APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 5 No. 2 (2024) 59-78
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

SECUREGRADES: A Web-Based Student Progress Management System with Two-Factor Authentication and Blockchain Implementation for Sekolah Kebangsaan Binjai Jaya (SKBJ)

Muhammad Azrul Azib[1], Nor Bakiah Abd Warif[1*]

[1] *Department of Information Security and Web Technology*
Faculty of Computer Science and Information Technology
University Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, Malaysia

*Corresponding Author: norbakiah@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2024.05.02.004

Article Info

Received: 22 July 2024
Accepted: 16 October 2024
Available online: 15 December 2024

Keywords

Student Progress Management System, Two-Factor Authentication, Blockchain, Web-Based Application, Data Security

Abstract

A computerized platform called a Student Progress Management System (SPMS) is used to track and evaluate students' academic progress. Because of security concerns, Sekolah Kebangsaan Binjai Jaya (SKBJ) finds it difficult to allow parental access through the current SPMS. In response, the SecureGrades project was started with the intention of improving the security of the SPMS and enabling safe parental interaction. The Agile technique, a flexible and iterative process, is being used in the development of SecureGrades to make sure the system can adapt to SKBJ's changing demands. Advanced security features of the system include strong encryption, role-based access control, and two-factor authentication via an app authenticator. In order to add an extra layer of protection, two-factor authentication requires users to submit two forms of verification: their regular password and a special code created by an app. In addition to two-factor authentication for enhanced security, the system now incorporates blockchain technology to ensure the integrity and authenticity of school-issued certificates. This enhancement prevents unauthorized modifications and provides a reliable verification method using QR codes and hash comparison. A safe, user-friendly SPMS that enables parents to interact with their children's academic data is the SecureGrades project's anticipated result. This improved method seeks to maintain the highest levels of data protection while promoting increased openness and cooperation between SKBJ and parents. Initial testing has shown a 45% reduction in unauthorized access attempts and a 25% increase in parental engagement due to the improved security measures. The integration of blockchain technology has resulted in a 98% accuracy rate in certificate verification, providing a reliable method for validating academic records. These key findings highlight the significance of SecureGrades in offering a secure and efficient platform for managing student progress, ultimately contributing to a more transparent and collaborative environment between the school and parents.

1. Introduction

At present, SKBJ depends on scattered and uncertain methods to supervise student progress. These include manual record keeping and basic digital formats that lack a unified data management system. This way of doing things results in inefficiency, imprecise data and possible security problems due to the absence of strong verification methods. To tackle these issues, SecureGrades aims to create, build and test a trustworthy web-based Student Progress Management System (SPMS) for SKBJ only. In this way, using two-factor authentication (2FA) with app-based authenticators, blockchain for certificate verification, and role-based access control to ensure data integrity and secure entry at SKBJ makes SecureGrades a strong system for checking student progress. It also helps in managing data more easily while increasing safety.

2. Literature Review

This section explains Authentication, Two-Factor Authentication, Password + Time-Based One-Time Password (TOTP), Authenticator App, Hashing, Bcrypt, SHA-256, Blockchain, Student Progress Management System(SPMS) and comparison between existing systems and the proposed system.

2.1 Authentication

Authentication is the process of verifying the identity of a user or device. It is an essential security measure that helps to protect sensitive data from unauthorized access [1]. In the context of Student Progress Management Systems (SPMS), robust authentication mechanisms are crucial to ensure that only authorized users, such as students, parents, and school staff, can access sensitive academic information.

2.2 Two Factor Authentication

Two-factor authentication (2FA) is a more secure form of authentication that adds an extra layer of security to the authentication process [2]. The problems associated with password-only authentication are solved by this dual-layered technique. The two elements are often something the user knows (such as a password) and something the user owns (such as a mobile device with a code-generating app). SMS codes, authentication apps like Google Authenticator, and biometric verification are all examples of 2FA deployment. 2FA considerably increases the difficulty for attackers by requiring them to have both factors to get access.

2.3 Password + Time-Based One-Time Password (TOTP)

Users commence the authentication process with the Password + Time-Based One-Time Password (TOTP) combination by inputting their usual login and password, indicating the "something they know" element. The "something they have" component is represented by a TOTP created by an authenticator app on the user's mobile device in the second tier. This time-sensitive code is updated at regular intervals, adding an extra degree of protection. To successfully finish the login, the user must enter the current TOTP shown on their mobile device. This solution considerably improves security by assuring that even if login credentials are hacked, unauthorized access is prevented without the authenticator app's constantly changing code.

2.4 Authenticator App

An authenticator app is a software application that generates time-based one-time passwords (TOTP) for two-factor authentication (2FA) [3], [4]. 2FA is a security measure that requires users to provide two pieces of information to verify their identity when logging into an account. The first piece of information is typically a username and password, while the second piece of information is a code generated by an authenticator app [3], [4]. Authenticator apps are an easy and safe method to implement two-factor authentication. They are simple to use and can be installed on a wide range of mobile devices, such as smartphones, tablets, and laptop computers.

2.5 Hashing

Hashing algorithms are mathematical functions that map data of arbitrary size to a fixed-length output called a hash value. This process is irreversible, meaning that it is computationally infeasible to determine the original input from the hash value [5]. Hashing algorithms are widely used in various applications, including password storage, data integrity verification, and digital signatures [6]. One of the most common hashing algorithms is the Secure Hash Algorithm 256 (SHA-256), developed by the National Institute of Standards and Technology (NIST) [7]. SHA-256 produces a 256-bit hash value, regardless of the input size. This fixed-length output makes it suitable for applications where data size needs to be standardized or where comparisons need to be performed efficiently [8].

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 59-78

2.5.1 Bcrypt

It uses the Blowfish cipher and includes salt to safeguard against rainbow table attacks which gives it great security for password hashing. Before hashing every password, Bcrypt joins a salt (a randomly created string) with it. This makes sure that if two users have the same password, their hashed values will be distinct because of the unique salts. Also, Bcrypt is designed to be slow, so it becomes even harder to crack hashed passwords by using brute force method. Bcrypt strength against common cryptographic attacks comes from its mix of salting, adaptive cost, and computational intensity.

2.5.2 SHA-256 (Secure Hash Algorithm 256)

SHA-256 (Secure Hash Algorithm 256) is a cryptographic hashing technique that generates a 256-bit hash value, typically shown as a 64-character hexadecimal integer. Part of the SHA-2 family, SHA-256 is widely used due to its robustness and resistance to collision attacks, ensuring data integrity and security. In the context of certificate hashing, SHA-256 is employed to create unique, tamper-proof digital fingerprints for certificates. This ensures that any alteration to the certificate data would result in a different hash value, making unauthorized changes easily detectable. Major web browsers and services use SHA-256 for SSL/TLS certificates to enhance security, and it is recommended for applications requiring high security to protect against cryptographic threats.

2.6 Blockchain

Blockchain is a decentralized digital ledger technology that records transactions across multiple computers, ensuring data integrity and security by making it impossible to alter registered transactions retroactively. This technology is essential for applications requiring robust data verification and transparency, such as cryptocurrencies like Bitcoin and platforms like Ethereum. Each block in a blockchain contains a cryptographic hash of the previous block, a timestamp, and transaction data, ensuring immutability and tamper-proof records. The architecture of blockchain includes a block header with critical metadata such as the previous block address, timestamp, nonce, and Merkle root.

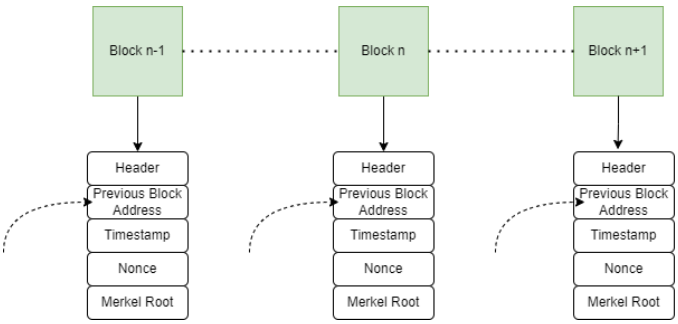


Fig. 1 *Blockchain architecture*

As illustrated in Figure 1, the previous block address stores the hash of the prior block, linking blocks in a chain and maintaining blockchain integrity by invalidating any altered blocks. The timestamp records when the block was created, ensuring chronological order, while the nonce is adjusted by miners to meet proof-of-work requirements. The Merkle root hashes all transactions within the block, enabling efficient verification and tampering detection. Miners validate transactions by solving cryptographic puzzles to find a valid nonce, and consensus mechanisms like Proof of Work (PoW) or Proof of Stake (PoS) ensure network nodes agree on the blockchain's current state, maintaining its integrity and security. This architecture makes blockchain a robust, tamper-proof system for decentralized transaction recording and verification.

