will be updated as new password. In this case, the new password is '\$2a\$12\$U1xvqeevJnycLB8Jjv/FduhrmWGgjPpA2HYawxPAIxXGTfRqQDeZK' as figure 5.2.4.8 displayed, the original password is 'chunming'.

```

_id: ObjectId('6315fa2fd6e4321540bccfe7')
isVerified: true
DocIsModified: false
qrcodeAuth: false
username: "Chun Ming (Hotmail)"
email: "chun.ming107@hotmail.com"
password: "$2a$12$U1xvqeevJnycLB8Jjv/FduhrmWGgjPpA2HYawxPAIxXGTfRqQDeZK"
googleId: null
provider: "email"
theStatus: "Valid"
proofId: "5fcbb188-300c-4d80-a887-afb84e268ab0"
submittedTime: 2022-09-05T14:06:40.297+00:00
theVersion: 444
blockNum: 11327685
__v: 0

```

Figure 5.2.4.8: Password be updated

Chapter 6

System Evaluation and Discussion

6.1 System Testing and Testing Result

6.1.1 Recover Account

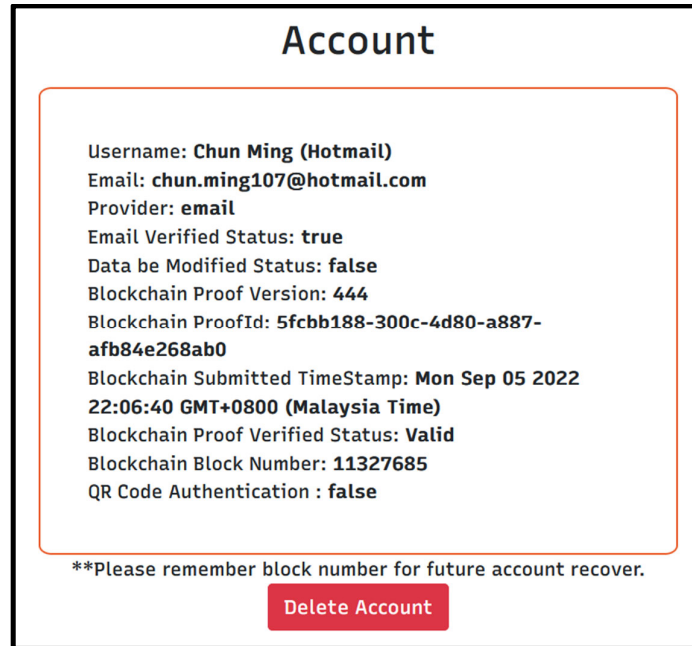


Figure 6.1.1.1: User account data

From the figure 6.1.1.1, user able to delete account by clicking the ‘Delete Account’ button. However, if user want to recover back the account then to memorize the block number as security code.

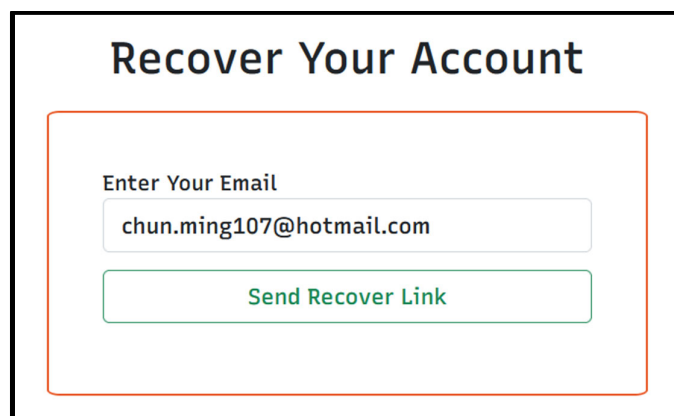


Figure 6.1.1.2: Enter email address to receive recover account link

From the figure 6.1.1.2, user need to fill in the valid email address that used to registered account before to receive the recover link. Next, the figure 6.1.1.3 shown the email sent. Then, user able to receive the recover link at mail box that presented as figure 6.1.1.4.

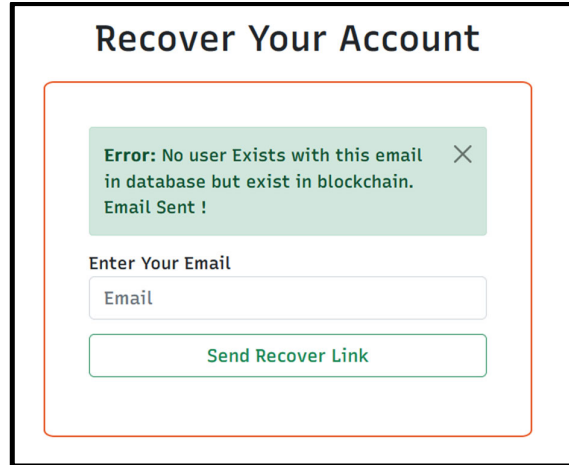


Figure 6.1.1.3: Recover account link sent



Figure 6.1.1.4: User mail box received recover account link

After click the recover account link, the display will be navigated to reset password form as figure 6.1.1.5 shown to fill in 2 new passwords that both must be consistent. The figure 6.1.1.6 demonstrated that filled in new password which both are same.

The screenshot shows a web form titled "Recover Your Account". It contains three input fields: "Enter New Username" with a placeholder "Username", "Enter Block Number" with a placeholder "Block Number", and "Confirm Block Number" with a placeholder "Block Number Again". Below these fields is a green button labeled "Recover Account".

