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

AITCS

Vol. 5 No. 2 (2024) 371-387
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Development of Textbook Borrowing System Using Web-based Approach

Liaw Wan Qing[1], Hairulnizam Mahdin[2]*

[1,2] *Fakulti Sains Komputer dan Teknologi Maklumat,*
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: hairuln@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2024.05.02.020

Article Info

Received: 13 june 2024
Accepted: 28 September 2024
Available online: 15 December 2024

Keywords

Textbook Borrowing System, SPBT (Skim Pijaman Buku Teks), Web-based, Waterfall method

Abstract

The Textbook Loan Scheme Program (SPBT) in Malaysia has provided educational resources since 1975. At SMK Bekok, the manual textbook management system, overseen by Coordinator Teacher Miss Pang, results in inefficiencies, inaccuracies, and tracking challenges. To address these issues, a modern Textbook Borrowing System is proposed to streamline processes, ensure accurate record-keeping, and enable real-time tracking. Utilizing the Waterfall Model of the Software Development Life Cycle (SDLC), the project follows a structured approach through planning, analysis, design, implementation, testing, and maintenance phases. The expected outcome is a web-based system that automates manual processes, enhances record accuracy, and simplifies borrowing and returning textbooks, ultimately creating a more efficient and conducive learning environment for SMK Bekok students.

1. Introduction

The Textbook Loan Scheme Program in Malaysia, known as Skim Pinjaman Buku Teks (SPBT), was initiated in 1975 only for government-assisted schools [1]. This program ensures that all students in government schools, including primary and secondary schools, have access to loaned textbooks throughout their school years. In Bekok, Segamat has a secondary school called SMK Bekok [2], and their SPBT process is managed by the coordinator teacher, Miss Pang Chin Yin. Through this process, students can easily borrow textbooks from the school, facilitating access to essential learning resources without outright purchases [3].

However, the current process at SMK Bekok relies on manual methods such as handwritten notes to record student information related to who had borrowed textbooks. Coordinator teachers and class teachers handle the tracking, distribution, and regular updating of records for borrowed textbooks. The use of physical forms and paper-based documentation in the current process leads to inaccurate record-keeping and difficulties in monitoring textbook availability.

The problems arising from the manual system include challenges in maintaining accurate records, potential loss of handwritten documents, and difficulties in tracking the availability of textbooks. Coordinator teachers must cross-check handwritten records and communicate with students and class teachers to clarify lending and returning books. Moreover, insufficient textbooks from the previous batch can impact the learning progress of the next set of students, creating a burden on resource managers.

To address these problems, the proposed solution is to implement a proper modern Textbook Borrowing System to improve the effectiveness of textbook management at SMK Bekok. The system will prepare the database to ensure data accuracy, non-duplication, and easy tracking of specific textbooks. This does not lead to confusion, and textbooks can be distributed to students on time within a specified time. This can streamline the process and

provide textbooks with real-time tracking capabilities, enabling SMK Bekok students to create a more conducive learning environment. The project aims to provide clear direction for the system:

i. Design a user-friendly textbook borrowing system for SMK Bekok.
ii. Develop a web-based textbook borrowing system.
iii. Test the developed system for functionality and usability.

The study domain of the proposed textbook borrowing system mainly focuses on managing textbooks and retrieval processes within the secondary school named SMK Bekok. The system users include the coordinator teacher, class teachers, and students responsible for textbook borrowing. The system comprises seven modules which are login, student management, textbook catalog, record management, borrowing, return, and reporting.

2. Related Work

2.1 Manual Textbook Borrowing System

The manual textbook borrowing system at SMK Bekok includes six classes, which are Form 1 to Form 5 and a Transition class known as Peralihan in Malay. Each class is subdivided into three classes, except Peralihan, which only consists of one class. The government has established protocols to guarantee that every student can access textbooks through a systematic borrowing and returning process that aligns with the school year.

In this system, students are required to provide their signatures during the borrowing and returning processes. These signatures prove their responsibility for the textbooks, facilitating accurate record-keeping and transaction verification. The distribution of textbooks begins at the start of the school year, ensuring that students have the necessary resources. Textbooks are returned at the end of the academic year, and all returned books are carried over to the next batch of students, maintaining a continuous borrowing cycle.

The distribution and management of textbooks involves the coordinator teacher and class teachers. The coordinator teacher oversees the entire process, managing the system and ensuring the appropriate distribution of textbooks among classes. Class teachers are responsible for managing student-specific textbook records and handling transactions related to borrowing and returns. This collaborative effort among students and teachers forms the foundation of the manual textbook borrowing system.

This manual process approaches the variable that includes the number of borrowed textbooks, the condition of the textbook, whether it is damaged or lost, and the need for a signature by the student to receive or return the textbook. Manual processes are vulnerable to challenges such as the risk of losing records and the difficulty of tracking the availability of textbooks. The manual workflow for borrowing and returning textbooks in SMK Bekok is in Appendix A.