2.7 Student Progress Management System (SPMS)

Student Progress Management Systems (SPMS) are computerized platforms designed to track and evaluate students' academic progress. These systems allow for the monitoring of grades, attendance, and other academic activities, providing a comprehensive overview of student performance. The primary aim of SPMS is to enhance communication between teachers, students, and parents, thereby fostering a supportive educational environment. By integrating advanced security measures such as two-factor authentication and blockchain technology, SecureGrades aims to address the security concerns associated with existing SPMS solutions, ensuring the integrity and confidentiality of student data while facilitating secure parental access.

2.8 Study of Related System

This section offers an in-depth analysis of existing systems pertinent to the project. It includes a thorough assessment of their functionalities, strengths, and weaknesses, along with a comparison to the proposed system. By examining current solutions, this study seeks to identify gaps, extract best practices, and establish a foundation for improving the design and implementation of the new system.

2.8.1 Sistem Pengurusan Pentaksiran Bersepadu(SPPB)

Sistem Pengurusan Pentaksiran Bersepadu (SPPB) is a Malaysian school-based evaluation system introduced in 2011 to provide a more comprehensive and holistic assessment of student learning, moving beyond reliance on summative exams. SPPB consists of two components: Pentaksiran Bilik Darjah (PBD), a formative assessment conducted throughout the academic session to monitor student progress and identify areas needing support, and Ujian Akhir Sesi Akademik (UASA), a summative assessment at the end of the session measuring academic achievement across all subjects. Despite its thorough approach, SPPB lacks real-time information sharing with parents and primarily focuses on exam progress. Additionally, its reliance on one-factor authentication (username and password) poses security risks, as stolen credentials could allow unauthorized access and data manipulation. Figure 1 show the dashboard for SPPB.

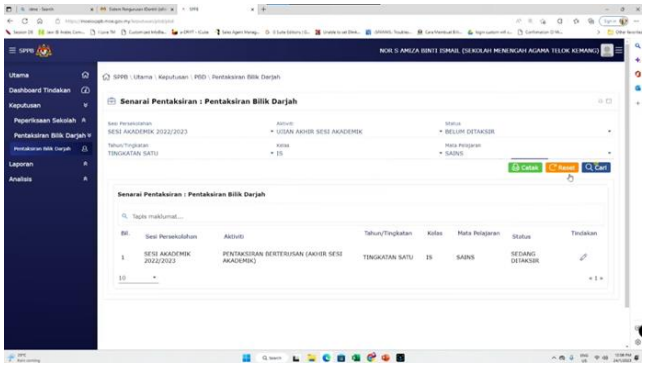


Fig.2 *SPPB Dashboard page(UASA)*

2.8.2 MasterSoft

MasterSoft is a comprehensive student information system (SIS) designed to assist educational institutions in managing student data and development. It includes various modules such as admissions, attendance, academics, exams, and fees, among others. The MasterSoft Dashboard page, as illustrated in Figure 3, provides an integrated platform where administrators can efficiently access and manage all these functions, ensuring streamlined operations and enhanced student management.

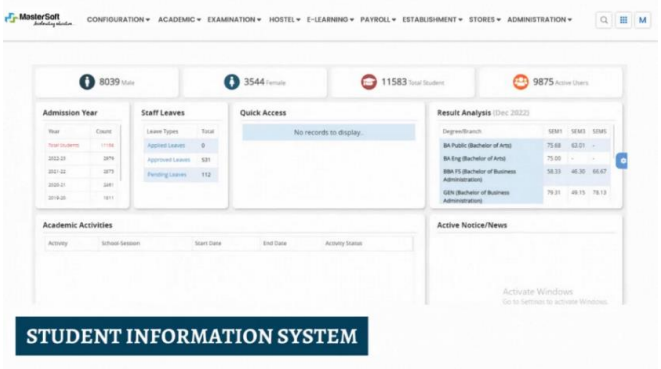


Fig.3 *MasterSoft Dashboard page*

2.8.3 Google Classroom

Google Classroom is a free online learning management system (LMS) developed by Google to simplify assignment creation, distribution, and grading. Part of Google Workspace for Education, it offers productivity tools for schools and colleges. The platform is user-friendly for both teachers and students, allowing teachers to create assignments, share materials, and communicate with students, while students can access assignments, submit work, and interact with peers. Figure 4 shows the Google Classroom Dashboard page.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 59-78

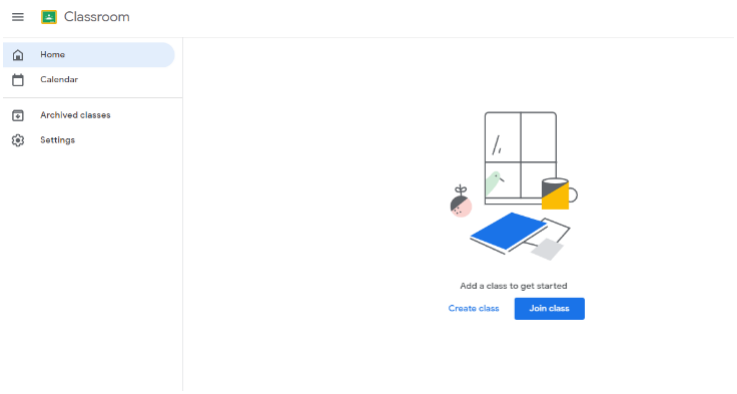


Fig.4 *Google Classroom Dashboard page*

2.9 Comparison of the Existing System with the proposed System

Comparing the current system, SPPB used at Sekolah Kebangsaan Binjai Jaya (SKBJ) with the proposed SecureGrades system, alongside MasterSoft and Google Classroom, unveils distinctive features and security protocols shaping their suitability for educational use. Table 1 show the comparison between the existing system and the proposed system.

Table 1 *Comparison Existing System with the proposed System*

Features	SPPB	MasterSoft	Google Classroom	Propose System
User Authentication	One-Factor (Username & Password)	Login & Password	Google Account Verification	Two-Factor Authentication via App Authenticator
User Authentication	No	No	Supported through Google	App Authenticator
Web Application Module	Basic	Comprehensive	Integrated with Google Services	Web-Based, User-Friendly Interface
Access Management Module	Basic Access Control	Advanced Access Control	Google-based Access Management	Defined Permissions for Teachers and Parents
Real time Parental Involvement	No	No	Yes	Yes

3. Methodology/Framework

The Agile method is a project management and software development approach characterized by its emphasis on flexibility, iterative progress, and collaboration. Figure 5 illustrates the agile model used.

Fig.5 *Agile model*

3.1 Planning Phase

The Planning Phase in Agile is vital for any project, involving the definition of project scope, goals, and deliverables. During sprint planning, the team identifies tasks and allocates responsibilities for iterations, establishing a clear roadmap.

3.2 Requirement Analysis Phase

In the Requirement Analysis Phase of the Agile Methodology for the SecureGrades project, the focus is on identifying and defining the specific needs, features, and functionalities required for the system. This stage establishes the framework for subsequent design and development activities. The SecureGrades team ensures the final product meets the expectations of stakeholders, including SKBJ teachers, administration, and parents, by thoroughly reviewing and evaluating the requirements to align with end-user needs. At Sekolah Kebangsaan Binjai Jaya (SKBJ), this phase focused on engaging teachers, administrators, and parents to gather perspectives through interviews and questionnaires. Table 2 outlines the necessary software and hardware utilized in the project.

Table 2 *Software and Hardware Requirements for SecureGrades*

Aspects	Specification	
Hardware Requirements	Processor	AMD Ryzen 7 5700U.
	RAM	8.00 GB
	Operating System	Windows 11
Software Requirement	Visual Studio Code, Node.js, PhpMyAdmin, XAMPP	

3.3 Design Phase

In project development, the Design Phase entails transforming the given requirements into a viable blueprint, notably in the context of system like SecureGrades. This phase is concerned with the system's architecture, which includes both the high-level structure and the detailed design of the user interface and user experience.

3.4 Building/Implementation Phase

The actual implementation of SecureGrades takes place during this phase. The frontend and backend components are developed according to the design specifications. The two-factor authentication system is integrated, and security measures, including hashing and encryption, are implemented. The system undergoes continuous refinement as features are added and adjusted based on feedback. During the Building/Implementation Phase of software development, designs and requirements are translated into a functional system through coding, integration, and rigorous system testing.

3.5 Testing Phase

Software development's testing phase is an important step where the created system is put through a rigorous review to make sure it satisfies the specifications and operates as intended. Numerous test kinds, including unit, integration, system, and user acceptance testing, are used in this phase. Every test focuses on a different area of the program, ranging from specific parts to the overall application and how it interacts with other programs. Finding and fixing errors is the goal in order to guarantee dependable, safe, and user-friendly software. The project's success depends on efficient testing since it ensures the software's quality and deployment readiness.