Figure 6.1.1.5: Reset password page navigated by email received link

The screenshot shows the same "Recover Your Account" form. The "Enter New Username" field is now filled with the text "Chun Ming (Hotmail)". The "Enter Block Number" and "Confirm Block Number" fields are filled with seven black dots each, indicating masked input. The "Recover Account" button remains at the bottom.

Figure 6.1.1.6: Fill in 2 consistent new passwords

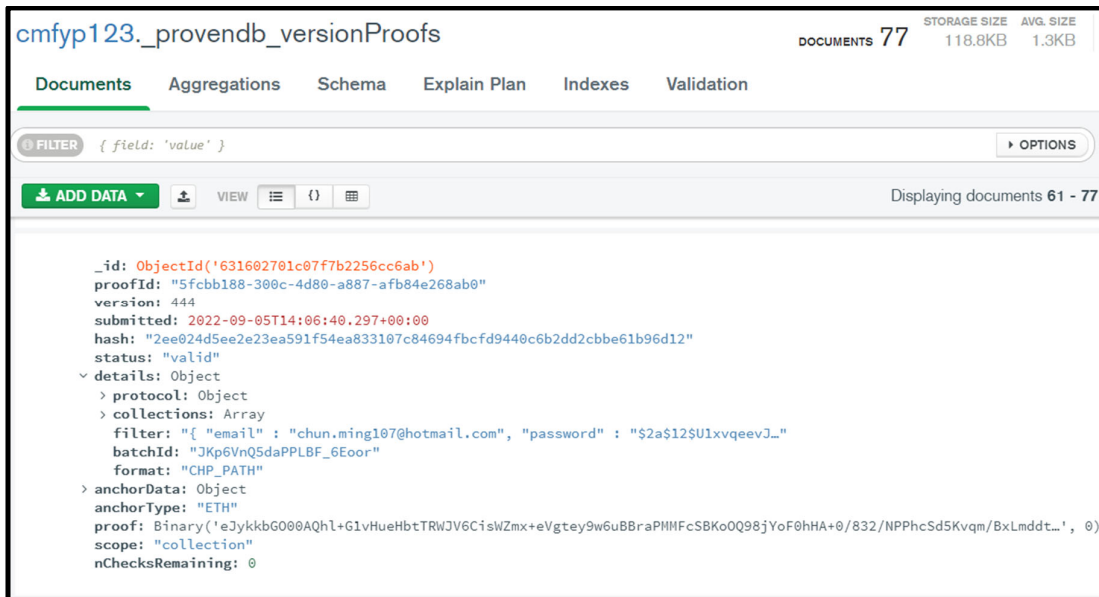


Figure 6.1.1.7: User deleted account related record in blockchain

The following step is system based on the block number to find the specific block as figure 6.1.1.7 then retrieved the related data to create a user record in MongoDB database with the old information. This is because the collection: ‘_provendb_versionProofs’ is storing all the data that submitted to blockchain network. Once the data be validated then the record will be stored in block permanently then link to each other to become blockchain. However, the record only can be read but cannot be updated or deleted so the record is immutable and cannot be modified. After account has been recovered with old data, the display will navigate to below figure 6.1.1.8 shown and user able to login with it.

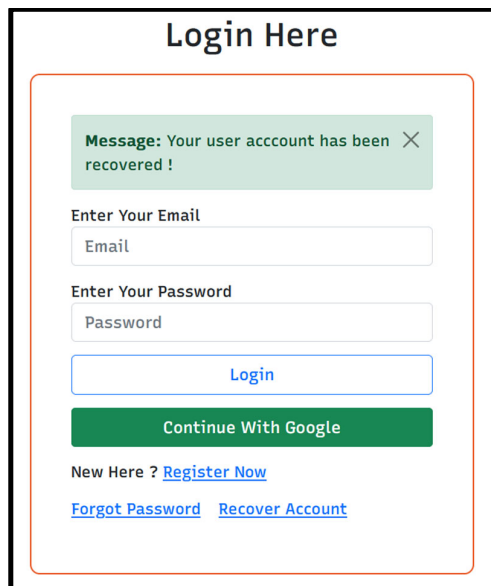


Figure 6.1.1.8: Account recovered and navigated back to login page

6.1.2 ProvenDB Query Operations

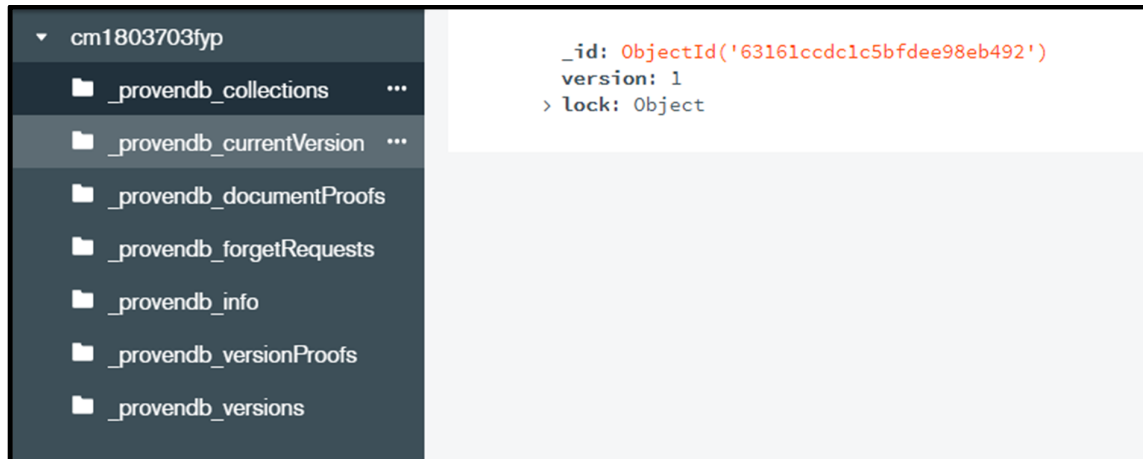


Figure 6.1.2.1: ProvenDB original collections

At this subsection, the figure 6.1.2.1 shown that there few collections in ProvenDB which all collections are built in and related blockchain data when first setup the ProvenDB service.

