Applied Information Technology And Computer Science Vol. 4 No. 2 (2023) 191-209


**AITCS**

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

# Trusted Transcript Management System using Blockchain

**Muhammad Irshad Ishak[1], Nur Ziadah Harun[1]***

[1]Fakulti Sains Komputer dan Teknologi Maklumat,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

DOI: https://doi.org/10.30880/aitcs.2023.04.02.012
Received 21 August 2023; Accepted 06 November 2023; Available online 30 November 2023

**Abstract**: Transcript Management System is a document management system that records student transcripts for the institution. In education, a transcript is a certified record of a student throughout study having entire enrollment history including all courses attempted, grades earned and degree. Counterfeit and forged transcripts can be a significant problem in the job market, causing difficulties for educational institutions, employers, and students. Trusted Transcript Management System using Blockchain (TTMSB) is a system solution for this problem by securely storing and verifying educational transcripts. TTMSB uses blockchain technology to create an immutable record of a student's academic achievements, improving the efficiency and security of transcript management. By using blockchain technology, TTMSB can increase traceability, improve performance, and speed in the management process, and make it easier to spot and verify fake transcripts. TTMSB is developed using the Node.js programming method and is integrated with the Polygon Testnet blockchain, ensuring a robust and efficient platform for securely managing and verifying educational transcripts. As a result, TTMSB ensures the authenticity and immutability of transcripts records, significantly enhancing the existing system between the educational institution and industry to have proper system to verify educational transcripts.

**Keywords**: Document Management System, Transcript, Blockchain, Verification, Ethereum

## 1. Introduction

Transcripts are certificates issued formally by the responsible education university to students as a prove finishing their studies. Trusted Transcript Management System using Blockchain (TTMSB) is a system that stores transcripts and a system that can verify original transcripts. Nowadays, the rising of counterfeit transcript certificates is often seen in the job industry, where people desperately forge certificates to compete in applying for a job. This system is proposed to higher educational institution and employer to counter the fake transcript in applying for a job [1]. In a recent view, forgery transcripts constitute a significant problem according to industry administration and management in physical

*Corresponding author: nurziadah@uthm.edu.my

transcript form is easy to manipulate and hardly verifiable. TTMSB is important to verify the counterfeit transcript used in the industry.

Nowadays, the management system still uses the traditional method to store data and a fragile database system. In the educational institution system, there are a lot of management systems to store transcripts but there does not have any verification system for industry and employers to verify counterfeit and fake transcripts.

Furthermore, the recent review showed that the transcript is easy to forge and manipulated by irresponsible parties [1]. The transcript which involves the educational institution and industry for the job requirement has problems during the verification process [2]. Usually, it will be costing time and finances because involving intermediaries to verify and it is hard to verify.

Trusted Transcript Management System using Blockchain is proposed to solve the problem and improve the existing system by using blockchain technology. Blockchain technology is an immutable collection of records that records all transactions and assets in a network [3]. Blockchain proposes data security by decentralized storage, having cryptographic hash functions and providing the concept that all nodes are connected to each block of transactions [3]. Thus, it can trace the original transaction which can enable one to know the counterfeit and forgery transactions. Trusted Transcript Management System using Blockchain is a transcript management system using blockchain-based. This system using blockchain will provide the system immutable and trustworthy to have a trusted transcript management system [4]. The blockchain system improves traceability and can enable counterfeit and forgery transactions to spot and verify the original transaction of the transcript. The blockchain-based system can increase performance and speed in the management system.

The main objective of TTMSB is to propose a trusted and immutable transcript management system using blockchain, to develop a trusted and immutable transcript management system using blockchain and to test the functionality of the system to verify the original transcript from the user. This system is designed for the PPA UTHM and Employer. PPA UTHM handled the transcript of the student to collect data of academic result every semester and secured in the system. On the other hand, for Employer need verification transcript system that ease them for recruiting and verifying data efficiently.

## 2. Literature Review

This section describes the literature review related to document management systems, blockchain architecture, blockchain technology and existing comparative system. The goal of literature review is to understand and discuss the project background related to management system and blockchain.

### 2.1 Transcript Management System

Transcript Management System is a document management system that records student transcripts for the institution. In education, a transcript is a certified record of a student throughout study having entire enrollment history including all courses attempted, grades earned and degree[5]. By digitalizing student transcripts in the transcript management system to achieve the utmost efficiency and desired goals. Thus, a transcript management system captures the entire education business process and makes all operations accessible, allowing schools to serve all stakeholders, students, lecturers, and administrators effectively [5]. An academic transcript may be required for various reasons, including professional, educational, or personal ones. For example, a transcript is often seen in the job industry, where people give their certificates for the requirement in applying for a job. People desperately forge certificates to compete in applying for a job.

### 2.2 Quick Response (QR) Code

QR code is a type of barcode that can be scanned using a smartphone camera or QR code reader app. It contains information in the form of a matrix of dots that can be decoded by the app to reveal the data

stored within. QR codes have become increasingly popular in recent years due to their ability to store large amounts of information in a small space [6].

One of the key advantages of QR codes is their ease of use. Users can simply point their smartphone camera at a code and instantly access the information stored within. This makes them a convenient tool for marketers and advertisers looking to provide quick access to product information or promotional offers. There are also some limitations to QR codes. One challenge is ensuring that users have the necessary technology to read the codes. Some older smartphones may not have a built-in QR code reader, requiring users to download a separate app [6].

### 2.3 Blockchain

Blockchain is a continuous block sequence that stores transaction records that are linked to one another and is online distribution ledger that records every single transaction made and it allows data or to be tracked all over the internet, so it cannot be modified or counterfeited. In blockchain, each block has its own hash value that acts as a unique identifier, making blockchain immutable. This technology is described as a trustless and fully decentralized peer-to-peer data storage platform that is spread over all the participant nodes. Blockchain technology aims at creating a decentralized environment where no third party is in control of the transactions and stores data. The blockchain serves as the basis for immutable ledgers, essentially recordings of transactions that cannot be changed, erased, or otherwise destroyed because blockchain is a decentralized and distributed database that is widely used to record every different transaction in each block that are encrypted with the hash of cryptography each of block. Blockchain has been utilized or proposed for applications for educational purposes to protect data integrity. Blockchain technology is used to record every single transaction made and it allows data or to be tracked all over the internet, so it cannot be modified or counterfeited [7]. The technology does not require the involvement of any third party, meaning that it is decentralized. Blockchain technology is a disruptive technology because it can be implemented in almost any sector requiring data immutability and transparency [4]. That means Blockchain technology has the potential to accelerate the demise of paper-based system for certificates. Up to now, the adoption of digital certificates has been held back by the ease with which they may be forged. In blockchain technology, there are three types of blockchains that must be considered according to the application requirements. Table 1 shows the type of blockchain.

**Table 1: Type of blockchain [8]**

| Type of Blockchain | Description |
|---|---|
| Public | A public blockchain is fully decentralized and permissionless, that means it is where anyone is free to join and participate. Anyone can read, write, and audit the blockchain network activities. |
| Private | This type of blockchain can participate only by invitation from authorities and it's owned by a single entity that controls participation in the network. This type of blockchain, are more efficient but offers a lower level of immutability compared to public blockchains. |
| Hybrid | This blockchain is both public and permissioned, which means that it is only public for a specific group of participants which are predetermined number of people who can participate in the consensus process, and not all users are from the same firm in terms of centralization. |

The blockchain provides a way for organizations to issue immutable digital certificates which are valid in perpetuity since their authenticity can be verified against the blockchain. The value proposition of digital certificates is greatly increased by these benefits over existing methods, which is expected to bring digital certification into the mainstream. Educational institutions are no longer required to verify credentials thanks to blockchain technology. Educational institutions won't have to devote resources to process since credentials issued on the blockchain may be automatically verified.