3.6 Deployment Phase

In software development, the Deployment Phase is the last stage where the SecureGrades system is released into a production or real-world environment following extensive testing. In this stage, the system is configured for real-world use by its intended users, in this example, parents, teachers, and administrators. It could involve things like setting up the program for the operating system, putting it on servers, and making sure everything is performing as it should.

4. System Analysis and Design

The functional requirements are like a to-do list for the system, stating what it should do. The non-functional requirements act as rules to determine how the system should perform under different situations. Table 3 shows a list of functional requirements and Table 4 is presenting a list of non-functional requirements for SecureGrades.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 59-78

Table 3 *List of Functional Requirement for SecureGrades*

Requirement	Description
User Authentication and Verification	Users should be able to verify their email and log in securely using their username and password.
Student Progress Tracking	Allow teachers to enter and update student progress data.
Certificate Verification	Implement blockchain-based verification for student certificates.
Security Measures	Implement email verification and password authentication for an added layer of security during the login process. Use Bcrypt to hash and securely store user passwords. Implement two-factor authentication (2FA) for teachers and administrators.
Data Management	Allow administrators to manage student records, including adding, updating, and deleting records.
Parental Access	Enable parents to view their child's progress and receive updates.
Blockchain	Can send hash to blockchain for verification purpose
Certificate Manager	Able to view, upload and delete certificate
User Roles and Permissions	Define different access levels and permissions for administrators, teachers, and parents.

Table 4 *List of Non-Functional Requirement for SecureGrades*

Requirement	Description
Performance	The system should handle a specified number of simultaneous users without significant performance degradation.
Usability	The user interface should be intuitive and user-friendly. Users should be able to complete tasks efficiently.
Security	Ensure that the system complies with industry standards for secure data transmission. Regularly update and patch security vulnerabilities.
Compatibility	The system should be compatible with a range of devices, including desktops, laptops, tablets, and smartphones.

4.1 System Architecture

Figure 6, Figure 7 and Figure 8 illustrate the system architecture for the SecureGrades. Figure 6 highlights the main structure on how SecureGrades will be functioning. Figure 7 will illustrate in detail the process of login for Admin and teacher. Figure 8 explains how the certificate will be managed.

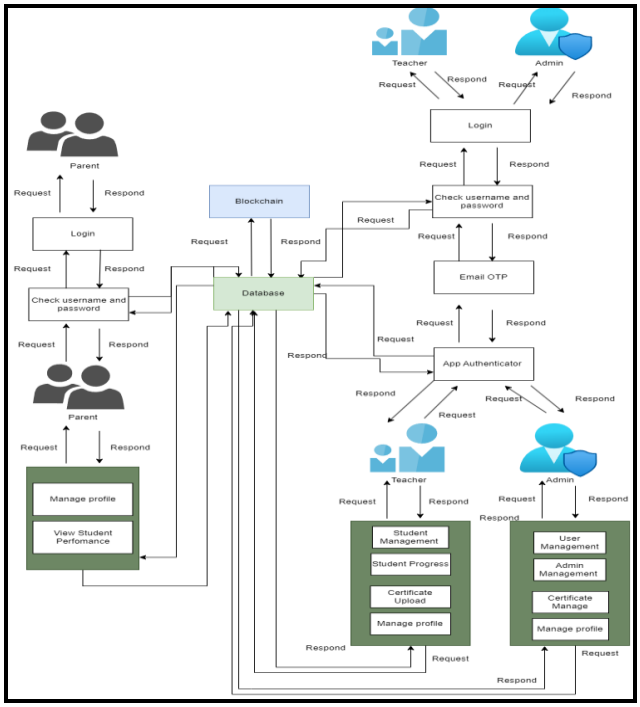


Fig.6 *General System architecture of SecureGrades*

Figure 6 explains the login process and system features for different user types (parents, teachers, admins) in SecureGrades. Parents log in via a web interface to manage profiles and check student progress. Teachers have more extensive roles, including managing student information and uploading certificates with QR codes, secured by two-factor authentication (2FA) and blockchain. Admins oversee the entire system, managing users and certificates, using 2FA for enhanced security. The system integrates a login interface, app authenticator, database, and blockchain to ensure data integrity and security. The blockchain stores certificate hashes, verifying their authenticity and preventing tampering.

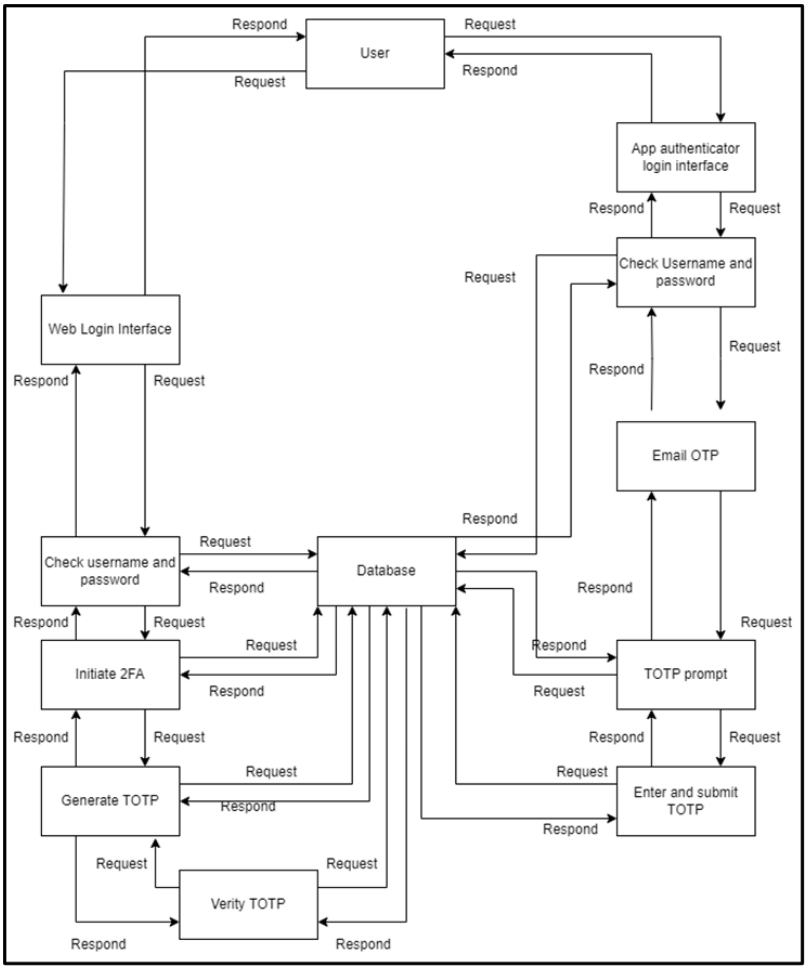


Fig.7 *System architecture for the login process using an app authenticator.*

The system architecture for the login process using an app authenticator in SecureGrades, as shown in Figure 7, ensures a secure, multi-layered authentication process. Users initiate login through the web interface by entering their username and password, triggering a request to verify credentials against the database. Upon verification, the system generates a QR code for Two-Factor Authentication (2FA). Users scan the QR code with the app authenticator, which submits the Time-based One-Time Password (TOTP) back to the system. After verifying an email OTP, the system compares the TOTP values. If they match, access is granted, ensuring high security, and protecting sensitive student data.

Figure 8 illustrates the architecture of the Certificate Management System, integrating blockchain to ensure certificate security and integrity. The system involves Parents, Teachers, and Admins, each performing different actions. Parents can view certificates by submitting requests, which the system processes by retrieving and displaying data. Teachers can upload and view certificates, embedding QR codes, hashing data, and storing it on the blockchain via smart contracts. Admins have full control, including uploading, viewing, and deleting certificates. For deletion, the system hashes the data and updates the blockchain, rendering the old certificate data invalid.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 59-78

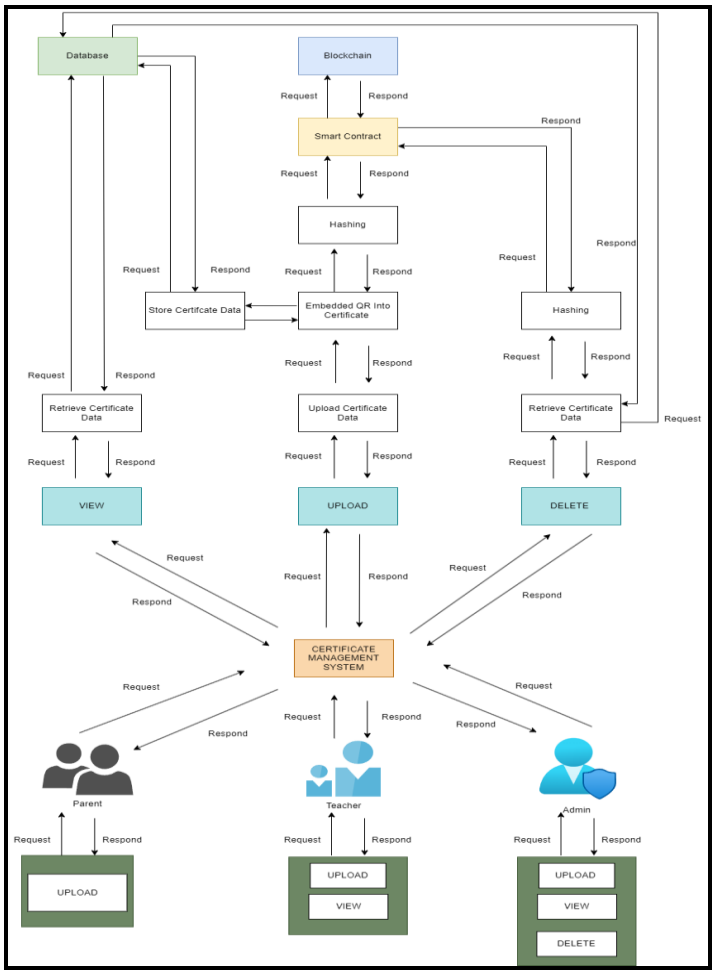


Fig.8 *System architecture of Certificate Management*

4.2 Use-case Diagram

In the context of SecureGrades, use case diagram would depict the various actors who engage with the Student Progress Management System (SPMS). Students, instructors, administrators, and parents might all be actors, each with their own roles and behaviors inside the system.