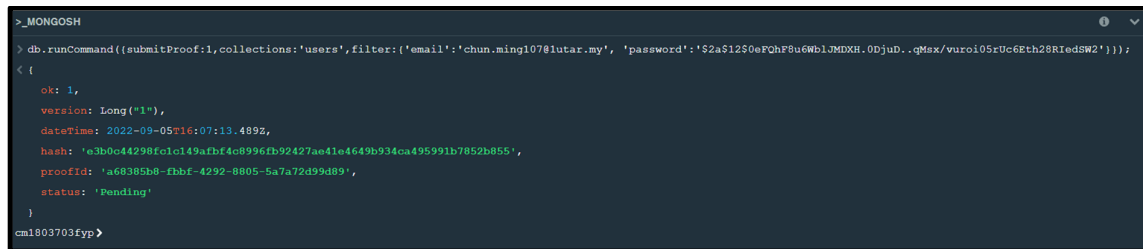


Figure 6.1.2.2: Submit a record to blockchain

In order to prove that the original collections are connected to blockchain and inherit blockchain characteristics such as immutable, security and transparency. From the figure 6.1.2.2 shown, I had submitted a record to blockchain with query:

“db.runCommand({submitProof:1,collections:'users',filter:{'email':'chun.ming107@1utar.my','password':'\$2a\$12\$0eFQhF8u6WbIJMDXH.0DjuD..qMsx/vuroi05rUc6Eth28RIedSW2'}});” . After submit and wait for few minutes, the transaction record has been validated as figure 6.1.2.3 displayed. In this scenario, the next following figures and explanation will describe the operations regarding on the collection record.

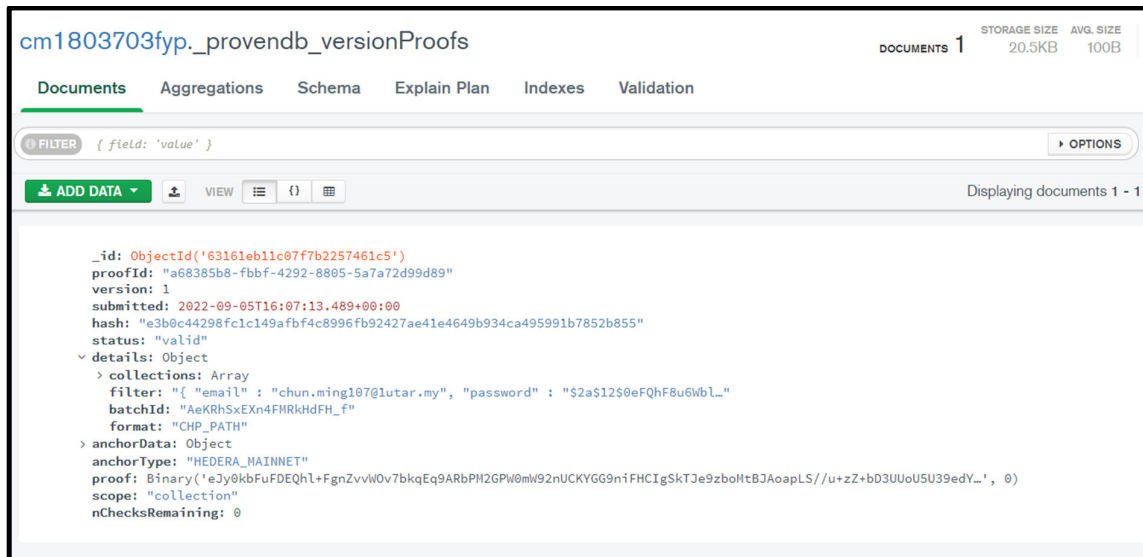


Figure 6.1.2.3: Submitted data be validated in blockchain



Figure 6.1.2.4: Find and retrieve record in _provendb_versionProofs collection

Next, the original built-in collections only can be read but cannot insert data, update and delete data. Therefore, the figure 6.1.2.4 demonstrated that the query: “db.getCollection('_provendb_versionProofs').find({}).pretty();” can find and retrieve record in _provendb_versionProofs collection. The only record proof ID in this case is: 'a68385b8-fbbf-4292-8805-5a7a72d99d89'.

```

> db.getCollection('_provendb_versionProofs').insertOne({proofId: 'chunming1'});
uncaught exception: WriteCommandError({
  "ok" : 0,
  "errmsg" : "Unauthorized operation on restricted collection '_provendb_versionProofs'",
  "code" : 13,
  "codeName" : "Unauthorized"
}) :
WriteCommandError({
  "ok" : 0,
  "errmsg" : "Unauthorized operation on restricted collection '_provendb_versionProofs'",
  "code" : 13,
  "codeName" : "Unauthorized"
})
WriteCommandError@src/mongo/shell/bulk_api.js:421:48
executeBatch@src/mongo/shell/bulk_api.js:936:23
Bulk/this.execute@src/mongo/shell/bulk_api.js:1182:21
DBCollection.prototype.insertOne@src/mongo/shell/crud_api.js:264:9
@(shell):1:1