2.2 Web-Based Textbook Borrowing System

The chosen technique for the textbook borrowing system in SMK Bekok is a web-based approach, leveraging the power of the internet to enhance accessibility and streamline processes. This decision was made to overcome the limitations of traditional manual methods, bringing efficiency and ease of use to the management of textbook borrowing. In a web-based system, users, including coordinator teachers and class teachers, can access the system from any device with internet connectivity. This approach enables flexibility, allowing teachers to manage textbook-related tasks from different locations within the school premises or remotely. As SMK Bekok is a school environment, a stable internet connection is available, facilitating the seamless operation of the web-based system.

2.3 Database

The textbook borrowing system will employ MySQL as its database, and this database will be facilitated by phpMyAdmin, an open-source tool specifically designed for MySQL database management. This combination ensures a strong and reliable foundation for handling the system's data. MySQL will efficiently store and retrieve textbooks, student, and transaction data. Its structured data organization capabilities enable seamless interactions between various system components.

PhpMyAdmin is a user-friendly interface for MySQL administration that provides an accessible platform for managing databases. It allows for tasks such as creating, modifying, and deleting databases, tables, rows, or columns to be carried out effortlessly. Moreover, the interface supports the execution of SQL statements, a critical functionality for effective database management and querying.

2.4 Input Validation

In a web-based textbook borrowing system for SMK Bekok, input validation is critical to ensuring data integrity and security. Input validation involves checking and verifying the data entered by users to ensure it meets specific

criteria before it is processed or stored. Its primary function is to prevent invalid data from entering the system, which can cause errors, data corruption, or security vulnerabilities.

When a class teacher logs into the system, they must enter their credentials, which typically include an email and password. Input validation mechanisms will check that the email and password fields are not empty, ensuring no login attempts can be made with blank fields. Additionally, the system will verify that the input matches the expected format for emails and passwords. This includes checking for a minimum length, including letters and numbers, excluding prohibited characters, and ensuring the email field contains an '@' symbol.

For enhanced security, the password input should be hashed before being transmitted to the server to prevent interception. Once submitted, the system compares the entered credentials against the stored, encrypted data in the database. If the credentials are correct, the class teacher can access their dashboard. This dashboard allows them to manage textbook records, including borrowing and returns, by providing a streamlined interface to enter, update, and track textbook transactions. This ensures a secure and efficient process, minimizing errors and enhancing the overall management of the textbook borrowing system.

2.5 Study of Existing Related Systems

The comparison between the existing systems (Rent textbooks through Google Forms., eSPBT, and Librarika) and the proposed system reveals key distinctions in features that significantly impact the efficiency and functionality of the textbook borrowing process.

SMK Tengku Ampuan Rahimah in Klang, Malaysia [1], employs Google Forms to manage efficiently and record textbook borrowing information [4]. This online system allows teachers to input textbook details, enhancing records' accuracy. The advantage of this approach lies in its user-friendly interface and accessibility. However, its functionalities may be limited compared to a comprehensive database management system.

The eSPBT was developed by the Ministry of Education to centrally manage the supply of textbooks for SPBT projects in Malaysia [5]. It facilitates data collection and the ordering of new teaching materials. Available to registered schools, it focuses on the larger scale of the SPBT project and lacks the specific features of daily textbook lending for individual schools.

Librarika is an online library management system [6] with modules for acquisition, circulation, cataloging, report generation, and user management [7]. Operating on a web-based platform, it provides efficient book collection organization and supporting features like cataloging, book loan tracking, and member management. Librarika offers a user-friendly interface, supports barcode scanning, and allows the creation of a public catalog for users to explore available books.

Table 1 shows the comparison of the features between the existing system and the proposed system. The features listed are the online system, database server, login modules, etc.

Table 1 *Comparison Between Existing System and Proposed System*

Features/System	System A	System B	System C	Proposed System
Online System	Yes	Yes	Yes	Yes
Database Server	No	Yes	Yes	Yes
Login Modules	No	Yes	Yes	Yes
Manage Students Modules	Yes	Yes	No	Yes
Textbook Management Modules	No	No	Yes	Yes
Textbook Record Modules	Yes	Yes	Yes	Yes
Borrowing and Return Modules	Yes	Yes	Yes	Yes
Report Modules	No	No	Yes	Yes

System A: Rent textbooks through Google Form

System B: eSPBT (System Pengurusan Buku Teks)

System C: Librarika (Library Management System)

Proposed System: Textbook Borrowing System for SMK Bekok

3. Methodology

A software development methodology is the framework for planning, managing, and controlling processes [8]. The software development methodology is known as the software development life cycle (SDLC) and is used in many fields. The SDLC has various structure approaches, including waterfall development-based (sequential), Agile development-based (iterative and flexible), and others, each with its own principles and practices. The

software development methodology has been chosen in this chapter, and each phase will be explained and discussed.