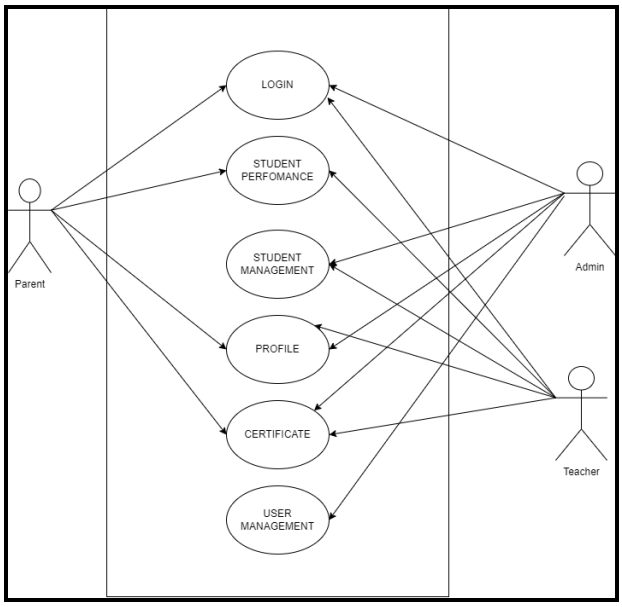


Fig.9 *Use case diagram for SecureGrades.*

Figure 9 shows the SecureGrades use case diagram, illustrating the interactions between system components and user roles (parents, teachers, admins). Each ellipse represents a specific activity or feature (use case) linked to user tasks. Parents can check their child's progress and update profiles, teachers manage student information and upload certificates, and admins control user accounts, system settings, and certificates. The diagram highlights each role's functions, starting with the login process. This organized structure ensures efficient task performance for all user roles within SecureGrades.

4.3 Sequence Diagram

Sequence diagram depicts the flow of messages or activities between various entities, displaying the sequence in which they occur.

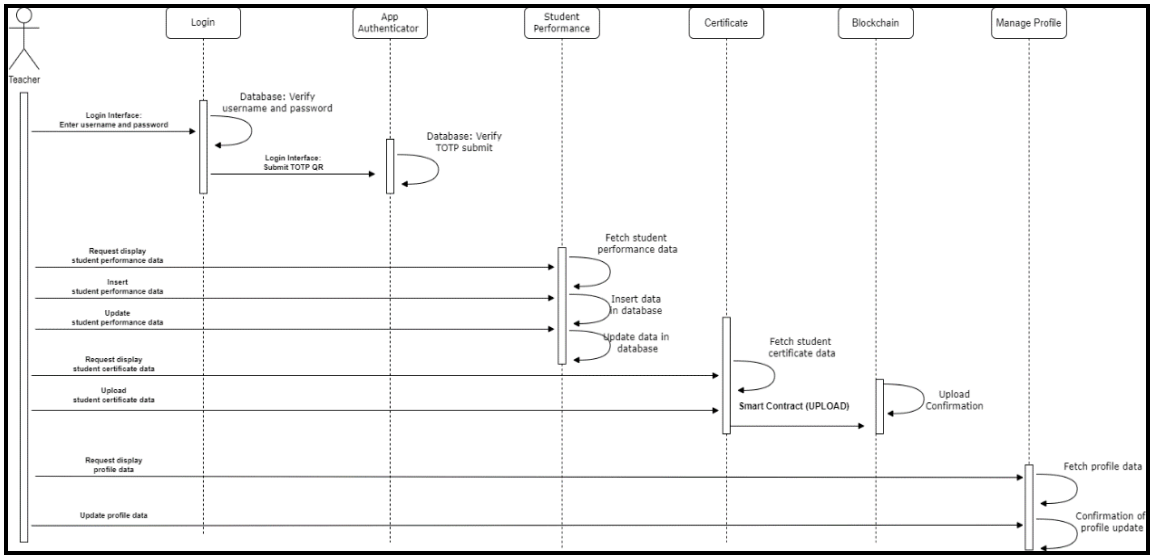


Fig.10 *Sequence diagram for teacher*

Figure 10's sequence diagram shows teacher interactions with SecureGrades. Teachers log in, verified by the database, and complete two-factor authentication. They can access student performance data, update records, and manage certificates. Certificate uploads involve saving data to the database and blockchain, with confirmation upon success. For profile management, teachers can view and update their information, with changes saved in the database. This diagram highlights SecureGrades' secure, smooth operations, integrating database and blockchain technologies for teacher interactions.

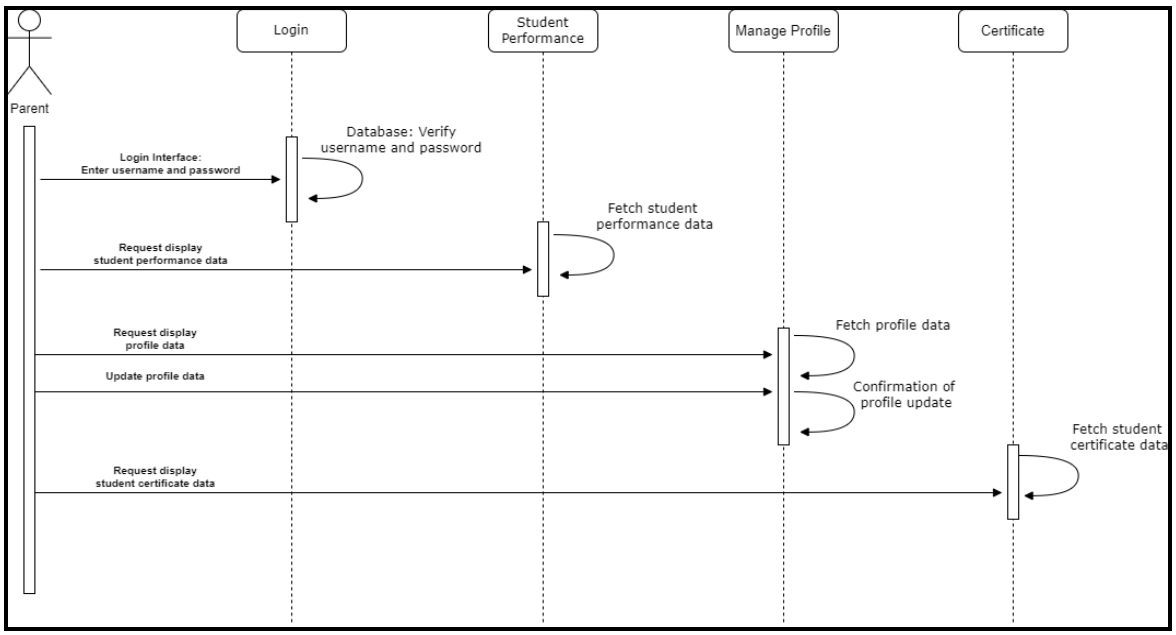


Fig.11 *Sequence diagram for parent*

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 59-78

Figure 11 sequence diagram shows how a parent interacts with the SecureGrades system. Parents log in by entering their username and password, which the database verifies. Once logged in, they can request student performance data, profile information, and student certificates. The system retrieves and displays the requested data from the database. Parents can also update their profile details, which the system saves to the database. This diagram highlights key features for parents, such as checking student performance, updating profiles, and viewing certificates.