## 2.4 Blockchain Architecture

Blockchain is peer to peer network that charge of validating transactions, generating new blocks, and confirming the accuracy of freshly generated blocks. Most peers in the network concur on a particular network rule in order to enforce it on freshly formed blocks, but also in order to certify the legitimacy of transactions contained in a block [2]. Blockchain rule to protect data integrity, all peers must adhere to a well-known consensus mechanism that eliminates the need to determine whether a network node is trustworthy or malicious [2].

Figure 1 shows the blockchain architecture, and how the blocks are linked together that include data of previous Hash, Hash, timestamp and nonce. The blockchain is literally a chain of interconnected blocks that are linked together, each of which is connected by a pointer that is identified by the hash value of the block before it. Because the hash value of the block would change if even one bit of the data in the current block changed, this approach ensures that data in a current block cannot be altered [9]. Sequentially, it not only severs the connection between the current block and the following block, but also separates the following block from the following block, and so forth. In a decentralized peer-to-peer network, a peer can store either a full copy or a lite copy of a blockchain file, and this is known as a distributed ledger.

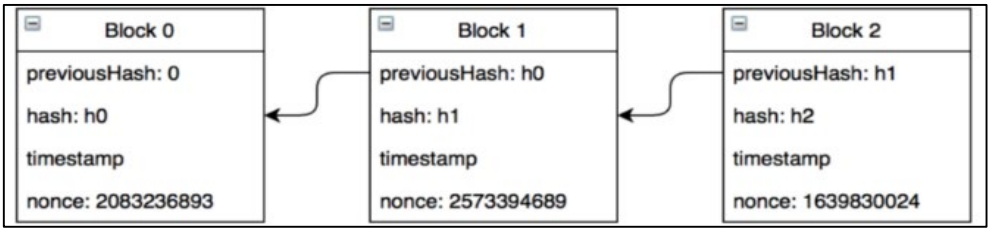


**Figure 1: Blockchain Architecture**

## 2.5 Ethereum

Ethereum is an open software platform that enables developers to build and deploy decentralized applications. Currently, Ethereum is the most advanced platform for coding and smart contracts among the current crop of cryptocurrencies Ethereum blockchain is blockchain with a built-in fully fledged Turing-complete programming language that can create contracts for the purposes of encoding arbitrary state transition functions, allowing users to create smart contracts and decentralized applications in which rules can be made according to one's desire, such as rules for ownership and transaction formats [10]. The Ethereum blockchain was used as it offers smart contract capabilities. Traditionally, the voters privacy relies on the role of a third party, who would be decrypting and tallying the votes in a verifiable manner. The role of the third party was completely removed with the implementation of blockchain as the voting and the tally computation can be without any assistance [11].

Ethereum is different from the Bitcoin blockchain as it is not entirely focused on finance like Bitcoin does. The Ethereum blockchain runs the programming code of any decentralized application. Ethereum also has its own currency which is called Ether (ETH). ETH is the native cryptocurrency of Ethereum. The main purpose of ETH is to enable a market for computation that offers participants a financial incentive to approve and carry out transaction requests and contribute computational resources to the network [12]. However, the purpose of the currency is different as Ether works as a fuel to execute smart contracts to prevent the abuse of limited network resources. On the other hand, the utilization of Ethereum's technology is essential to the currency's future. Microsoft stated that it chose Ethereum over Bitcoin because Ethereum offers the flexibility and extensibility that many of our customers were seeking for, even while a platform like Bitcoin has many fantastic uses in and of itself as a cryptocurrency. Microsoft is making the technology accessible to many more users than might otherwise utilize it by including Ethereum as part of Azure. This will encourage further inventions [12].

### 2.6 Smart Contract

Smart Contract is a self-executable, computerized transaction protocol which is useful for facilitating and verifying each contract. Smart contract has a code function that consists of a complete series of Turing operations and makes a contract with code, the code that is run by the network on the blockchain once the contract is called, so each contract is stored in a decentralized database and cannot be changed [11]. For Ethereum, the contracts are written in the Solidity programming language. The contracts are executed on the Ethereum Virtual Machine (EVM), a runtime environment for smart contracts, sandboxed and completely isolated - as the code running in the EVM has no access to the network file system [13] The code is implemented by inputting proper logic when writing the smart contract. The encrypted code is then sent to other computers or nodes via a distributed ledger. Following that, the execution of the contract is recorded. The result of this execution would be an individual agreement, and this will eliminate the manipulation of a single party [12].

Smart contracts are executed independently when validation on a transaction is carried out, to use smart contracts on objects on the blockchain, transactions must be executed to notify that there is a new contract to be entered on the blockchain and the new contract is given a unique address with a 160-bit length, and the code is uploaded on the blockchain, after the contract is completed, the smart contract consists of the contract address, the contract balance, the nonce, and the transaction id [12].

Many benefits and advantages to smart contracts. First, it is fast, efficient, and accurate. When the conditions are met, the contract executes automatically and instantly. Second, it is trustworthy and transparent. Since there are not any third parties, and it is on the blockchain, it is entirely transparent, and nobody can alter it for their benefit [12]. Finally, it is almost entirely secure. Since blockchain transactions are encrypted, it is extremely tough to hack. Additionally, since it is considered a block, and each block is connected to a chain, a malicious party would have to hack the entire chain for them to change only one record. For all these reasons, smart contracts deliver many undeniable benefits.

### 2.7 SmartCert

SmartCert is a universal cloud-based platform that streamlines the transfer of digital quality certificates by receiving, processing, and sharing them all at once. SmarCert technology is built on Microsodt Azure Blob Storage or Microsoft SQL Server which provide reliable, fast and secure platform to store data and exchanging between company. SmartCert is designed to eliminate lost paperwork and advance sustainability to protect the exchange of necessary papers between businesses. In addition, SmartCert makes use of 24/7 access and instant document sharing with supply chain partners all around the world. Certificates are stored in redundancy archived in the cloud by SmartCert, and a backup copy is kept on local servers. By offering them a free archive of the documents sent via SmartCert, it removes the chance of clients losing paperwork. Adjustments in the document and data are sent in real-time, this will eliminate the need to resend back paperwork [13]

SmartCert can creates a digital database of the quality certs, links certs and documents to a product as it moves through supply chain, connects all the required paperwork to a sales order, automates the transfer of paperwork to the customers and updates cert packages in real-time with any changes [13]. SmartCert eliminates information and document silos with one shared platform. It will upgrade from inefficient and limited local storage to a secure, cloud-based storage platform. Then, SmarCert can manage subscription and user accounts. And it designates usage and privileges by user management and permissions settings ensures maintain the integrity of cert packages.

### 2.8 Sertifier

Sertifier is an integrable software and web service that revolutionizes the conventional credentialing process by allowing its customers to create and design digital and verifiable credentials. It automates the process by enabling bulk sending via one template to all receivers [14]. Customers can always track

the status of their credentials including whether they were delivered, read, shared on social media, and the type of skill base that was developed inside the organization and sector. The certificates become digital certificates with sharing options that can be posted on LinkedIn, Twitter, or any social platform. Sertifier is a top digital credential platform that provides the most social sharing options. Certificates don't get lost in space anymore and can be tracked or interact with digital credentials.