```

Figure 6.1.2.5: Insert one record in `_provendb_versionProofs` collection

The query: “`db.getCollection('_provendb_versionProofs').insertOne({proofId: 'chunming1'})`” can’t be executing successfully in this collection so there will be error shown in this case because the data only can be create by using ‘submitProof’ command to submit data into the blockchain.

```

> db.getCollection('_provendb_versionProofs').findOneAndUpdate({proofId:
'a68385b8-fbbf-4292-8805-5a7a72d99d89'}, {$set: {status: 'success123'}});
null

```

Figure 6.1.2.6: Find specific record and update it in `_provendb_versionProofs` collection

The query: “`db.getCollection('_provendb_versionProofs').findOneAndUpdate({proofId: 'a68385b8-fbbf-4292-8805-5a7a72d99d89'}, {$set: {status: 'success123'}})`” can’t be executed successfully because blockchain data is immutable and impossible to be tampered. Same as this situation, this collection record cannot be updated even though manual update in MongoDB compass which is a GUI interaction software tool for user.

```

> db.getCollection('_provendb_versionProofs').deleteOne({proofId: 'a68385b8-fbbf-4292-8805-5a7a72d99d89'});
uncaught exception: WriteCommandError({
  "ok" : 0,
  "errmsg" : "You have insufficient privileges",
  "code" : 13,
  "codeName" : "Unauthorized"
}) :
WriteCommandError({
  "ok" : 0,
  "errmsg" : "You have insufficient privileges",
  "code" : 13,
  "codeName" : "Unauthorized"
})
WriteCommandError@src/mongo/shell/bulk_api.js:421:48
executeBatch@src/mongo/shell/bulk_api.js:936:23
Bulk/this.execute@src/mongo/shell/bulk_api.js:1182:21
DBCollection.prototype.deleteOne@src/mongo/shell/crud_api.js:375:17
@(shell):1:1

```

Figure 6.1.2.7: Delete specific record in `_provendb_versionProofs` collection

Same as update, the query:

“`db.getCollection('_provendb_versionProofs').deleteOne({proofId: 'a68385b8-fbbf-4292-8805-5a7a72d99d89'})`,” also failed to run. The reason is same as previous, blockchain data is immutable at all.

```

> db.users.insertOne({proofId: 'chunming1', status: 'success'});
{
  "acknowledged" : true,
  "insertedId" : ObjectId("631629360f71f645990b21d0")
}

```

Figure 6.1.2.8: Insert one record in ‘users’ collection

Different from previous case, this scenario is a normal collection in ProvenDB which is similar like a MongoDB collection. Thus, user not only can read data, insert, update and delete operation can be run successfully. This is to show that the built-in collections are immutable while normal collection has no limit as the built-in collections. First, I run the query: “`db.users.insertOne({proofId: 'chunming1', status: 'success'})`,” to create a simple record.

```

> show collections
_provendb_collections
_provendb_currentVersion
_provendb_documentProofs
_provendb_forgetRequests
_provendb_info
_provendb_versionProofs
_provendb_versions
users
> db.users.find({}).pretty();
{
  "_id" : ObjectId("631629360f71f645990b21d0"),
  "proofId" : "chunming1",
  "status" : "success"
}
>

```

Figure 6.1.2.9: Find and retrieve record in ‘users’ collection

After create the data, the query: “db.users.find({}).pretty();” can be used to retrieve the collection data.

```

> db.users.findOneAndUpdate({proofId: 'chunming1'}, {$set: {status: 'success123'}});
{
  "_id" : ObjectId("631629fa36aae1000697b82b"),
  "proofId" : "chunming1",
  "status" : "success"
}
> db.users.find({}).pretty();
{
  "_id" : ObjectId("631629360f71f645990b21d0"),
  "proofId" : "chunming1",
  "status" : "success123"
}

```

Figure 6.1.2.10: Find specific record and update it in ‘users’ collection

Not only that, the data can updated through query: “db.users.findOneAndUpdate({proofId: 'chunming1'}, {\$set: {status: 'success123'}});”

```

> db.users.deleteOne({proofId: 'chunming1'});
{ "acknowledged" : true, "deletedCount" : 1 }
> db.users.find({}).pretty();
>

```

Figure 6.1.2.11: Delete specific record in ‘users’ collection

Last but not least, the data also can be delete through running the query: “db.users.deleteOne({proofId: 'chunming1'});”.

6.2 Project Challenges

During implementation, the issues that I faced most is the system error that caused by the ProvenDB cloud database services. The reason is when system run the get or submit proof request command to the ProvenDB, the particular service may not respond or fail to execute. Every time this issue occurs, I need to restart ProvenDB services then the service is recovered and my system particular request can be successful execute. In this situation, I assume there is much more request be sent to the ProvenDB, and the service plan currently is free package that I chose. Therefore, the service is stuck and can't handle any request about the blockchain proof anymore. The temporarily solution to fixed this problem is using local ProvenDB database service instead of using cloud service. However, this will bring another problem which is I spend few of time to setup the Docker Desktop which is a container that hosting microservice like ProvenDB service in local environment. In additional, the local ProvenDB service do not support numbers of blockchain network so the blockchain network options has been limited.

The main challenge of this proposed approach is same as the existing proposed solution that applied public blockchain which is performance and waiting time for blockchain network validate data. This is because there are large number of nodes be included in public blockchain network. After participants submit proof, the network nodes need minimum few minutes for the validating proof process and deal with the transactions. Afterward, the submitted information will be added with proof records include timestamp, hash, status, etc. This can show that the performance of public blockchain still is a concern and challenge at present time.

6.3 Objectives Evaluation

The purpose of the thesis is to propose a decentralized authentication system by utilizing blockchain. The list of main objectives is:

- To develop a login database utilizing blockchain
- To ensure available of database as a public ledger
- To provide confidentiality of the login credentials in the database
- To design an authentication system utilizing the blockchain database

First of all, the ProvenDB include MongoDB compatible database service that integrated with blockchain and inherit both traditional database features and blockchain characteristics. The database has equipped with login featured and it is linked with blockchain so the first objective has been achieved.