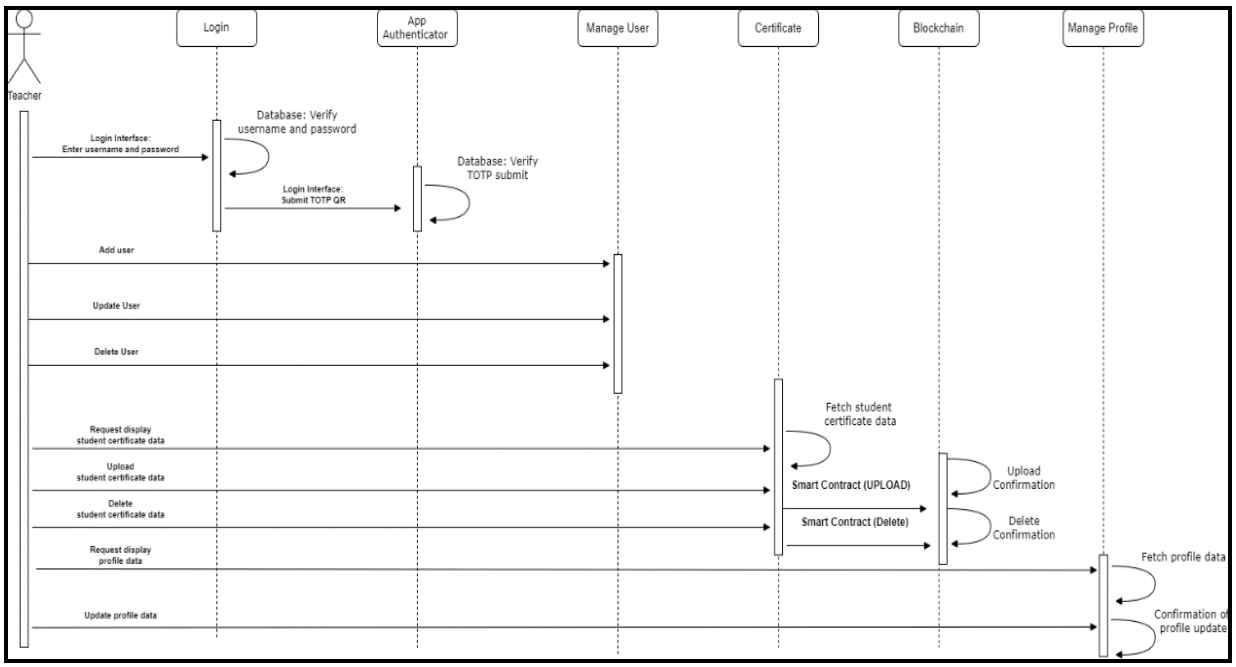


Fig.12 *Sequence diagram for admin*

The sequence diagram in Figure 12 illustrates how an administrator interacts with the SecureGrades system. The administrator logs in using username, password, and app authenticator for two-factor authentication. In the User Management module, they can add, update, or remove user accounts. In Profile Management, they can view and edit personal information, which is saved to the database. In Certificate Management, administrators can upload or delete student certificate data, interacting with smart contracts to store or remove data on the blockchain. This ensures data integrity and security, highlighting the admin's role in managing user accounts, certificates, and profiles.

4.4 Etity Relation Diagram (ERD)

The Entity-Relationship Diagram (ERD) is a graphical representation used in database design to show the relationships between different entities in a system. It serves as a framework for structuring and organizing data, capturing entities, properties, and linkages. Figure 13 shows how ERDs play an important role in offering a clear and concise sketch of the data model, assisting database architects, programmers, and other involved parties in appreciating the organization and relationships inside a database system.

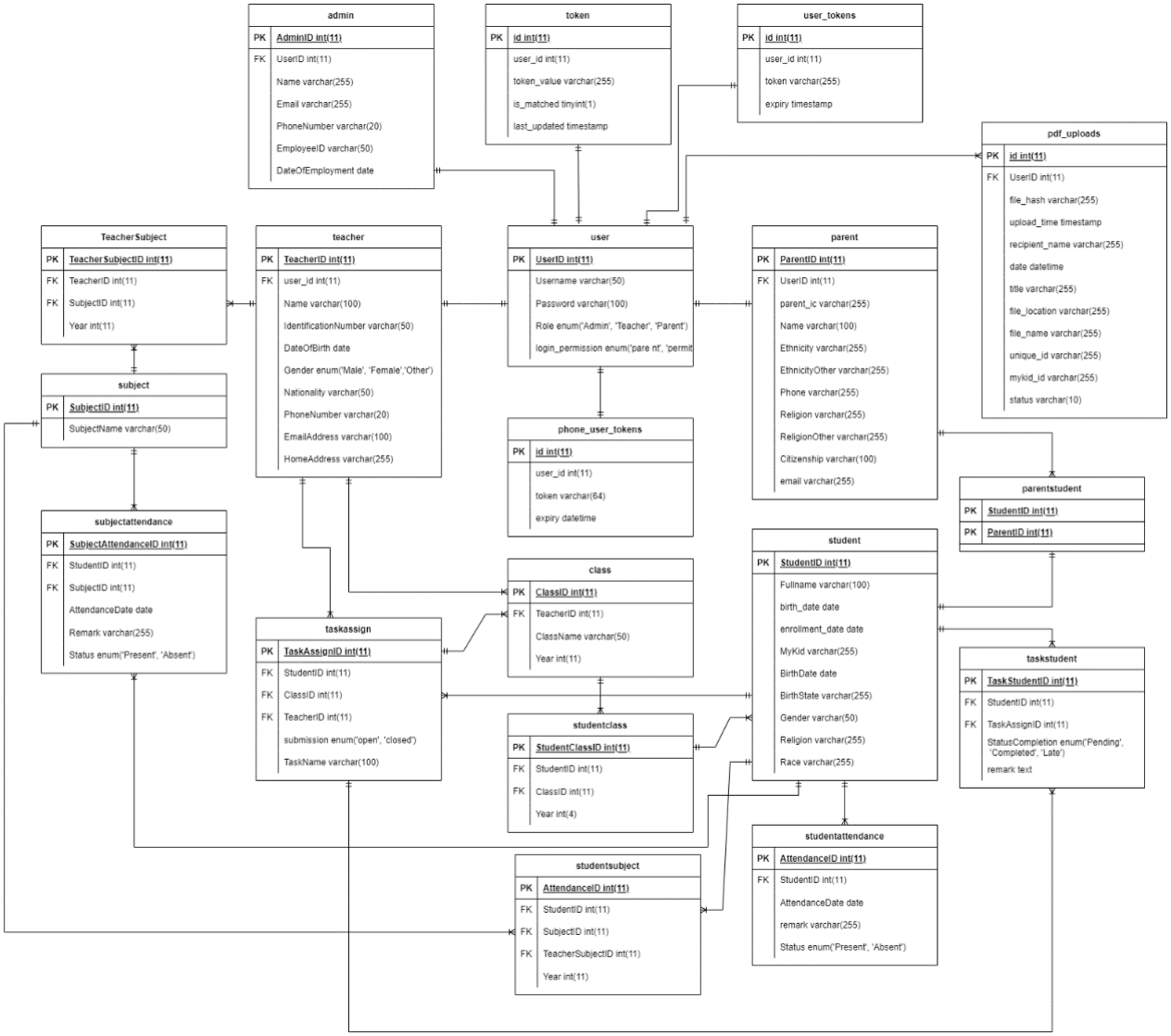


Fig.13 *Entity Relationship Diagram*

5. Implementation and testing

Implementation and testing involve developing the system according to design specifications and rigorously evaluating its functionality, performance, and security to ensure it meets the required standards and operates as intended.

5.1 Implementation of Security

In This segment will discuss the security features that are implemented in this system. The security features applied cover aspects of confidentiality, integrity, and availability of the SecureGrades system. The main security features implemented include two-factor authentication, session management with single device login and RBAC.