Sertifier benefits have a professional design that design process can be completed quickly and easily with Sertifier drag and drop features certificate and open badge design function. Then, it can make a custom variable that is used for automated certificate distribution. While creating the certificate, specific earning information will be automatically included in the user design. Additionally, it features a talent library used for certificate verification [14]. It also has a skill library that is used in certificates for verification. Users can associate each certificate with the skill they have acquired. The digital badging made it easy in three steps to create, publish and track their certificate using digital badges. One of the key components of starting a successful digital badging programme is using expert badge designs. Start by designing distinctive badges that represent the user brand using the digital badge creator from Sertifier. In the next phase, users can publish in mass using digital badge systems that let the user upload the contact information of user earners and do so quickly. Earners will be notified through email and can obtain their badges by just clicking a special link. Track badge engagement in the third step. Sertifier is a magical tracker that can show the user how the brand benefits from the digital badging programme in addition to being a maker of digital badges. The brilliance of digital badging is getting insights on online presence and interaction [14].

### 2.9 Comparison with the Existing Systems

Table 2 shows the comparison between the existing system; SmartCert and Sertifier and proposed system Trusted Transcript Management System using Blockchain (TTMSB). This table compares the features and architecture of the system.

**Table 2: System's Comparison**

| Features/System | SmartCert | Sertifier | TTMSB |
|---|---|---|---|
| Blockchain | X | X | √ |
| Verification | √ (QR Code) | √ (Badges and ID Credential) | √ (QR Code) |
| Immutable | X | X | √ |
| Accountability | √ | √ | √ |
| Database | MySQL | MySQL | Ethereum |

For blockchain, both SmartCert and Certified do not use blockchain system while TTMSB use blockchain system to store the transaction of the transcript. The blockchain transaction data is immutable and cannot forge and alter.

For verification, all the systems have different methods of verification document of the document store. Both SmartCert and TTMSB use QR Code as the verification to verify the document uploaded while Sertifier use Badges and ID Credential as their verification method. As for Badges and ID Credential need to verify by asking Sertifier if the Badges and the ID Credential is legit or not. TTMSB use QR Code can be use by all people by scanning the QR images and it will redirect to the link or by scanning using autorised scanner apps to verify the data.

For immutable, both SmartCert and Sertifier are not immutable because both systems do not use blockchain to store transaction data all the documents uploaded. TTMSB is immutable because it uses blockchain to store the transaction of each uploaded document. The blockchain characteristic is immutable ledger that cannot been altered and forged.

For accountability, all the systems are accountability because SmartCert and Sertified have their established system that cover up public society certification upload. For TTMSB, the system itself is accountability that verifying and uploading process designation for UTHM community.

For databases, both SmartCert and Sertifier use MySQL for the database that stores the content of uploaded file. While TTMSB use MySQL and Ethereum Blockchain for the system. MySQL is to store the uploaded document and Etheruem Blockchain is to store the immutable transaction hash.

## 3. Methodology

This chapter explains in detail the listing approaches that are utilized in the system. The chosen model is referred to as the prototyping methodology. An appropriate model is chosen to guarantee that every step taken to complete this project goes well and it is completed on time. Additionally, the evaluation of the information received, the requirements for software and hardware, and the process for system development will all be covered. The prototyping methodology consists of six main phases which are planning, analysis, design, prototype, implementation and final implementation. The finished prototype will also serve as the foundation for the product's ultimate form [15].

The planning phase is the first step in the Prototype model's development process. The project's management structure will be built at this phase [15]. The process of choosing the project's title, organization, and supervisor is completed in this part. A proposal that includes the objective, problem statement, scope and expected result.

In analysis phase, the requirements for the proposed system are analyzed based on the academic paper related to Trusted Transcript Management System using Blockchain. There are two existing systems that have been compared, SmartCert and Sertifier are open certificate systems that have been used for the literature review. The benefits of the current systems are examined in-depth by drawing comparisons with the developed system [16]. The academic paper also analyzed includes articles, journals, and conference papers published in IEEE, Google Scholar, and Research Gate.

In the design stage emphasize on designing the system functions and qualities the system [17]. Designing the necessary database and interfaces is one of the tasks that will be done. Through the process of analysis, a logical plan is changed into a blueprint for physical implementation. Then, in this phase, all the UI was designed by using wireframes that will eventually act as the UI's design guidelines. Figma is used as the software to make the wireframe. Then, the system will be designed based on the Use Case Diagrams, Class Diagrams and Sequence Diagrams will be created as it is used to describe the database design that stores the data of the system.

In the system prototype phase, about to begin the implementation to create a prototype system. The system used NodeJS for web development backend. The system also used Ethereum environment for the blockchain system. The user interface was designed using HTML5 and node.js for the user to interact with the system through an HTML5 and JavaScript-based web page. Then, Ethereum blockchain used Solidity programming to construct and design smart contracts that run in the Ethereum Virtual Machine (EVM). The prototype system will be tested by PPA UTHM and some companies to collect feedback to overcome it.

In the implementation phase, every line of code is written and executed. Without adequate testing, delivering trustworthy software can be difficult. Additionally, once the proposed system is finished, the system will be tested to find out the identify the system flaws and bugs. During final implementation of the system as well as the testing on the developed system through the data gathered from the planning, analysis, design and system prototype and implementation phases. This phase is the final phase to deploy and integrate the system into PPA UTHM system. The final implementation is the final product to the implement and used by the users.

## 4. Analysis and Design

This section explained the system analysis and design that have been conducted for this project.

### 4.1 Context Diagram

A context diagram is a diagram that represents the entire system. The purpose of this diagram is to show the expected inputs and outputs from the system. This context diagram was created based on the user of the system perspective. There are three factors involved which are PPA UTHM, Students and Employees. Figure 2 shows the context diagram of TTMSB.

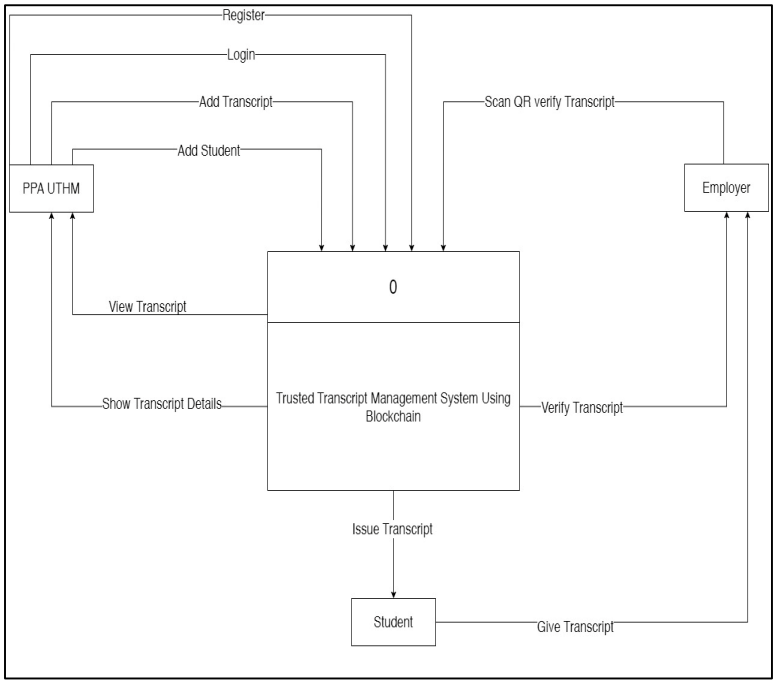


**Figure 2: Context Diagram**

Figure 2 shows the Context Diagram of Trusted Transcript Management System using Blockchain. PPA UTHM can register, login, add transcript, view transcript, and show transcript details. While Employee can scan QR code to verify transcript and verify the originality transcript. Student can get transcript and student give transcript to Employee.