3.1 Waterfall Model

The waterfall model is a software development life cycle model that was originally defined by Royce in the 1970s [9]. In the waterfall method, software development follows a series of logical phases, with progress flowing sequentially from one phase to the next. The fundamental premise was that requirements must be established upfront to facilitate software design, construction, and testing [10]. The Waterfall model is a linear and sequential approach that ensures each phase must be completed before the next phase begins. The progression resembles a waterfall, flowing steadily downward through the phases.

The Waterfall Model was chosen for the Textbook Borrowing System for SMK Bekok because it is structured and systematic. The requirements for the system are well-defined and unlikely to change significantly; the Waterfall Model provides a clear path from gathering requirements to deployment, making it easier to plan and manage the development process. The Waterfall model is divided into five phases which are the planning phase, analysis phase, design phase, implementation phase, and testing phase. Fig. 1 shows the Waterfall model diagram. The next section will discuss the activities conducted in each phase of the Waterfall model.

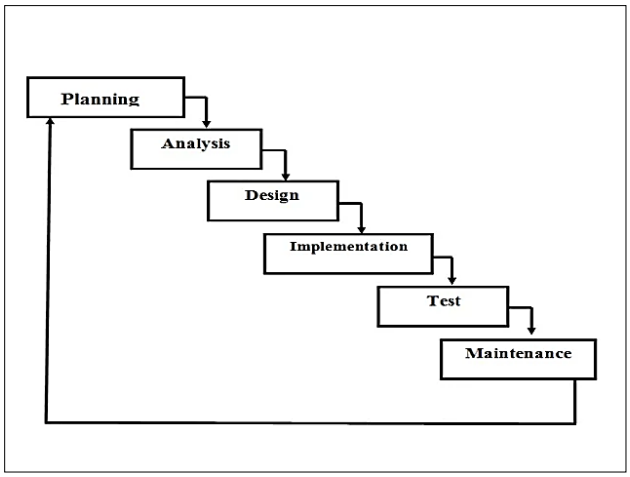

Fig. 1 *The Waterfall Process Model*

3.2 System Development Workflow

There are a total of six phases in the waterfall model. As shown in Table 2, each phase has its own assignment and output that needs to be produced during the entire project development. Besides that, the output had been completed within the specific days that have been given.

Table 2 *Software development activities and their task*

Phase	Task	Output
Planning	1. Propose the project and define the scope. 2. Determine the project schedule, activities, and output. 3. Conduct a Face-to-face interview section with the coordinator teacher.	1. Project proposal. 2. Develop a Gantt chart. 3. Collect and record all user requirements.
Analysis	1. Analysis and discussion of problem research. 2. Identified the hardware and software requirements. 3. Distinguish between functional and non-functional requirements.	1. Comparison between existing web-based system and proposed system. 2. Requirements for hardware and software. 3. Requirement both functional and non-functional.
Design	1. Design the system's flowchart, CD, DFD, and ERD. 2. Design the database scheme and data dictionary. 3. Design the wireframe of the system.	1. Flowchart, CD, DFD, and ERD production. 2. Database. 3. Wireframe production.

Table 2 *Software development activities and their task (continue)*

Phase	Task	Output
Implementation	1. Build and complete the module. 2. Establish a connection to the database.	1. Proposed system. 2. Error found and fixed
Testing	1. Conduct system testing. 2. Verification and validation of the functions of the system.	1. Fix and improve the errors or bugs. 2. Ensure that the system can run properly.
Maintenance	1. Addressing issues during actual use.	1. Solve the system errors.

4. Result and Discussion

The results and discussion section presents data and analysis. It includes system requirement analysis, flowchart, CD, DFD, ERD, and implementation and testing.

4.1 System Requirements Analysis

System requirement analysis is a crucial step in developing any system or software. It aims to understand users' requirements and make software specifications. It involves identifying, documenting, and validating the needs and expectations of users to ensure that the system meets their requirements. Therefore, the functional and non-functional requirements of the proposed system are specified and will be discussed in this section. Table 3 and Table 4 show the functional and non-functional requirements of the proposed system.

Table 3 *Functional requirements of the proposed system*

No	Module	Functionality
1.	Login Module	• Users (class teachers) must enter a valid email and password. • Coordinator Teachers must set and manage email and passwords for class teachers.
2.	Manage Students Module	• Class Teachers can add, edit, and delete student information. • Student details include IC, name, class, and subclass.
3.	Textbook Management Module	• Coordinator Teachers manage the catalog of textbooks. • Check the availability of textbooks in real-time. • Check the quantity and status of each textbook.
4.	Textbook Record Module	• Coordinator Teachers can insert and modify textbook details according to each class. • Coordinator Teachers can delete outdated or incorrect textbook records.
5.	Textbook Borrowing Module	• Class Teachers must be able to check the availability of textbooks. • Teachers should check the status of textbooks after the borrowing process. • Class Teachers should be able to edit and delete the borrowed textbook records for students.
6.	Textbook Return Module	• Teachers should check the status of textbooks after the return process. • Class Teachers should be able to edit and delete the returning textbook records for students.
7.	Report Module	• Coordinator teachers generate reports for each class. • Reports include the total number of textbooks available and the number of borrowed and returned textbooks.