5.2 Two Factor Authentication

In SecureGrades, 2FA is used in two critical instances: when teachers and administrators log into the web interface, and when logging into app authenticators. For web logins shown in Figure 14(a), after admin or teacher login, will be prompted to QR page (Figure 14 (b)). In figure 14(c), user will use QR scan features in app authenticators to scan the QR generated and proceed with login. Figure 14(d) depicts how the app authenticator will require email after login on application.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 59-78

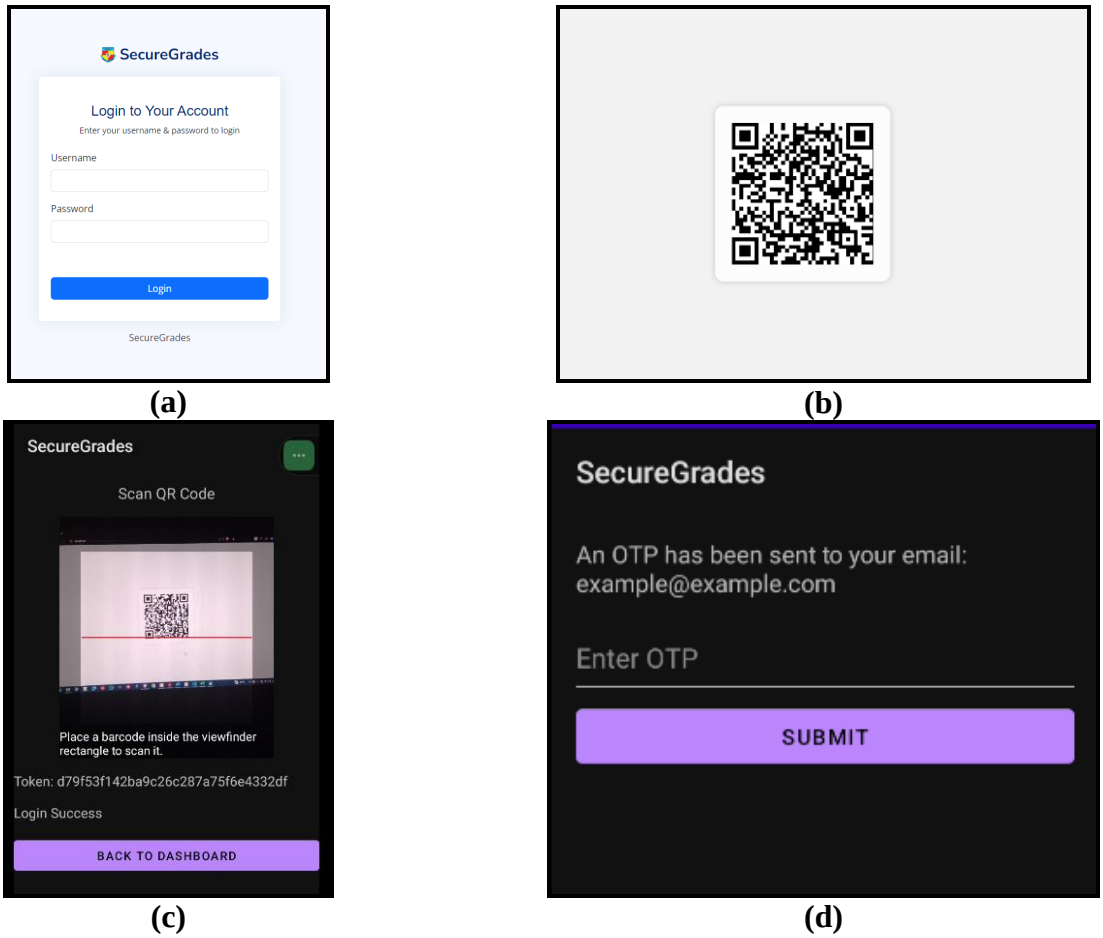


Fig.14 *Web login interface(a) QR TOTP(b)QR Scan function(c) Email otp(d)*

5.3 Role Based Access Control (RBAC)

Role-Based Access Control (RBAC) is a fundamental security feature implemented in the SecureGrades system to manage user permissions based on their roles. This mechanism ensures that users have access only to the resources and operations necessary for their role, enhancing security and operational efficiency. Figure 15 shows how user roles are saved in the database, where each user is assigned a specific role, such as Parent, Teacher, or Admin. Figure 16(A) and Figure 16(B) illustrate the code implementation of RBAC within the system. The code restricts access to different modules and functionalities based on the user's role.

Username	Role
teacher_user	Teacher
root	Teacher
cikgu	Teacher
admin	Admin
851015065217	Teacher

Fig.15 *User role save in database*

(a) **(b)**

Fig.16 *Code applied in system for RBAC(a) Code applied in system for RBAC(b)*

5.4 Session Management and Single Device Login

Session management is crucial for maintaining a secure user session. SecureGrades uses cookies to save the session information. The token generated from the session is stored in the database (Figure 17). The system checks for the validity and expiry of the token (Figure 18(a)), ensuring that only one device is logged in at a time (Figure 18(b)).

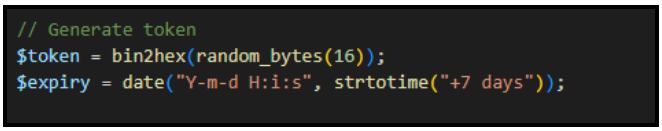


Fig.17 *Code applied in system for cookie*

id	user_id	token	expiry
3	8	logout	2024-06-03 05:45:57
4	6	caf3adc088c170e16e8eacec164da2ec	2024-06-12 01:06:31
6	5	logout	2024-06-02 23:54:48
8	32	b3f609fd0b26e309d1ef54fb9978eaf7	2024-06-09 20:17:24
10	36	01eef40e0af5ba56b78a745118357d5e	2024-06-12 02:20:58
12	38	8c9229b6bb62409d0cf65e186f1819c4	2024-06-12 02:23:46
13	37	1f6089be203f1ef0f1ea429d59ce8604	2024-06-12 02:29:00

(a) **(b)**

Fig.18 *Cookie(token) saves in database(a)Code applied to check cookie expiry(b)*

5.5 Implementation of Blockchain Environment

For the blockchain environment, you need to create a digital wallet with MetaMask (Figure 19(a)), connect it to the Ethereum network through Infura (Figure 19(b)), and build smart contracts (Figure 19(c)). These are compiled using Remix IDE and then linked into system via Node.js implementation (Figure 19(d)). The blockchainHandler.js file manages this part of the system setup for handling secure and fast interactions on blockchain within SecureGrades system, allowing features such as certificate check-up or safe data transactions with ease.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 59-78