### 4.2 System Architecture

System architecture defines the structure and behavior of a system. It includes the process and flow of information. The general system architecture for the proposed system is illustrated in Figure 3.

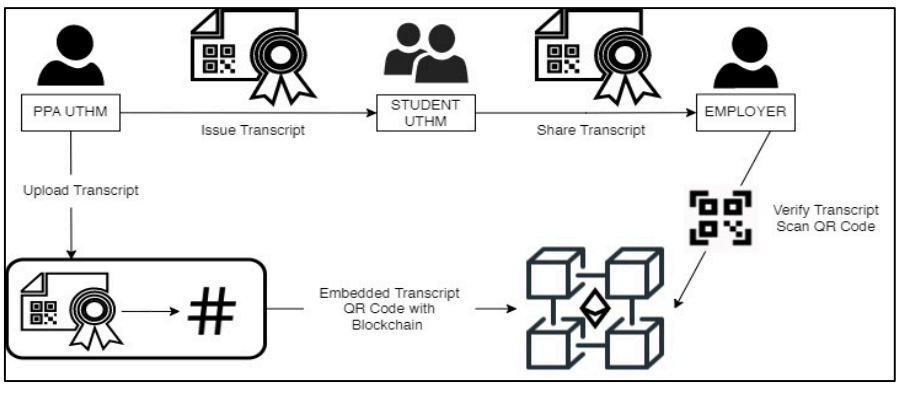


**Figure 3: System Architecture**

Figure 3 explains the user flow where there are three users involved in the system Pusat Pengurusan Akademik Universiti Tun Hussein Onn (PPA UTHM), Student UTHM and Employer. First, PPA UTHM issues the physical transcripts to Student UTHM after they ended their study. The transcript is embedded with the authenticity token that displays as QR Code. PPA UTHM also creates and uploads digital transcripts embedded with the smart contract and the authenticity token to the Ethereum blockchain system. Then, Student UTHM receives the physical transcripts issued by PPA UTHM after completing their study. Students at UTHM share their transcript with the employer to apply for the job, for one of the requirements of job recruiting. Next, the Employer receives transcripts from Student UTHM for the job application requirements. Employers need to check the originality of the student UTHM transcript to find out if it is counterfeit or not. Employer scan QR code from the transcript to verify the originality of the transcript.

## 4.2 Functional Requirements

The proposed system has four functional requirements as shown in Table 3. The login function that is for the authorized account to enter the system that only for PPA UTHM. The home function that is for the user to display the main menu of the system that only for PPA UTHM. The upload function is for the user to upload transcript that will be embedded with smart contract to the blockchain that only for PPA UTHM. The verification function that is user can verify the certificate using QR code that has authenticity token that link to the verify site and it embedded on bottom right of the transcript.

**Table 3: Functional requirements**

| Function | Functionality | User |
|---|---|---|
| Login | The system shall allow authorized account to allow login into the system. | PPA UTHM |
| Home | Users can view the main menu to use the system | PPA UTHM |
| Upload | Users can upload the transcript embedded with a smart contract | PPA UTHM |
| Verification | Users can verify the certificate using an authenticity token by scanning a QR code into the system | Employee |

## 4.3 Non-Functional Requirement

There are four categories of non-functional requirements, which are operational, performance, security, and usability, that are shown in Table 4. Non-functional requirements are crucial since they act as selection criteria for deciding between countless potential designs and final implementations.

**Table 4: Non-functional Requirement system**

| Requirement | Functionality |
|---|---|
| Operational | The system can work on a web browser. The system can work on mobile browsers.<br>Only the administrator has full access to the system. |
| Performance | The system is available for users to use 24 hours per day.<br>The system can respond to the user's interaction below or around 3.5 seconds.<br>The system can bandwidth with many users at one time. |
| Security | Only authorized users can login to the system.<br>Only authorized users can upload the file.<br>Only the admin can access the database of the system.<br>The password will be hashed.<br>The password will be encrypted with BCRYPT.<br>The file uploaded encrypted using SHA256 |
| Usability | The system is using the English language which it is easy for users to understand.<br>The system is easier to follow the step for verification. |

## 4.4 Entity Relationship Diagram

Figure 4 shows entity relationship diagram of Trusted Transcript Management System using Blockchain. The proposed system has seven entities which are Staff, Transcript, Students. There would

be a one-to-one relationship between Staff and Transcript, since many staff can issue one transcript. Then, one-to-many relationship between the Students and Transcripts entities, since one student can have many transcripts.

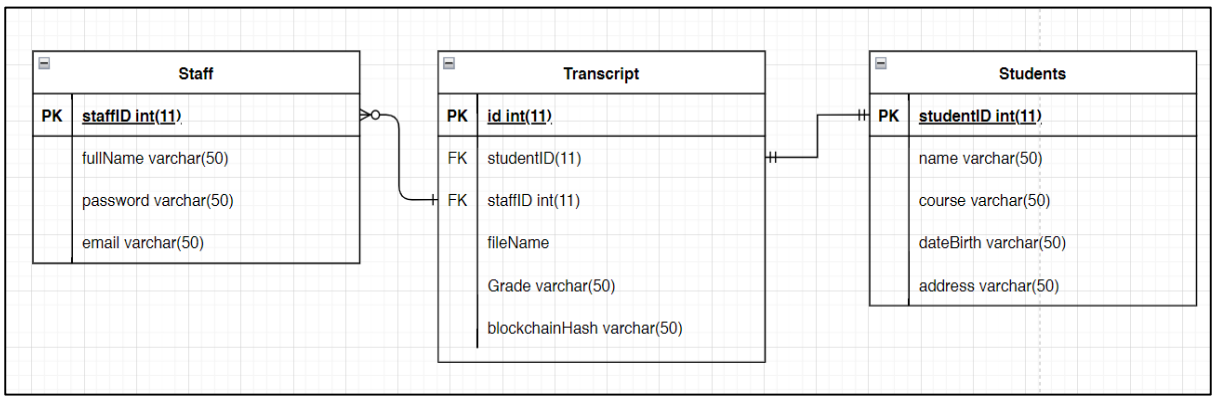


**Figure 4: Entity Relationship Diagram**

## 4.5 Interface Design

Figure 5 shows interface design of Trusted Transcript Management System using Blockchain for PPA managing Student. PPA or staff can manage students by create, read, update, delete and upload transcript student. The interface was designed using Figma.