Second, same concept as previous mentioned which is ProvenDB already comprised blockchain inside so the database has the blockchain characteristics which are immutable, decentralized, transparency, public ledger. Thus, the database can act as a public ledger that thanks to involved blockchain.

Third, the confidentiality of login credentials can be ensured because the user password is hashed in 60 unit that include capital letters, small letters, number and symbol. Even if the password has been accessed or leaked out but the password is hashed so is hard to get to know the original password content. The password also will be submitted to blockchain, so nobody can modify the login credentials and this can also ensure the data integrity.

At last, I had utilized ProvenDB which is a blockchain database to design and develop an authentication system. The ProvenDB had played 2 important roles which are the database and blockchain element in this project and I just need to handle 1 component instead of 2.

6.4 Concluding Remark

In summary, this proposed system is built in website-based and is code with HTML and JavaScript for frontend part. The backend logic and data process routing part are handled by Node.js and Express.js. The most important is the database and blockchain, the ProvenDB is combination of these 2 and able to provide a non-relational database and public blockchain. The non-relational database is MongoDB and is suitable for independent data and able to provide fast performance for request and response the data process. The blockchain is act as an immutable second database to store the credential data to ensure once data is submitted to blockchain and validated, the data can be tempered and will permanently store in a block for future retrieve and refer. The authentication logic is comparing the user input, database data and blockchain record to ensure all sides of information is consistent then allow user login to

Chapter 6

system, if one side of information is not same then there must be something wrong for the user data. Not only that, I had tried out both ProvenDB cloud and local service, the performance and limitation are different. In order to setup ProvenDB local service, I need to setup Docker Desktop as container to handle the microservice locally. After setup finish and the service is running, then I can connect the authentication system to the ProvenDB local service with a MongoDB compatible connection link. Lastly, with the third-party email address involve, this can make sure that the operation like email verify, reset password is only request and handle by account owner.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

To sum up all the things, I had designed an authentication system utilizing the blockchain database. I have been motivated to utilizing the blockchain technology and hash features to apply in the proposed system in order to increase the system security. Instead of using the traditional database that is without protection, this proposed solution is using the ProvenDB service as blockchain database to enhance the security aspect. The reason is ProvenDB have integrate the blockchain technology and features with the document-based database – MongoDB. Not only that, the hash features also be applied when storing the user password, this can make the data transparent to public and also ensure the privacy of user. Therefore, the database can become an eligible public ledger and confidentiality of login credentials is provided in the database at the same time. Moreover, general blockchain is immutable so it does not support the data modification and deletion but the ProvenDB provide version concept able to allow the modification and deletion operation on the blockchain database. Thus, the CRUD operation can be performed in this proposed system like a general database. However, this proposed project still have some problems need to face and solve. For examples: the unstable of ProvenDB cloud services and the performance of public blockchain. If come to cloud service request peak, the data submission to public blockchain as proof will stuck in the middle of the data event process. The solution of this problem is using the local ProvenDB service that hosted in container which is Docker Desktop in this case. Therefore, there will not have the problem of service submit proof to blockchain had stuck. Conversely, the blockchain anchor network has been limit to 2 choices compare to cloud service able to provide at least 4 options or above. Furthermore, not to mentioned that the public blockchain need public network nodes to validate the submitted data. This need much of time for over than half of the blockchain network nodes to finish the information validation process.

7.2 Future Work

In this project, the applied blockchain is public Bitcoin blockchain. Due to public blockchain has the limitation of performance and longer processing time compare to other types of blockchain such as private blockchain, consortium blockchain and hybrid blockchain. The future of work of this project can consider to switch the public blockchain to other types of blockchain. For the case of authentication system, the hybrid blockchain may be most suitable type of blockchain to be chosen to use. In fact, the owner of the hybrid blockchain can control the transparency of data. The reason is hybrid blockchain can control and manage the permission for access and only allow validated nodes participate in the network. This means that the security and transparent can be balanced and achieved, so some of the data can be setting as accessible but the security still can be ensured at the same time. In addition, the hybrid blockchain is flexible to manage it and able to modify the settings based on user requirements. Thus, the hybrid blockchain provide privacy but still can connect with public network. The popular blockchain attack like control 51% of node can be prevented in hybrid blockchain, all the participants must be verified before enrolling themselves in the network.

At present time, the most famous and popular blockchain is Ethereum blockchain compare to Bitcoin blockchain. The Ethereum blockchain has the Proof of Stake (PoS) consensus which is more advanced compare to Bitcoin blockchain Proof of Work (PoW). Moreover, Ethereum blockchain provide programmable smart contract that can manage and integrate with blockchain data contents. Developer can customize and do programming on the Ethereum smart contract by using Solidity programming language which is specified use for this. Furthermore, the smart contract can be run on the Ethereum Virtual Machine (EVM). The EVM is involved in every Ethereum node to convert the smart contract content into bytecode low-level machine language code then execute it. Other than that, the Ganache software testing tool is used to run the Ethereum blockchain application or project in local environment. This allow developer to develop, deploy and test out the application. Ganache provided advance mining controls to manage each block settings and blockchain log outputs to allow developer analyze the request and response output. Therefore, more users and developers choose Ethereum blockchain in their project compare to other blockchain.