Table 4 *Non-functional requirements of the proposed system*

No	Module	Description
1.	Usability	• The system should have an intuitive and user-friendly interface to facilitate ease of use.
2.	Performance	• The system should respond to user requests with minimum loading time.
3.	Security	• The system should have role-based access control to ensure data privacy.
4.	Reliability	• The system should have at least 99% uptime after launch.

Table 4 *Non-functional requirements of the proposed system (continue)*

No	Module	Description
5.	Operational	• The system should accommodate an increase in the number of students, teachers, and textbooks without significantly losing performance.

4.2 System Design

System design includes Context Diagram (CD), Data Flow Diagram (DFD), and Flowchart. A CD is a visual representation that provides an overview of a system and its interactions with external entities. Fig. 2 illustrates the CD of the proposed system. While DFD shows the flow of data within that proposed system and between its external entities and shows the inputs, outputs, processes, and data storage. Fig. 3 depicts the DFD level 0 of the proposed system. The Flowchart is a system process diagram illustrating the steps or activities involved in a system or process. Fig. 4 shows the flowchart of the proposed system for the coordinator teacher and class teacher.

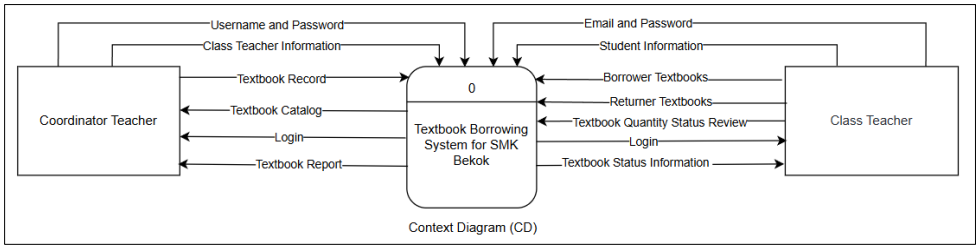

Fig. 2 *CD of the proposed system*

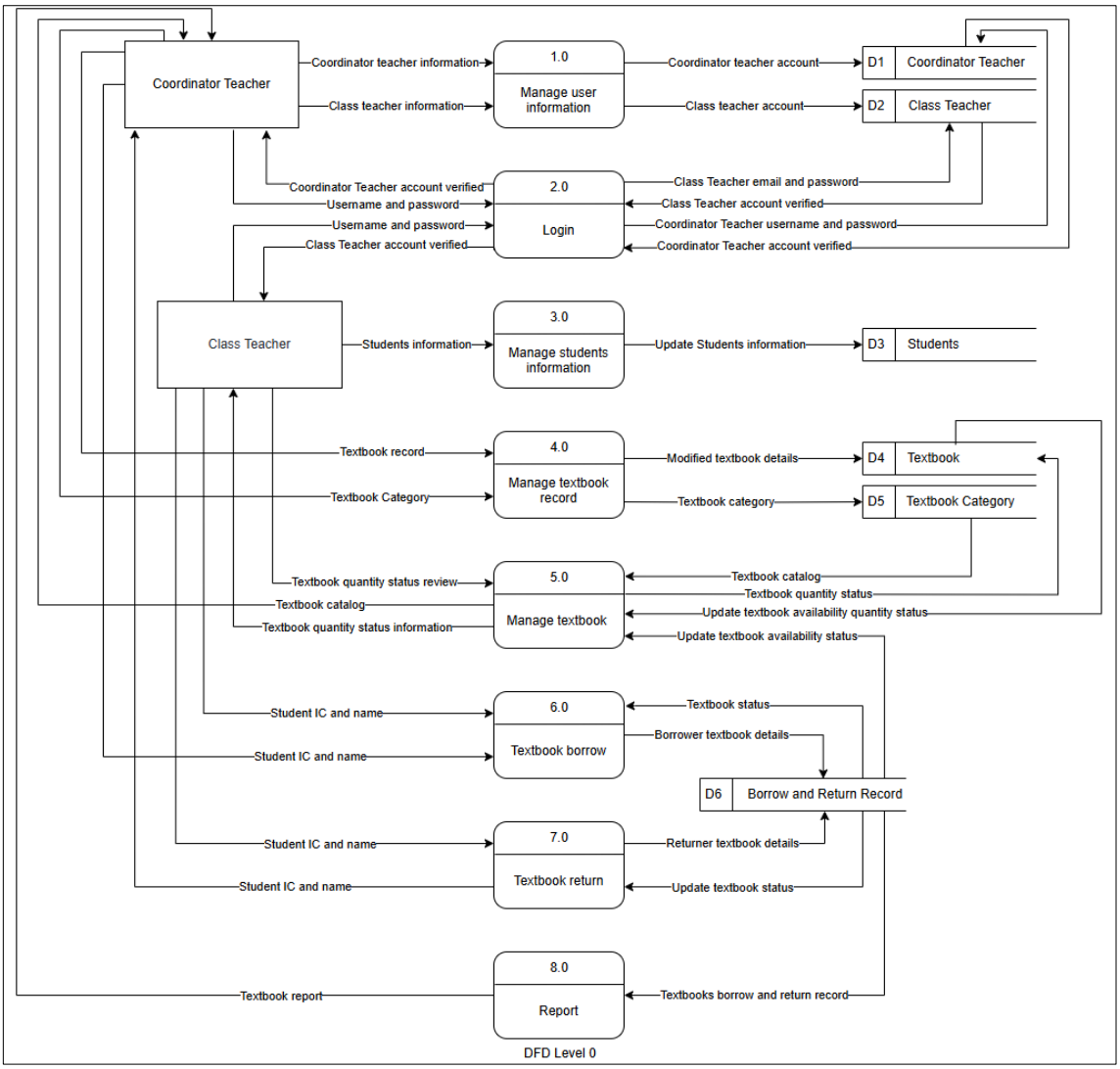

Fig.3 *DFD of the proposed system*

Applied Information Technology and Computer Science Vol. 5 No. 2 (YEAR) p. 371 - 387