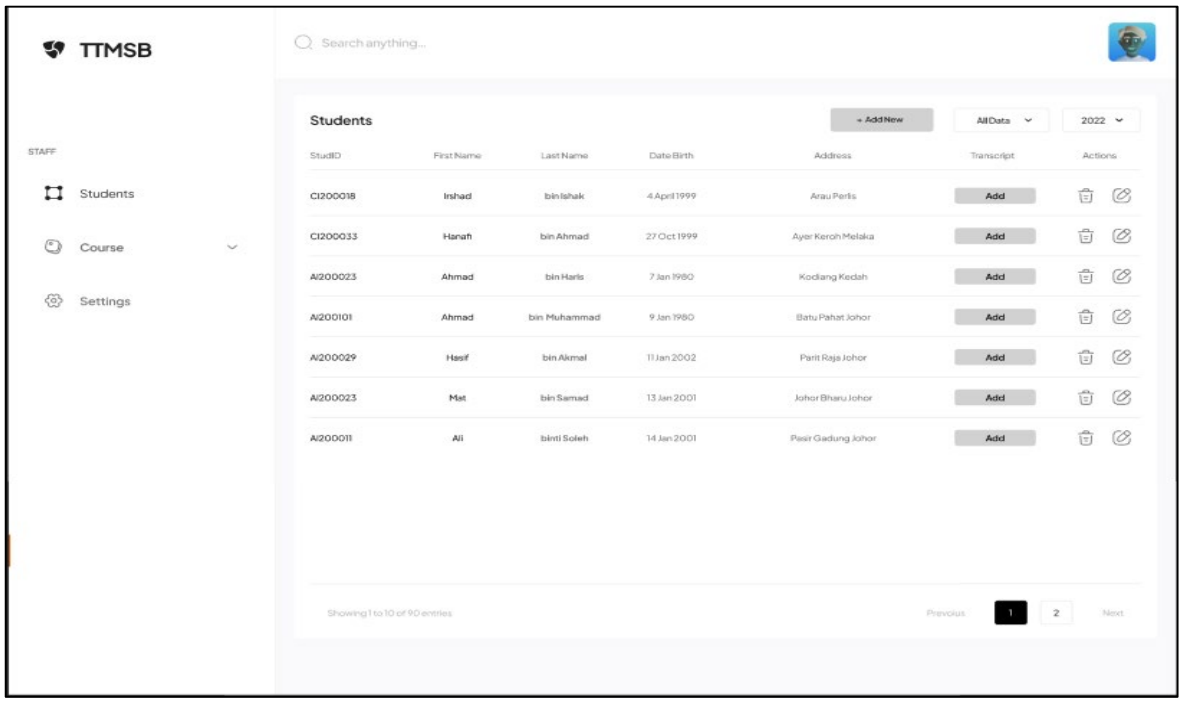


**Figure 5: Interface Design**

## 4.6 Security Test Plan

The security test plan is a security check of Trusted Transcript Management System using Blockchain. This security test plan is made to evaluate if the proposed system security is functioning as it expected to be shown in Table 5.

**Table 5: Security Test Plan**

| Checklist | Result |
|---|---|
| When user fails to login, the error message does not indicate which part of the credential data is incorrect. | Pass / Fail |
| Enforce strong password policy. Users are only allowed to input strong password during registration and password reset process. | Pass / Fail |
| Password is obscured in the textbox | Pass / Fail |
| Ensure user is not allowed to reset password using expired button or button that already been used. | Pass / Fail |
| Session is destroyed after 15 minutes of user inactivity. | Pass / Fail |
| Minimum value/length and maximum value/length in input field is specified. | Pass / Fail |

## 5. Implementation

This chapter explains in about the implementation of TTMSB. This chapter will describe the implementation of security module, implementation of Blockchain Environment and implementation of system property.

### 5.1 Implementation of Security Module

This section discusses the implementation of security modules which are safe login message, strong password, create and destroy session, secure password, hashing in database and reCAPTCHA.

Figure 6 shows the source code for encryption using BCRYPT hashing process. The BCRYPT library in nodejs is used for this module executed by calling require('bcrypt'). The BCRYPT method is bcrypt.hash(password, saltRound, function(err,hash). The password is the variable of password before the hashing process. This method requires to fill the number of saltRound or the cost factor control of the salt to encrypt the password and the system use 10 saltRound. This will make more difficult for brute-force decrypt.

```
bcrypt.hash(password, 10, (err, hash) => {
    if (err) {
      console.error(err);
      res.status(500).send('Internal Server Error');
    }
    else{
        var hashPass = hash;
    }


if (req.session.loggedin)
{
    db.query(
   "INSERT INTO staff (staffID,fullName,password,email) VALUES ('"+req.body.staffID+"'
     if (err) {
       if(err.code == "ER_DUP_ENTRY" ){
```

**Figure 6: Encryption using BCRYPT**

Figure 7 shows the source code for decryption using BCRYPT method. To the decrypt BCRYPT hashing method the syntax or method that use is bcrypt.compare(encryptPass, hash, function(err, match). The encryptPass is variable from password that fetch from database and the hash is the algorithm of bcrypt hashing include with salt. This method will compare the password whether true or false. If match the session for the login will be created.

```
See Real World Examples From GitHub
import bcrypt
export declare function genSaltSync(rounds?: number, minor?: "a" | "b"): string;

/**
bcrypt.compare(password, user.password, (err, match) => {
    if (err) {
        console.error(err);
        res.status(500).send('Internal Server Error');
    } else {
        if (match) {
            req.session.loggedin = true;
            req.session.username = username;
            if(username == 'admin')
```

**Figure 7: Decryption using BCRYPT**

Figure 8 shows the source code uses the user-provided password and measures the length of the password. A minimum of 8 characters and a maximum of 20 characters must be entered. The password characters are checked one last time to ensure that they contain at least one lowercase, one capital, one number, and one special character. If the user-provided password does not meet the system's criterion for a strong password, the user registration will be rejected.

```
<div class="form-group">
  <input type="password" class="form-control" name="password" placeholder="password *" value=""
title="Must contain at least one number and one special character and one uppercase and lowercase letter, and at least 8 or more characters"
  pattern="(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}" required="required" />
</div>
<div class="form-group">
```

**Figure 8: Strong Password**

## 5.2 Implementation Blockchain Environment

In this section show the Blockchain Environment for the proposed system. This includes implementation of Metamask setup, manage exporter, manage transcript.

Figure 9 shows setup the ownership to the Metamask. The wallet ID is included to set up the ownership to the system. The authority gives to add new exporter and to set up the system.

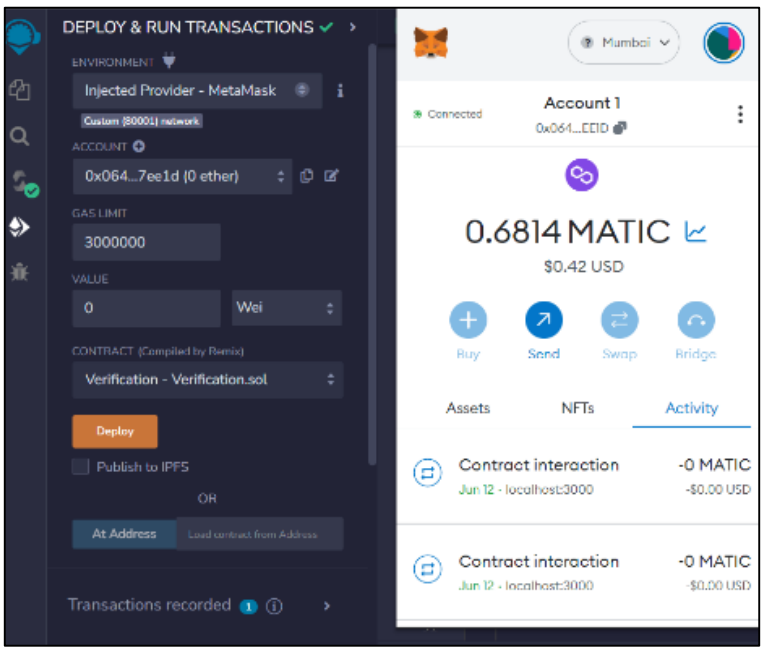


**Figure 9: Metamask Setup**

Figure 10 show source code of remix code with language Solidity that is used to create smart contract that to manage the exporter. The add_Exporter function is to add the authority to the user to upload document and store the wallet ID and author name. The alter_Exporter function is used to edit the user that has been authorized to edit author name. The delete_Exporter function is used to delete the authority by deleting the wallet ID.