REFERENCES

- [1] Cloudflare. “What is a DDoS attack?” Cloudflare.com. <https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/> (accessed March. 26, 2022).
- [2] G. Iredal. “6 Key Blockchain Features You Need to Know Now.” 101Blockchains.com. <https://101blockchains.com/introduction-to-blockchain-features> (accessed March. 26, 2022).
- [3] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” Bitcoin, March. 2009. [Online]. Available: <https://bitcoin.org/bitcoin.pdf>
- [4] M. Chand. “Top 5 Blockchain Programming Languages” c-sharpcorner.com. <https://www.c-sharpcorner.com/article/top-5-best-programming-languages-for-blockchain-development/> (accessed August. 21, 2022).
- [5] G. Wood, “Ethereum: A Secure Decentralised Generalised Transaction Ledger,” Ethereum, 2014. [Online]. Available: <https://gavwood.com/paper.pdf>
- [6] V. Buterin, “A Next Generation Smart Contract & Decentralized Application Platform,” Ethereum, December. 24, 2014. [online]. Available: https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_White_Paper_-_Buterin_2014.pdf
- [7] E. Tarasenko. “The Most Popular Blockchain Programming Languages” merehead.com. <https://merehead.com/blog/the-most-popular-blockchain-programming-languages/> (accessed August. 21, 2022).
- [8] The Linux Foundation, “Hyperledger Overview,” Hyperledger, July. 2018. [Online]. Available: https://www.hyperledger.org/wp-content/uploads/2018/07/Hyperledger-Overview_July-2018-3.pdf
- [9] C. Fromknecht, D. Velicanu and S. Yakoubov, “CertCoin: A NameCoin Based Decentralized Authentication System,” May. 14, 2014. [Online]. Available: <https://courses.csail.mit.edu/6.857/2014/files/19-fromknecht-velicann-yakoubov-certcoin.pdf>
- [10] L. Stockburger, G. Kokosioulis, A. Mukkamala, R. R. Mukkamala and M. Avital., “Blockchain-Enabled Decentralized Identity Management: The Case of Self-Sovereign Identity in Public Transportation,” in ScienceDirect, June, 2021. [Online]. Available: <https://doi.org/10.1016/j.bcr.2021.100014>
- [11] I. T. Javed, F. Alharbi, B. Bellaj, T. Margaria, N. Crespi, and K. N. Qureshi., “Health-ID: A Blockchain-Based Decentralized Identity Management for Remote Healthcare,” in MDPI, June. 10, 2021. [Online]. Available: <https://www.mdpi.com/2227-9032/9/6/712/htm>

References

- [12] C. Lin, D. He, N. Kumar, X. Huang, P. Vijayakumar and K. R. Choo, "HomeChain: A Blockchain-Based Secure Mutual Authentication System for Smart Homes," in *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 818-829, Feb. 2020, doi: 10.1109/JIOT.2019.2944400
- [13] Hammi, M. T., Hammi, B., Bellot, P. and Serhrouchni, A., "Bubbles of Trust: A Decentralized Blockchain-based Authentication System for IoT," in *ResearchGate*, June, 2018. [Online]. Available: https://www.researchgate.net/publication/326094774_Bubbles_of_Trust_a_decentralized_Blockchain-based_authentication_system_for_IoT
- [14] S. Patel, A. Sahoo, B. K. Mohanta, S. S. Panda and D. Jena, "DAuth: A Decentralized Web Authentication System using Ethereum based Blockchain," 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), 2019, pp. 1-5, doi: 10.1109/ViTECoN.2019.8899393.
- [15] S. Y. Lim, P. T. Fotsing, O. Musa and A. Almasri., "AuthChain: A Decentralized Blockchain-based Authentication System," in *International Research Journal of Engineering and Technology (IRJET)*, January. 2020. [Online]. Available: <http://ijettjournal.org/Special%20issue/CAT-2020/CATI1P212.pdf>
- [16] K. Patidar and S. Jain, "Decentralized E-Voting Portal Using Blockchain," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2019, pp. 1-4, doi: 10.1109/ICCCNT45670.2019.8944820.
- [17] N. B. Truong, K. Sun and Y. Guo, "Blockchain-based Personal Data Management: From Fiction to Solution," 2019 IEEE 18th International Symposium on Network Computing and Applications (NCA), 2019, pp. 1-8, doi: 10.1109/NCA.2019.8935049.
- [18] E. Leka, E. Kordha and K. Hamzallari. "Towards an IPFS-Blockchain based Authentication/Management System of Academic Certification in Western Balkans," 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), 2022, pp. 1448-1453, doi: 10.23919/MIPRO55190.2022.9803625.
- [19] C. Chien. "What is Rapid Application Development (RAD)?" *codebots.com*. <https://codebots.com/app-development/what-is-rapid-application-development-rad> (Accessed August. 22, 2022).

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	Study week no.: 1, 2
Student Name & ID: Wee Chun Ming, 18ACB03703	
Supervisor: Ts Dr Gan Ming Lee	
Project Title: Decentralized Authentication System Utilizing Blockchain	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Chapter 1: Introduction that include

- Problem statement and motivation
- Project scope and objective
- Contribution

Chapter 2: Review of Technology, Review of Existing Systems, Critical Remarks of Previous Works

2. WORK TO BE DONE

- Retrieved user credentials data from blockchain and compare with user password input.
- Retrieved user account related data history from blockchain.

3. PROBLEMS ENCOUNTERED

- Researching for implementation of getting user account credentials data from blockchain and compare with user password input.
- Researching for implementation of getting user account related data history from blockchain.

4. SELF EVALUATION OF THE PROGRESS

The progress is on the schedule.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	Study week no.: 3, 4
Student Name & ID: Wee Chun Ming, 18ACB03703	
Supervisor: Ts Dr Gan Ming Lee	
Project Title: Decentralized Authentication System Utilizing Blockchain	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Retrieved user credentials data from blockchain and compare with user input password and used in authentication process.
- Retrieved user account related data history from blockchain and display out the content on system frontend display.