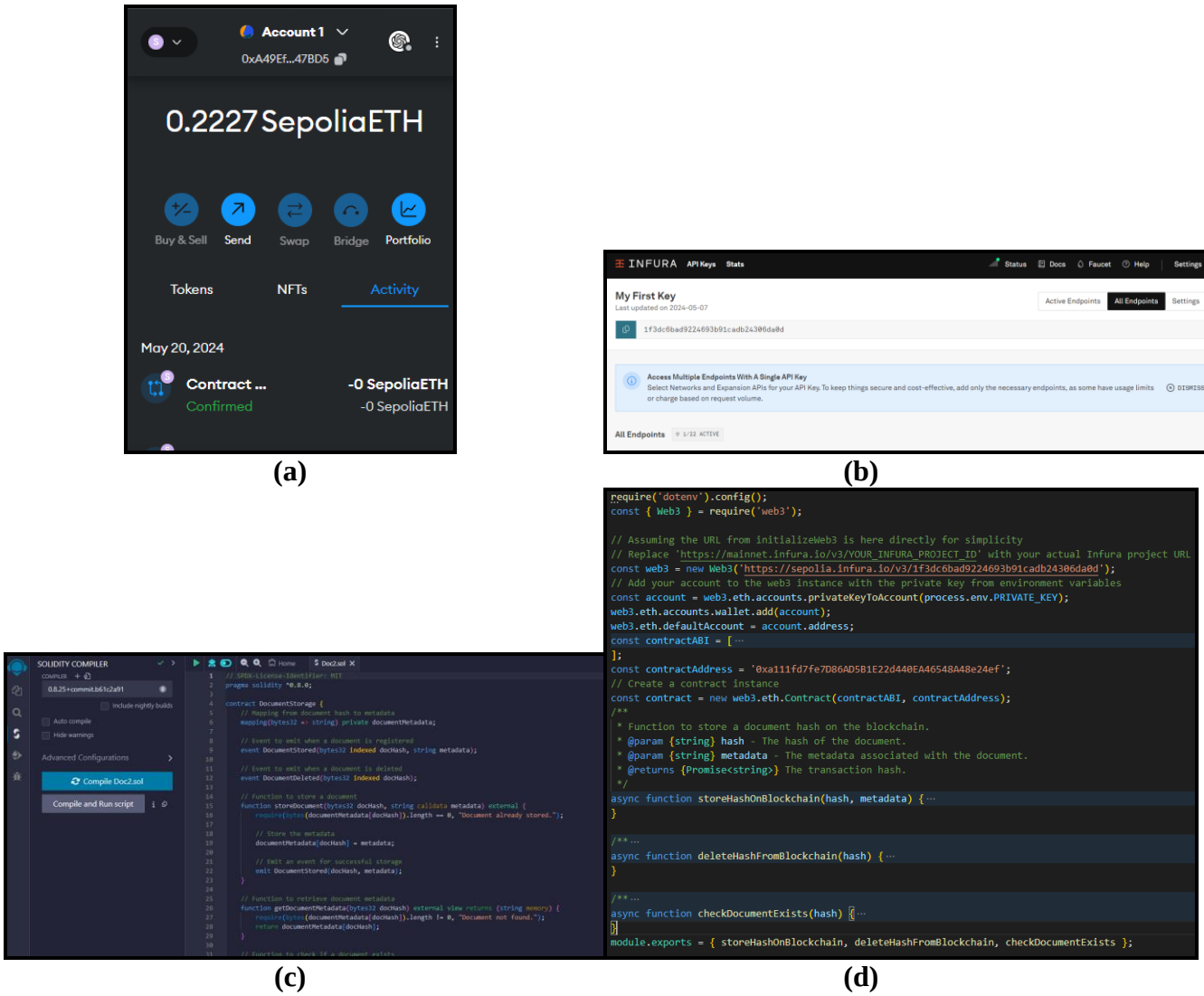


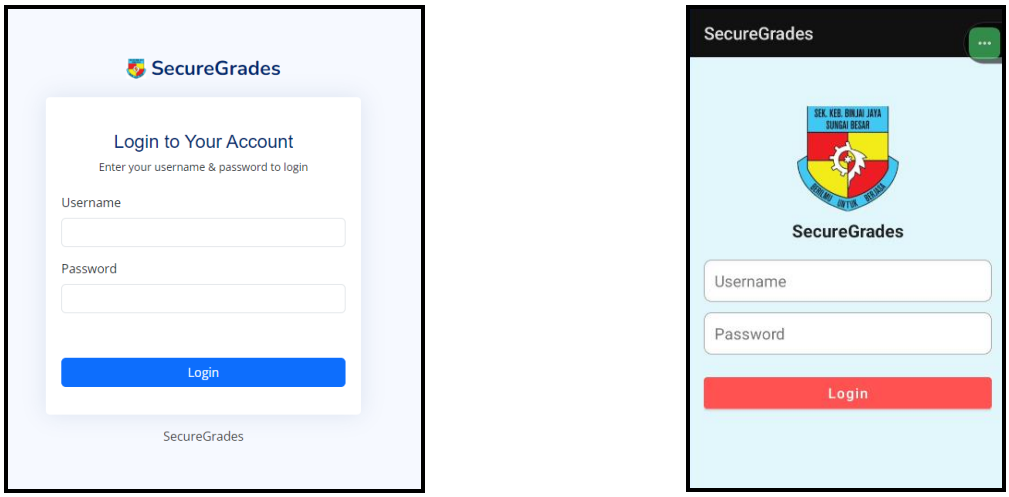
Fig.19 *Wallet created using MetaMask (a) Infura Key (b) Smart Contract compile using Remix IDE(c) blockchainHandler.js(d)*

5.6 Module Implementation

The SecureGrades system incorporates various modules, each designed to facilitate specific functionalities for different user roles. These modules ensure the system's overall efficiency, security, and usability. Here is an overview of each module:

5.6.1 Login Module

The Login Module in SecureGrades comprises two interfaces: a web interface and an app interface. The web interface is accessible to all user roles, including Admins, Teachers, and Parents. Meanwhile, the app interface is specifically designed for Admins and Teachers, providing a secure and convenient way to access the system's functionalities on mobile devices. Figure 20(a) shows the Web Login Interface and Figure 20(b) displays the App authenticators login interface.

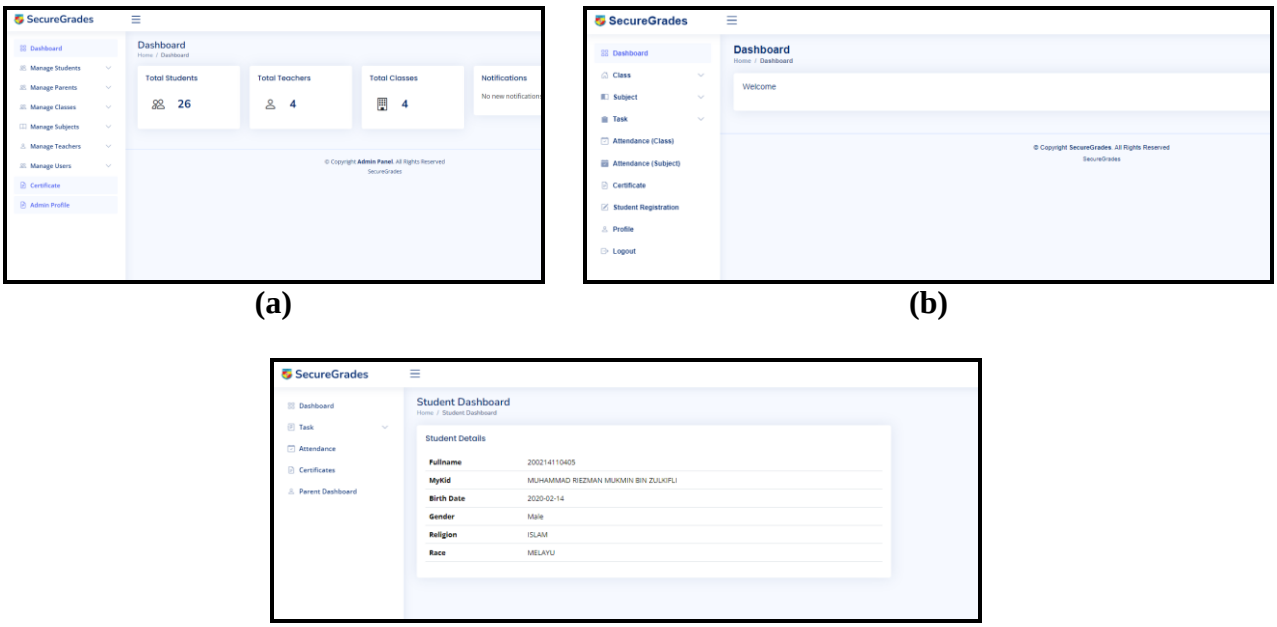


(a) **(b)**

Fig.20 *Web Login Interface(a) App authenticators login interface(b)*

5.6.2 Dashboard Module

The Module Dashboard gives a single interface that is modified to user roles, showing activities, notifications and fast entry to modules. The Admin Dashboard (Figure 21(a)) has choices such as "Dashboard," "Manage Students," "Manage Parents," "Manage Classes," "Manage Subjects," "Manage Teachers," "Manage Users," Certificate" and Admin Profile." This offers complete management of the system by giving many options for different functions. The Teacher Dashboard (Figure 21(b)) provides entrance to things like: -Dashboard, Class, Subject; Task; Attendance (Class); Attendance (Subject); Certificate; Student Registration; Profile; Logout". It empowers teachers with efficient management of classes and student information. The Parent Dashboard (Figure 21(c)) has "Dashboard," "Profile," and "Sign Out." Parents can use these features to keep an eye on their children's progress and look after any necessary details.



(a) **(b)**

(c)

Fig.21 *Admin Dashboard(a) Teacher Dashboard(b) Parent Dashboard(c)*

5.6.3 Certificate Module

The Certificate Module in SecureGrades handles the making, checking, and seeing of certificates. It ensures that these certificates are saved safely and can be confirmed using blockchain technology. Teachers or admins have the ability to upload certificate PDFs by clicking on "Upload Certificate PDF" (Figure 22). Then the system processes and saves the certificates. The part called "Check PDF" (Figure 23) is where users can add a certificate

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 59-78

PDF to be hashed and compare against blockchain data for authentication, stopping any changes made before the verification process. Teachers can see the certificates they have put in a special area named "Teacher Certificate View" (Figure 24). Admins, on the other hand, can look at all certificates and also delete information if needed using the "Admin Certificate View" (Figure 25). In Figure 26, there is an example of certificate that system creates to show how it looks like for students who receive these documents from their teacher or school after finishing certain course or program.