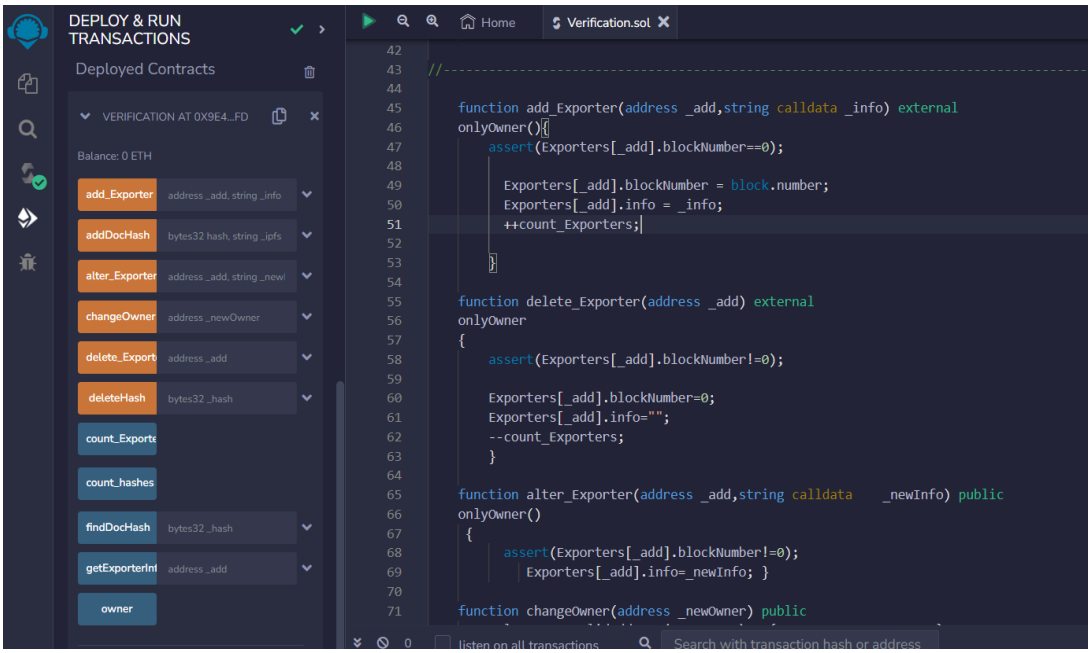


**Figure 10: Manage Exporter**

Figure 11 shows source code in Remix IDE that creates smart contract with the Ethereum Blockchain. The addDocHash function to upload transcript. The information contained in the smart contract are docHash and blockchainHash and the wallet ID. The findDocHash function is used to return data of docHash, blockchainHash and all information. This will be used when verifying transcript data when the hashDoc is same with database or the QR code link is triggered query docHash.

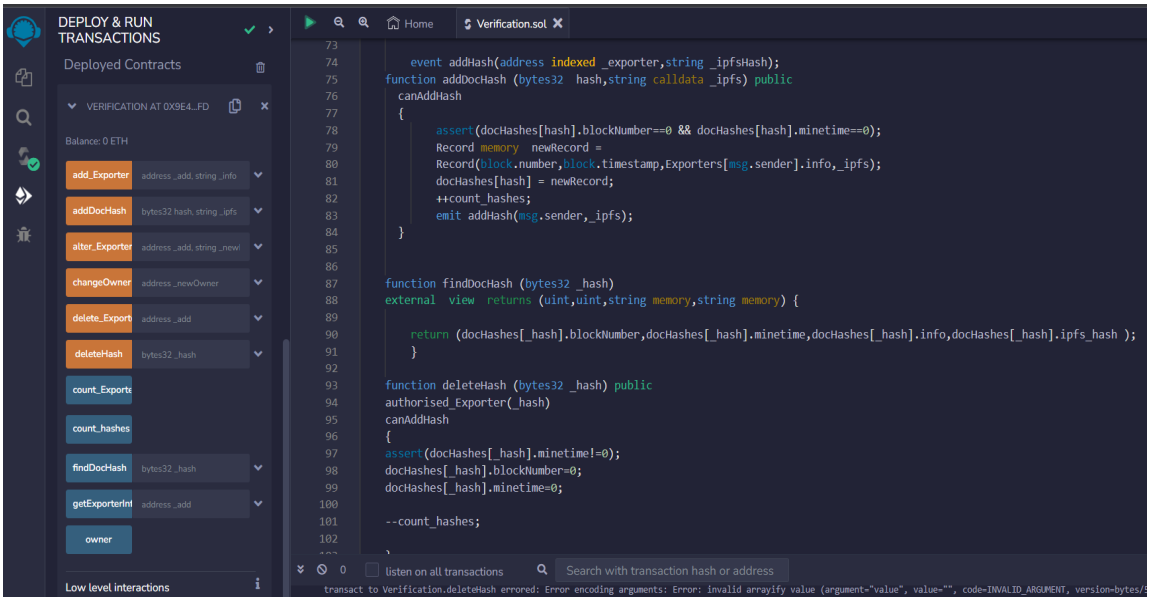


**Figure 11: Manage Upload**

## 5.3 Implementation System Properties

In this section, the system property implemented in the proposed system. This includes implementation of database, user management, student management and QR code embedded. Figure 12 shows the login function that is used to validate password by using bcrypt.compare() and if the username==admin it will redirect to admin site. If others username it will redirect to detailVIew site which is for staff authority. Then will create a session to login onto the website. Figure 13 shows the logout function that is used in the system. The session destroys function logout or destroys the current session and return to index page.

```
}

db.query('SELECT * FROM staff WHERE staffID = ?', [username], (err, results) => {
    if (err) {
        console.error(err);
        res.status(500).send('Internal Server Error');
    } else {
        if (results.length === 0) {
            // User not found
            res.status(401).send('Invalid username or password');
        } else {
            const user = results[0];

            // Compare the provided password with the stored hashed password
            bcrypt.compare(password, user.password, (err, match) => {
                if (err) {
                    console.error(err);
                    res.status(500).send('Internal Server Error');
                } else {
                    if (match) {
                        req.session.loggedin = true;
                        req.session.username = username;
                        if(username == 'admin')
                        {
                          res.redirect("/admin");
                        }else
                        res.redirect("/detailView");
                    } else {
                      res.status(401).send('Invalid username or password');
                    }
                }
            });
        }
    }
```

**Figure 12: Login Function**

```
router.get("/logout", function (req, res) {
  req.session.destroy(function (err) {
    if (err) {
      console.log(err);
    } else {
      res.redirect("/");
    }
  });
});
```