2. WORK TO BE DONE

Recover the user deleted account from blockchain.

3. PROBLEMS ENCOUNTERED

Researching for implementation of getting user account specific data from blockchain and recover back the deleted account.

4. SELF EVALUATION OF THE PROGRESS

The progress is on the schedule.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	Study week no.: 5, 6
Student Name & ID: Wee Chun Ming, 18ACB03703	
Supervisor: Ts Dr Gan Ming Lee	
Project Title: Decentralized Authentication System Utilizing Blockchain	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Allow user delete account, system able to retrieve the deleted account data from blockchain, recover the user deleted account.

2. WORK TO BE DONE

Apply QR code as extra authentication option for user after login with email & password or with Google account.

3. PROBLEMS ENCOUNTERED

- Researching generate, download, scan QR code features.
- Use the QR code in authentication process.

4. SELF EVALUATION OF THE PROGRESS

The progress is on the schedule, but is slightly slower than planned schedule.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	Study week no.: 7, 8
Student Name & ID: Wee Chun Ming, 18ACB03703	
Supervisor: Ts Dr Gan Ming Lee	
Project Title: Decentralized Authentication System Utilizing Blockchain	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- System able to generate QR code with user account proof ID to let user download as image file.
- Apply QR code as extra authentication option for user after login with email & password or with Google account.
- System able to scan QR code to retrieve result from image file or device webcam and compare the result with blockchain specific related data.

2. WORK TO BE DONE

Chapter 3: System Methodology/Approach

- System Architecture Diagram
- Use Case Diagram and Description
- Activity Diagram

Chapter 4: System Design

- System Block Diagram
- Methodology Model

3. PROBLEMS ENCOUNTERED

Planning the diagram contents and implement to draw the diagrams.

4. SELF EVALUATION OF THE PROGRESS

The progress is slightly slower than planned schedule.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	Study week no.: 9, 10
Student Name & ID: Wee Chun Ming, 18ACB03703	
Supervisor: Ts Dr Gan Ming Lee	
Project Title: Decentralized Authentication System Utilizing Blockchain	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Chapter 3: System Methodology/Approach

- System Architecture Diagram
- Use Case Diagram and Description
- Activity Diagram

Chapter 4: System Design

- System Block Diagram
- Methodology Model

2. WORK TO BE DONE

Chapter 5: System Implementation

- Software Setup, Setting and Configuration
- System Operation (with Screenshot)

Chapter 6: System Evaluation and Discussion

- System Testing and Testing Result
- Project Challenges
- Objectives Evaluation
- Concluding Remark

3. PROBLEMS ENCOUNTERED

System testing implementation method

4. SELF EVALUATION OF THE PROGRESS

The progress is on the schedule, but is slightly slower than planned schedule.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	Study week no.: 11, 12
Student Name & ID: Wee Chun Ming, 18ACB03703	
Supervisor: Ts Dr Gan Ming Lee	
Project Title: Decentralized Authentication System Utilizing Blockchain	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Chapter 5: System Implementation

- Software Setup, Setting and Configuration
- System Operation (with Screenshot)

Chapter 6: System Evaluation and Discussion

- System Testing and Testing Result
- Project Challenges
- Objectives Evaluation
- Concluding Remark

2. WORK TO BE DONE

- Chapter 7: Conclusion and Future Work
- Poster
- Turnitin Report

3. PROBLEMS ENCOUNTERED

No.

4. SELF EVALUATION OF THE PROGRESS

The schedule is in the progress and almost done.



Supervisor's signature



Student's signature

POSTER



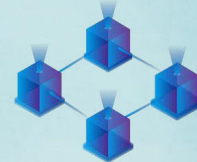
UNIVERSITI TUNKU ABDUL RAHMAN

Faculty of Information and
Communication Technology (FICT)

Student : Wee Chun Ming

Supervisor: Ts Dr Gan Ming Lee

Decentralized Authentication System Utilizing Blockchain



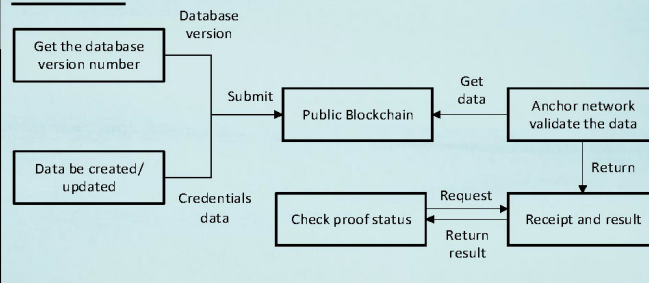
Introduction

This authentication system is aim to validate and authenticate the user identity with the help of ProvenDB blockchain database service. With blockchain technology, the user information can be immutable, decentralized, secured and privacy ensured. The blockchain database offer the database functionality but also added blockchain proofs, data provenance, versioned data and immutable data. In order to make the data become transparent and the database become public ledger, the particular data will be hashed to assure login credentials confidentiality.

Objectives

- To develop a login database utilizing blockchain
- To ensure available of database as a public ledger
- To provide confidentiality of the login credentials in the database
- To design an authentication system utilizing the blockchain database

Methods



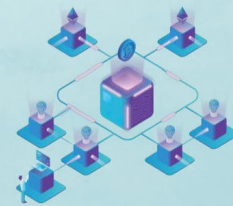
Discussion

It is an authentication system that utilizing blockchain technology. The decentralized characteristics that offered by blockchain help the proposed system distribute among all of the network peers. Due to blockchain data immutability, the ProvenDB has the number version features that provide delete and update like traditional database. This is to ensure the data integrity and immutability. For this reason, the proposed system able to gather the user info with the database number version, then submit to the blockchain. The proof result is not controlled by one node but is over half of the nodes between the network. However, although ProvenDB has the higher throughput and latency but the applied public blockchain performance still slower compare to private or hybrid types.