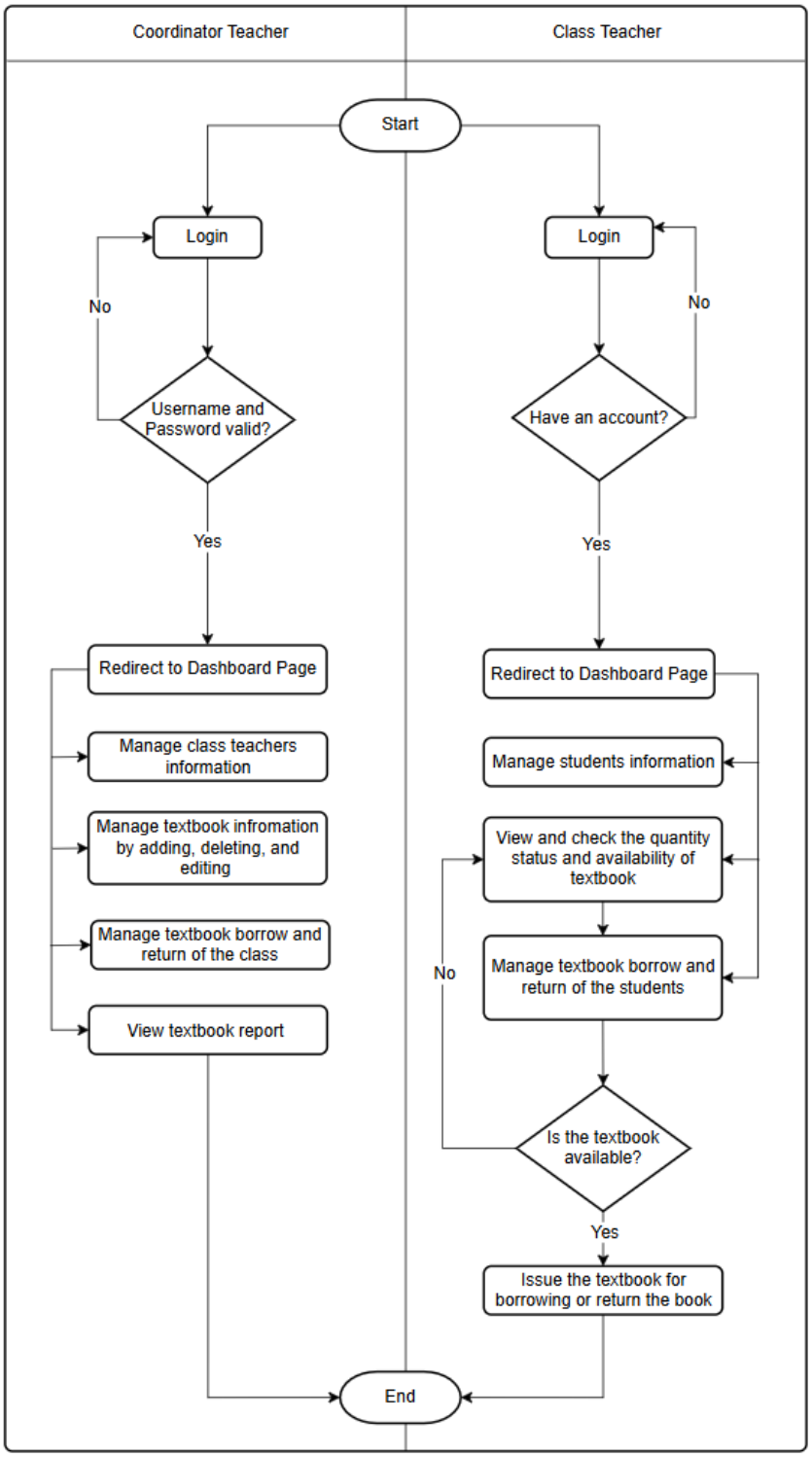

Fig. 4 *System flowchart of the proposed system*

4.3 Database Design

Database design defines the structure to facilitate and store data in a database system. It involves defining the database schema, which includes specifying the tables, columns, data types, relationships, and constraints that will be used to store and retrieve data efficiently. The database design components in this section include an Entity Relationship Diagram (ERD). Fig. 5 depicts the ERD of the proposed system.

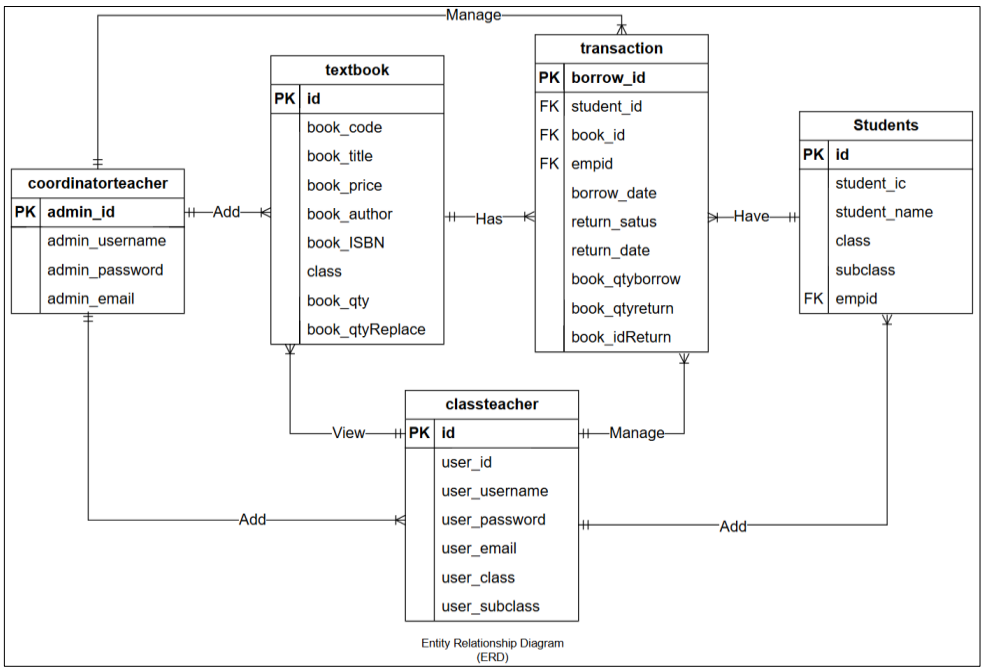

Fig. 5 *ERD of the proposed system*

4.4 Implementation and Testing

The implementation phase of a web system involves translating design specifications into actual code, ensuring that the system works as intended and meets established standards. This phase includes coding, database setup, and integration of various components. On the other hand, tests verify the web system's functionality, performance, and security. It includes rigorous checks to identify and correct errors or inconsistencies, ensuring the system operates reliably and meets user expectations.

4.4.1 Implementation

Textbook Borrowing System for SMK Bekok is a web-based platform that efficiently manages and tracks textbook loans. Data will be stored in a MySQL database, managed via SQL commands in the phpMyAdmin interface. A PHP file named `dbconn.php` will connect to the database using the MySQLi procedural approach shown in Fig. 6 The application will be developed in Visual Studio Code, using PHP, JavaScript, HTML, and CSS to create a robust, user-friendly interface.

```php
<?php
    define('DB_HOST','localhost');
    define('DB_USER','root');
    define('DB_PASS','');
    define('DB_NAME','textbookborrowsystem');

    // Create connection
    $dbh = mysqli_connect(DB_HOST, DB_USER, DB_PASS, DB_NAME);

    // Check connection
    if (!$dbh) {
        die("Connection Error: " . mysqli_connect_error());
    }
```

Fig. 6 *Database configuration of the proposed system*

Fig. 7 shows the coordinator teacher login interface, while Fig. 8 shows the code segment for the coordinator teacher login process. The coordinator teacher must enter the username and password before logging into the system. When the admin logs in successfully, the web page will go directly to the dashboard page, while if the login is unsuccessful, the alert will be prompted with 'Invalid Details.' Fig. 9 and Fig. 10 show the class teacher login interface and code segment for the class teacher login process. The class teacher must enter a valid email and password before logging into the system. When the class teacher logs in successfully, the web page will go directly to the dashboard page, while if the login is unsuccessful, the alert will be prompted: 'Sorry, Invalid Details.'