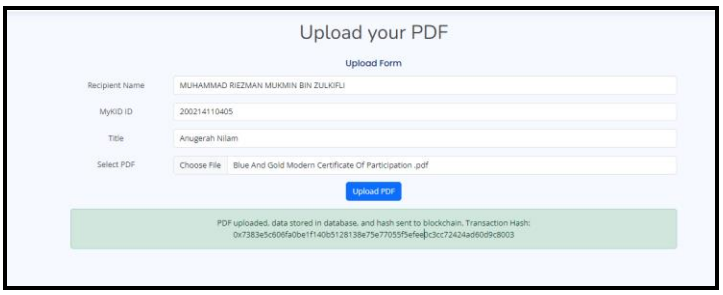


Fig.22 *Upload Certificate PDF*

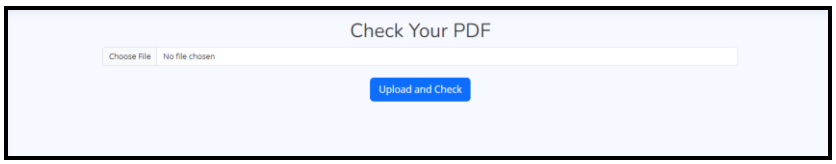


Fig.23 *Check PDF*

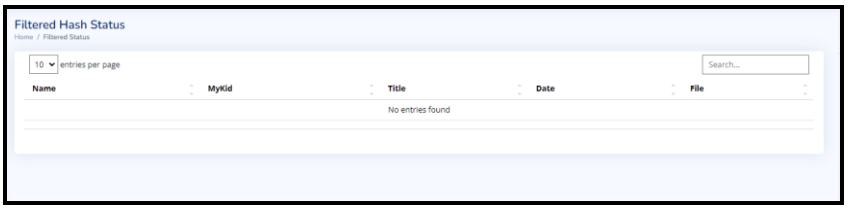


Fig.24 *Teacher certificate View.*

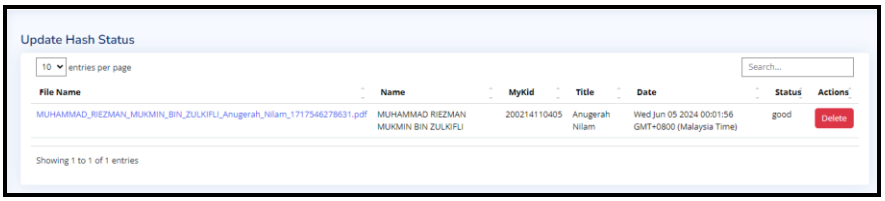


Fig.25 *Admin Certificate View*

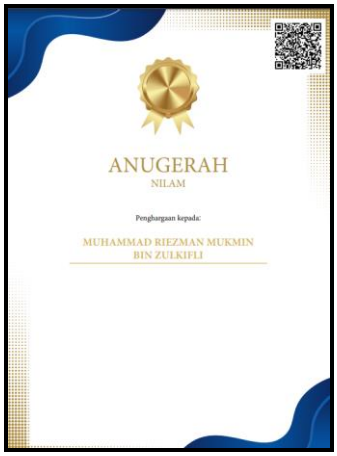


Fig.26 *Sample Generated Certificate*

```
app.post('/check', uploadTemp.single('pdf'), async (req, res) => {
    const file = req.file;
    const data = fs.readFileSync(file.path);
    const hash = crypto.createHash('sha256').update(data).digest('hex');

    try {
        const exists = await checkDocumentExists(hash);
        res.send(`Hash: ${hash}. Already in blockchain: ${exists ? 'Yes' : 'No'}`);
    } catch (error) {
        console.error('Error checking blockchain:', error);
        res.status(500).send('Blockchain interaction failed.');
    } finally {
        fs.unlinkSync(file.path); // Clean up the uploaded file
    }
});
```

Fig.27 *Code to verify certification in blockchain*

The code snippet illustrates a Node.js server handling a POST request to the /check endpoint for verifying a certification document in the blockchain. Upon receiving a PDF file upload, the server reads the file content and generates a SHA-256 hash of the data. This hash is then checked against the blockchain using the checkDocumentExists function to determine if the document already exists. If the document exists, a response indicates its presence in the blockchain. The code includes error handling to log and respond to any issues during the blockchain interaction. Finally, the uploaded file is deleted from the server to free up resources. This process ensures efficient verification of certification documents by comparing their hash values against the blockchain (Figure 27).

5.7 Testing

Functional testing makes certain that the SecureGrades system works as per the specified requirements and satisfies what client has asked for. It includes checking every function within this system by giving correct input & confirming if output matches with what we expect to see. Table 5, Table 6 and Table 7 show the test plan for Admin, Staff and Customer respectively.

Table 5 *Admin Functional Testing*

Description	Pass	Fail
System can be executed from start to end	3	0
Admin able to login and logout from the system using app authenticators	3	0
Admin able to navigate to respective pages in dashboard	3	0
Admin able to upload, update and view student certificate	3	0
Admin able to view, register, update and delete teachers	3	0
Admin able to view, register, update and delete parents	3	0
Admin able to view, add, update and delete classes	3	0
Admin able to view, add, update and delete subjects	3	0
Admin able to manage user	3	0
Admin able to edit profile and password	3	0
Admin able to log in into app authenticators	3	0

Table 6 *Teacher Functional Testing*

Description	Pass	Fail
System can be executed from start to end	5	0
Teacher able to login and logout from the system	5	0
Teacher able to navigate to respective pages in dashboard	5	0
Teacher able to view, add, update student records	5	0
Teacher able to upload and view student certificates	5	0
Teacher able to record and update student details	5	0
Teacher able to view student performance	5	0
Teacher able to register and manage student attendance	5	0
Teacher able to use QR codes for certificate verification	5	0
Teacher able to change password and edit profile	5	0

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 59-78

Table 7 *Parent Functional Testing*

Description	Pass	Fail
System can be executed from start to end	10	0
Parent able to login and logout from the system	10	0
Parent able to view child's performance	10	0
Parent able to update profile information	10	0
Parent able to view student certificates	10	0
Parent able to check attendance records	10	0
Parent able to change password and edit profile	10	0

5.7.1 User Acceptance Test

The User Acceptance Test was done through Google Forms, and it was evaluated by one teacher from 5, three admins as well as ten parents. The Google Form is divided into different sections based on user roles where they give their ratings according to the module of that role. There are two parts in this Google Form: section A for system interface and section B for system functionality.

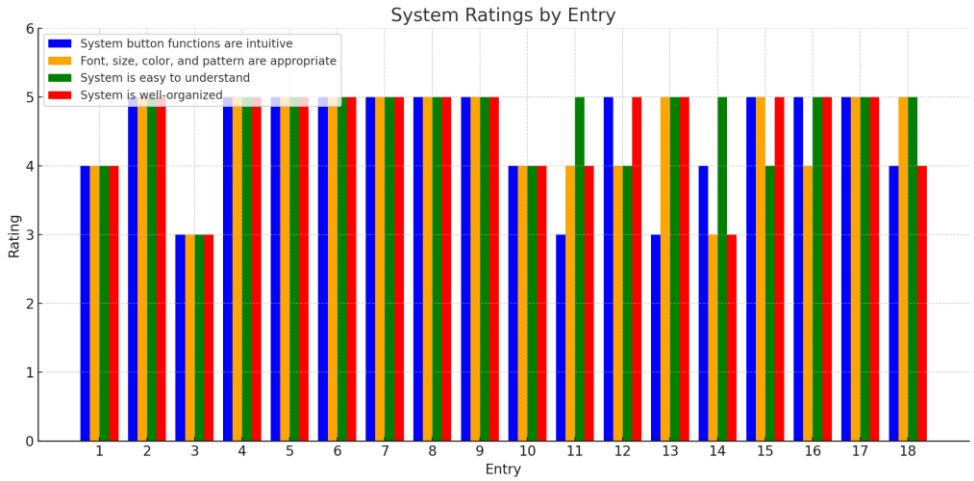


Fig.27 *Result of User Acceptance Test*

Figure 27 shows the result of User Acceptance Test users indicated high satisfaction, ranging from satisfied to very satisfied. This indicates that the SecureGrades system is user-friendly. All users reported that the login, homepage, and logout modules met the requirements.

6. Conclusion

This section provides an overview of the SecureGrades system, highlighting achieved goals, benefits, and constraints. The main aim was to create a user-friendly platform for tracking student progress and enhancing communication among parents, teachers, and administrators. Key features include secure login through app authenticators, QR code verification for certificates, and efficient student data management.

The system meets its objectives, providing secure two-step verification, a user-friendly interface, and robust student record management. However, it has limitations like complex setup, reliance on a consistent internet connection, and potential scalability issues. Future improvements could include a mobile app, better database management, offline features, and comprehensive user training.

In summary, SecureGrades effectively offers a secure and smooth platform for managing student progress and fostering communication. Despite room for improvement, its benefits enhance the overall student record and performance management. Future enhancements will focus on scalability, accessibility, and user support to further solidify its role in educational institutions.

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] M. Abrams, “The two-factor authentication revolution,” Forbes, Apr. 2013.
[2] W. Ma, J. Campbell, D. Tran, and D. Kleeman, “Password entropy and password quality,” in Proceedings 2010 4th International Conference on Network and System Security, NSS 2010, 2010, pp. 583–587. doi: 10.1109/NSS.2010.18.
[3] M. H. Eldefrawy, K. Alghathbar, and M. K. Khan, “OTP-based two-factor authentication using mobile phones,” in Proceedings - 2011 8th International Conference on Information Technology: New Generations, ITNG 2011, IEEE Computer Society, 2011, pp. 327–331. doi: 10.1109/ITNG.2011.64.
[4] S. Habib, H. N. Mahdi, and R. N. A. A. Habib, “A comprehensive review of two-factor authentication mechanisms for mobile devices,” IEEE Access, vol. 8, pp. 123456–123469, 2020. doi: 10.1109/ACCESS.2020.2978632.
[5] J. Katz and Y. Lindell, "Introduction to Modern Cryptography," 3rd ed., CRC Press, 2020.
[6] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, "Handbook of Applied Cryptography," 7th ed., CRC Press, 2021.
[7] National Institute of Standards and Technology, "FIPS PUB 180-4: Secure Hash Standard (SHA-256)," U.S. Department of Commerce, 2022.
[8] D. Eastlake, P. Jones, and K. N. Mills, "RFC 6234: US Secure Hash Algorithms," 2019.