**Figure 13: Login Function**

Figure 14 shows the source code to add transcript into database. The transcript details will be added into transcript database MySQL that connects with student database as a foreign key. The transcript file will be stored in the upload folder. Figure 15 shows the source code that interact transcript to smart contract hash. The Figure 16 show the interface after uploading the transcript student,

```
const storage = multer.diskStorage({
    destination: (req, file, callBack) => {
        callBack(null, './upload/');
    },
    filename: (req, file, callBack) => {
        callBack(null, file.originalname);
    }
});


const upload = multer({ storage: storage });

router.post('/addtranscript', upload.single('doc-file'), (req, res) => {
    if (req.session.loggedin)
    {
    const filePath = req.file.path;
    const studID = req.body.studID;
    const hashDoc = req.body.hashDoc;
    const fileName = req.file.originalname;
    saveFile(filePath,studID,fileName,hashDoc, (error, result) => {
        if (error) {
            console.error(error);
            res.status(500).send({ error: 'Error saving file path to database' });
        } else {
            res.status(200).send({ message: 'File uploaded and path saved to database', data: result });
        }
    });
else
res.redirect("/login");
});


function saveFile(filePath,studID,fileName,hashDoc, callback) {
    // const query = 'INSERT INTO transcript (file_path) VALUES (?)';
    db.query("INSERT INTO transcript (fileName,filePath,blockchainHash,studID) VALUES (?,?,?,?)",[fileName,filePath,hashDoc,studID] ,(error,
        callback(error, result);
    });
}
```

**Figure 14: Add Transcript Function**

```
await window.contract.methods
  .addDocHash(window.hashedfile, window.ipfsCid)
  .send({ from: window.userAddress })
  .on('transactionHash', function (_hash) {
    $('#note').html(
      `<h5 class="text-info p-1 text-center">Please wait for transaction to be mined...</h5>`,
    )
  })
```

**Figure 15: Add Transcript Function**

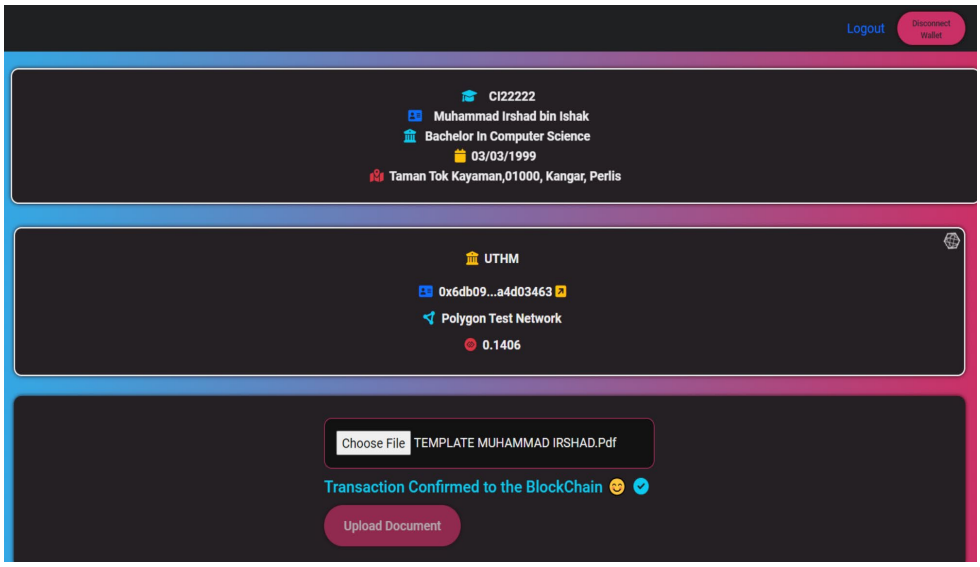


**Figure 16: Transcript Upload Interface**

Figure 17 shows the source code that embedded pdf with the QR code. The pdf-lib nodejs have been used to embed QR into pdf and relocate on the bottom of the page. Figure 18 shows the output of the transcript that has been embedded with QR.

```
const {  PDFDocument } = PDFLib

async function modifyPdf() {

  // Fetch an existing PDF document
  const file = document.getElementById('doc-file').files[0];
  const url = URL.createObjectURL(file);
  const existingPdfBytes = await fetch(url).then(res => res.arrayBuffer());
  var bytes = new Uint8Array(existingPdfBytes);
  // Load a PDFDocument from the existing PDF bytes
  const pdfDoc = await PDFDocument.load(bytes);
  // Get the first page of the document
  const pages = pdfDoc.getPages();
  const firstPage = pages[0];
    // Fetch QR PNG image
  const pngUrl = document.querySelector('#qrcode img',
  ).src;
  const pngImageBytes = await fetch(pngUrl).then((res) => res.arrayBuffer());
  const pngImage = await pdfDoc.embedPng(pngImageBytes);
  const pngDims = pngImage.scale(0.25);
          // Add a qr to to the document
 firstPage.drawImage(pngImage, {
    x: firstPage.getWidth()  - pngDims.width,
    y: 0,
    width: pngDims.width,
    height: pngDims.height,
  });
  const pdfBytes = await pdfDoc.save();

  // Serialize the PDFDocument to bytes (a Uint8Array)
  const pdfBlob = new Blob([pdfBytes], { type: 'application/pdf' });

  // Create a URL for the Blob object
  const pdfUrl = URL.createObjectURL(pdfBlob);
// Create a link element with the download attribute
const downloadLink = document.createElement('a');
downloadLink.setAttribute('download', file.name);
downloadLink.setAttribute('href', pdfUrl);

// Trigger a click event on the link element
downloadLink.click();
}
```

Figure 17: PDF Embedded Function

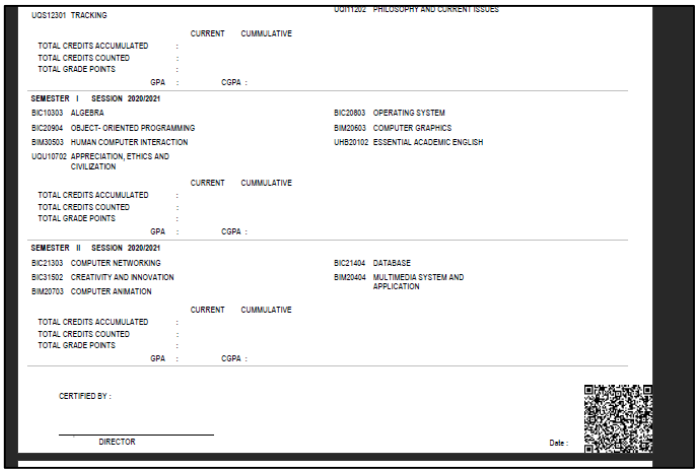


Figure 18: PDF Output Embedded QR Code

## 6. Result and Discussion

The section presents the results of the proposed system that are discussed. There are two types of testing result for the proposed system. The testing phase is performed on a whole system to verify the security and applicable requirements for the project.

### 6.1 Security Test Plan

The security test plan is made to evaluate if the proposed system security is functioning as it expected to be which shown in Table 6. The result of the security plan all the results are pass.