Applied Information Technology and Computer Science Vol. 5 No. 2 (YEAR) p. 371 - 387

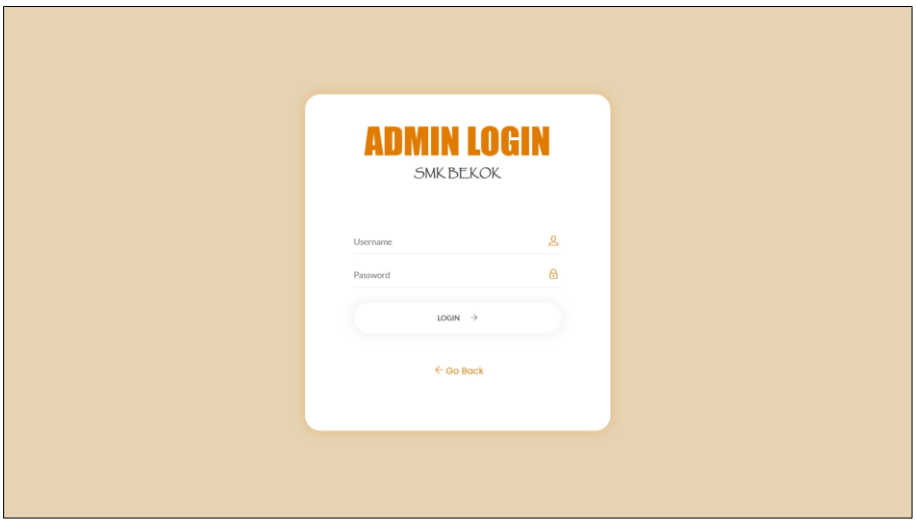

Fig. 7 *Coordinator teacher login interface*

```php
if (isset($_POST['signin'])) {
    $uname = $_POST['username'];
    $password = $_POST['password'];

    $sql = "SELECT admin_username, admin_password FROM coordinatorteacher WHERE admin_username = ? AND admin_password = ?";
    $query = mysqli_prepare($dbh, $sql);
    if ($query) {
        mysqli_stmt_bind_param($query, "ss", $uname, $password);
        mysqli_stmt_execute($query);
        mysqli_stmt_store_result($query);
        if (mysqli_stmt_num_rows($query) > 0) {
            $_SESSION['alogin'] = $uname;
            echo "<script type='text/javascript'> document.location = 'dashboard.php'; </script>";
        } else {
            echo "<script>alert('Invalid Details');</script>";
        }
        mysqli_stmt_close($query);
    } else {
        echo "Error in statement preparation: " . mysqli_error($dbh);
    }
    mysqli_close($dbh);
}
```