Results and Conclusion

- Authenticate using 3 sides of data: user input, database, blockchain.
- Able to retrieve and view data history from blockchain based on user information.
- Able to recover back old account from blockchain stored record.
- Extra authentication: QR code.

In conclusions, the applied blockchain technology able to offer much many of advantages to this project to ensure the security, transparent, immutable and privacy of the data. The ProvenDB number version and data proof ID can ensure the data integrity with the serializable consistency of versions.



PLAGIARISM CHECK RESULT

Match Overview			×
2%			
<hr/>			
<			>
1	E. Leka, E. Kordha, K. H... Publication	<1%	>
2	insightsociety.org Internet Source	<1%	>
3	eprints.utar.edu.my Internet Source	<1%	>
4	fict.utar.edu.my Internet Source	<1%	>
5	www.mdpi.com Internet Source	<1%	>
6	*Software Engineering ... Publication	<1%	>
7	www.spp.org Internet Source	<1%	>
8	hdl.handle.net Internet Source	<1%	>
9	doaj.org Internet Source	<1%	>
10	www.ukessays.com Internet Source	<1%	>
11	ph02.tci-thaijo.org Internet Source	<1%	>
12	patents.google.com Internet Source	<1%	>
13	researchr.org Internet Source	<1%	>
14	www.semanticscholar.... Internet Source	<1%	>
15	Lukas Stockburger, Ge... Publication	<1%	>

Match Overview			×
2%			
<		>	
12	patents.google.com Internet Source	<1%	>
13	researchr.org Internet Source	<1%	>
14	www.semanticscholar.... Internet Source	<1%	>
15	Lukas Stockburger, Ge... Publication	<1%	>
16	mdpi-res.com Internet Source	<1%	>
17	support.easykicks.com Internet Source	<1%	>
18	wemake-python-styleg... Internet Source	<1%	>
19	"Intelligent Systems De... Publication	<1%	>
20	Bikramaditya Singhal, ... Publication	<1%	>
21	Mohamed Tahar Ham... Publication	<1%	>
22	Papanicolas, Irene, Smi... Publication	<1%	>
23	Shibasis Patel, Anisha ... Publication	<1%	>
24	Yang Liu, Debiao He, M... Publication	<1%	>
25	www.researchgate.net Internet Source	<1%	>
26	Ajay Kumar Shrivastav... Publication	<1%	>

Turnitin Originality Report

Processed on: 07-Sep-2022 09:55 +08
ID: 1894052702
Word Count: 15962
Submitted: 1

fyp2 By Wee CHUN MING

Similarity Index		Similarity by Source	
2%		Internet Sources:	2%
		Publications:	1%
		Student Papers:	N/A

include quoted	include bibliography	exclude small matches	mode: quickview (classic) report	Change mode	print	download
<1% match (publications) E. Leka, E. Kordha, K. Hamzallari. "Towards an IPFS-Blockchain based Authentication/Management System of Academic Certification in Western Balkans", 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), 2022						
<1% match () Lim, Shu Yun, Tankam Fotsing, Pascal, Almasri, Abdullah, Musa, Omar, Mat Kiah, Miss Laiha, Ang, Tan Fong, Jsmall, Reza. "Blockchain Technology, the Identity Management and Authentication Service Disruptor: A Survey", International Journal on Advanced Science, Engineering and Information Technology, 2018						
<1% match (Internet from 27-Jul-2021) http://eprints.utar.edu.my						
<1% match (Internet from 27-Jul-2021) http://eprints.utar.edu.my						
<1% match (Internet from 27-Jul-2021) http://eprints.utar.edu.my						
<1% match (Internet from 10-Nov-2021) https://fict.utar.edu.my/documents/FYP/FYP2_template/FYP2_Report_Template_IB.docx						
<1% match (Internet from 12-Aug-2022) https://www.mdpi.com/2076-3417/11/22/11011/html						
<1% match (Internet from 03-Sep-2020) https://www.mdpi.com/1424-8220/16/12/2123/htm						
<1% match (publications) "Software Engineering and Knowledge Engineering: Theory and Practice", Springer Science and Business Media LLC, 2012						
<1% match (Internet from 12-Oct-2021)						

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)	Wee Chun Ming
ID Number(s)	18ACB03703
Programme / Course	CN (Communication and Networking)
Title of Final Year Project	Decentralized Authentication System Utilizing Blockchain

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)
Overall similarity index: <u> 2 </u> % Similarity by source Internet Sources: <u> 2 </u> % Publications: <u> 1 </u> % Student Papers: <u> N/A </u> %	
Number of individual sources listed of more than 3% similarity: <u> 0 </u>	
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.

GML

Signature of Supervisor

Signature of Co-Supervisor

Name: Gan Ming Lee

Name: _____

Date: 8/9/2022

Date: _____



UNIVERSITI TUNKU ABDUL RAHMAN

**FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)**

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	18ACB03703
Student Name	Wee Chun Ming
Supervisor Name	Ts Dr Gan Ming Lee

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 Plastic Cover (for hardcopy)
✓	Title Page
✓	Signed Report Status Declaration Form
✓	Signed FYP Thesis Submission Form
✓	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)
✓	Weekly Log
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
✓	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

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

I, the author, have checked and confirmed all the items listed in the table are included in my report.

Ming

(Signature of Student)

Date: 7/9/2022