**Table 6: Test Security Plan**

| Checklist | Result |
|---|---|
| When user fails to login, the error message does not indicate which part of the credential data is incorrect. | Pass |
| Enforce strong password policy. Users are only allowed to input strong password during registration. | Pass |
| Password is obscured in the textbox | Pass |
| Ensure user is not allowed to reset password using expired button or button that already been used. | Pass |
| Session is destroyed after the logout | Pass |
| Password is hash using the BCRYPT algorithm in database | Pass |

## 6.2 User Acceptance Testing

The purpose of the user acceptance form is to assess the test case of the suggested system from the viewpoint of the user [18]. In the system have to use that interact with the system different ways which are PPA and Employee. The user acceptance form was collected from the PPA and Employee by using survey after use the system. The survey was conducted with QR code The user acceptance form shown in Table 7 and Table 8.

**Table 7: User Acceptance Form for PPA**

| No | Question | Result |
|---|---|---|
| | | Dissatisfied 1 -7 Satisfied |
| **Login Page** | | |
| 1 | User can login without problem | 6 |
| 2 | User can use reCAPTCHA | 6 |
| 3 | Display message easy to understand | 7 |
| **Staff Page** | | |
| 4 | User can add staff | 6 |
| 5 | User can edit staff data | 6 |
| 6 | User can view the student information | 5 |
| **Student Page** | | |
| 7 | User can add student | 6 |
| 8 | User can edit student | 6 |
| 9 | User can view the student information | 5 |
| **Transcript Page** | | |
| 10 | User can add student transcript for each student | 7 |
| 11 | User can view transcript details after upload | 6 |
| 12 | User can download transcript that embedded with QR code | 6 |
| 13 | User can verify transcript using verify link | 7 |
| 14 | User can view transcript details after verifying | 7 |
| **Overall System** | | |
| 15 | Each user can use system without problem | 6 |
| 16 | The interface easy to understand | 6 |
| 17 | The system easy to understand | 6 |

According to Table 7, the user acceptance form for PPA. The user acceptance form is a survey answer from PPA staff. The result of the questions is based on the scale guideline which is (Extreme Dissatisfied 1 -7 Extremely Satisfied). The result shows that PPA satisfied with the project where the result was very positive. The survey covers all of the modules in the system including login page, staff page, student page, transcript page and overall, of the system.

**Table 8: User Acceptance Form for Employee**

| No | Question | Result |
|---|---|---|
| | | Dissatisfied 1 -7 Satisfied |
| 1 | Each user can use system without problem | 6 |
| 2 | The interface easy to understand | 7 |
| 3 | User can verify uploaded transcript by scanning QR Code | 6 |
| 5 | User can view transcript details after verifying | 6 |
| 6 | The system shows the data of the uploaded file | 6 |
| 7 | The system can download the uploaded file | 7 |

According to Table 8, user acceptance form for Employee. The user acceptance form is a survey answer from an Employee. The result of the questions is based on the scale guideline which is (Extreme Dissatisfied 1 -7 Extremely Satisfied). The result shows that Employee satisfy with the project where the result very positive. The survey covers the verification of the transcript.

## 7. Conclusion

In conclusion, Trusted Transcript Management System Using Blockchain has completed and has fulfilled the objectives based on the project scope, system, and user requirements. Although the proposed application met the predetermined objective there are still lots of flaws and features that need to be fixed and innovated.

The system advantages are the system has verification of transcript that uploaded by scanning the QR code, the system hashed the users' password with the BCRYPT algorithm, the system enforces strong password requirement for user account and the system interact to smart contract Ethereum the transaction upload transcript.

Furthermore, there are many things that can be done after analyzing the system advantages and disadvantages of the system to improve and fix the flaws of the system. There are several things that can be fixed with the system. First, the system blockchain is using Polygon Testnet Faucet that is a testing blockchain. The system can change to use the Ethereum Blockchain or use private blockchain like Hyperledger Blockchain. Second, the interaction between the user to the smart contract in the blockchain system is needed to use Metamask Wallet. The user will complain about it, to remove this wallet the system needs to use private blockchain or use another code that uses one variable wallet ID to interact with the smart contract. There are also things that need to work out with the system, especially for the interface system. The interface system made the user feel easy and attractive to use and more professional to use the system.

## Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

## References

[1] S. Sunitha kumari and D. Saveetha, "Blockchain and Smart Contract for Digital Document Verification," *International Journal of Engineering & Technology*, vol. 7, no. 4.6, 2018, doi: 10.14419/ijet.v7i4.6.28449.

[2] T. Nguyen, "Gradubique, An Academic Transcript Database Using Blockchain Architecture," San Jose State University, San Jose, CA, USA, 2018. doi: 10.31979/etd.42nu-nsnp.

[3] N. Chaniago, P. Sukarno, and A. A. Wardana, "Electronic document authenticity verification of diploma and transcript using smart contract on ethereum blockchain," *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 7, no. 2, 2021, doi: 10.26594/REGISTER.V7I2.1959.

[4] G. Caldarelli and J. Ellul, "Trusted academic transcripts on the blockchain: A systematic literature review," *Applied Sciences (Switzerland)*, vol. 11, no. 4. 2021. doi: 10.3390/app11041842.

[5] H. S. Ahmad, I. M. Bazlamit, and M. D. Ayoush, "Investigation of Document Management Systems in Small Size Construction Companies in Jordan," in *Procedia Engineering*, 2017. doi: 10.1016/j.proeng.2017.03.101.

[6] J. Coleman, "QR Codes: What Are They and Why Should You Care?," *Kansas Library Association College and University Libraries Section Proceedings*, vol. 1, no. 1, 2011, doi: 10.4148/culs.v1i0.1355.

[7] Martin Garriga, M. Arias, Alan De Rensis, R. Li, and Y. Wu, "Blockchain based Academic Certificate Authentication System Overview," *In Proceedings of Sample Conference*, 2018.

[8] "Public, Private, Permissioned Blockchains Compared." https://www.investopedia.com/news/public-private-permissioned-blockchains-compared/ (accessed Dec. 04, 2022).

[9] W. Grather, S. Kolvenbach, R. Ruland, J. Schutte, C. Torres, and F. Wendland, "Blockchain for Education: Lifelong Learning Passport," *Proceedings of 16th European Conference on Computer-Supported Cooperative Work*, 2018.

[10] "Intro to Ethereum | ethereum.org." https://ethereum.org/en/developers/docs/intro-to-ethereum/ (accessed Dec. 04, 2022).

[11] C. Dannen, *Introducing ethereum and solidity: Foundations of cryptocurrency and blockchain programming for beginners*. 2017. doi: 10.1007/978-1-4842-2535-6.

[12] "Introduction to smart contracts | ethereum.org." https://ethereum.org/en/developers/docs/smart-contracts/ (accessed Dec. 04, 2022).

[13] "Cloud-based Digital Cert Management." https://www.smartcert.tech/ (accessed Nov. 24, 2022).

[14] "Certificates | Sertifier | The smartest certificate!" https://sertifier.com/certificates/ (accessed Nov. 24, 2022).

[15] Tutorialspoint, "SDLC Software Prototype Model," *Tutorialspoint*, 2017.

[16] Alan. Dennis, B. H. Wixom, D. Paul. Tegarden, and Alan. Dennis, "Systems analysis design, UML version 2.0 : an object-oriented approach," 2009.

[17] R. v. Stumpf and L. C. Teague, "Systems Analysis and Design with UML," *Proc ISECON*, vol. 22, no. October, 2005.

[18] T. Hamilton, "What is User Acceptance Testing (UAT)? ," *https://www.guru99.com/user-acceptance-testing.html*, 2022.