Fig. 8 *Code segment for the coordinator teacher login process*

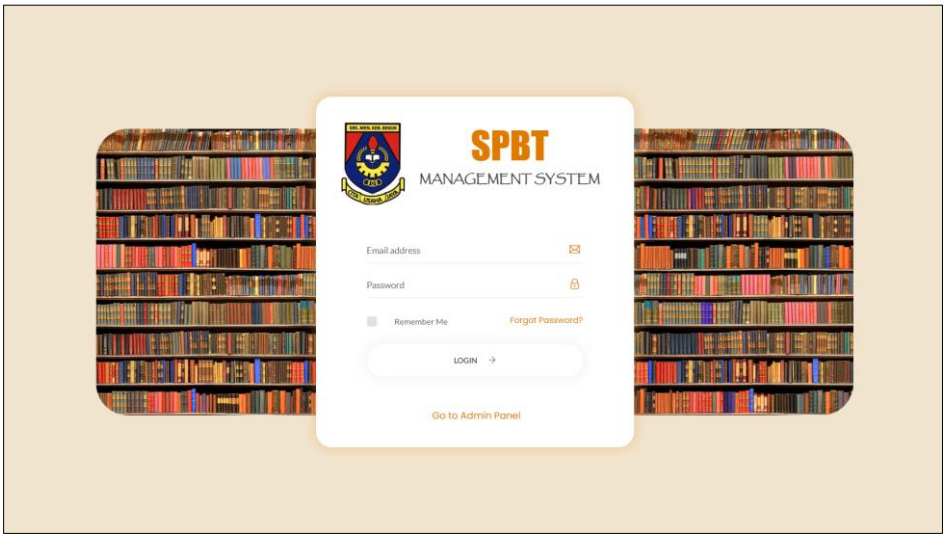

Fig. 9 *Class teacher login interface*

```php
if (isset($_POST['signin'])) {
    $uname = $_POST['username'];
    $password = $_POST['password'];

    $sql = "SELECT user_email, user_password, Status, id
            FROM classteacher WHERE user_email=?";
    $query = mysqli_prepare($dbh, $sql);
    mysqli_stmt_bind_param($query, "s", $uname);
    mysqli_stmt_execute($query);
    $result = mysqli_stmt_get_result($query);
    if (mysqli_num_rows($result) > 0) {
        while ($row = mysqli_fetch_assoc($result)) {
            $hashed_password = $row['user_password'];
            $status = $row['Status'];
            $empid = $row['id'];
            if (password_verify($password, $hashed_password)) {
                if ($status == 0) {
                    $msg = "In-Active Account. Please contact your administrator!";
                } else {
                    $_SESSION['emplogin'] = $uname;
                    $_SESSION['eid'] = $empid;
                    echo "<script type='text/javascript'> document.location = 'user/user-dashboard.php'; </script>";
                }
            } else {
                $error = "Sorry, Invalid Details.";
            }
        }
    } else {
        $error = "Sorry, Invalid Details.";
    }
}
```

Fig. 10 *Code segment for the class teacher login process*

Fig. 11 and Fig. 12 are the manage user (class teacher) interface and code segment of the manage user process. The coordinator teacher needs to add the class teacher's details, including the username, email, and class, and set up a specific password for each class teacher. Furthermore, the coordinator teacher must also manage the textbook information by class. Fig. 13 shows the manage textbook interface, while Fig. 14 shows the code segment of the manage textbook process.

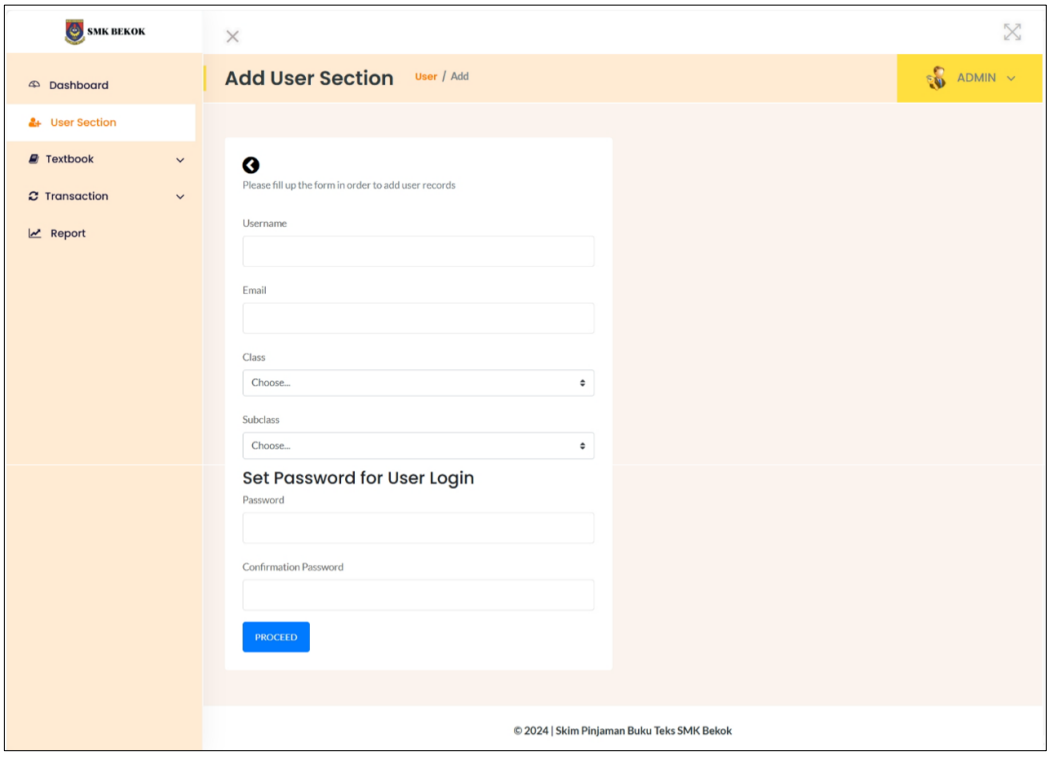

Fig. 11 *Manage user (class teacher) interface*

Fig. 12 shows the code segment for the add user page. The class teacher's data will be added and saved in the 'classteacher' table in the database. The Teacher ID is created randomly between 0 and 9999 using the 'mt_rand(0, 9999)' function. The ID will always have at least 4 digits, with the letter 'T' added before the number, resulting in a Teacher ID like 'T2546'.

```php
if (isset($_POST['add'])) {
    $userid = 'T' . str_pad(mt_rand(0, 9999), 4, '0', STR_PAD_LEFT);
    $username = $_POST['user_username'];
    $password = password_hash($_POST['password'], PASSWORD_BCRYPT);
    $email = $_POST['email'];
    $class = $_POST['user_class'];
    $subclass = $_POST['user_subclass'];
    $status = 1;

    $sql = "INSERT INTO classteacher(user_id, user_username, user_password, user_email, user_class, user_subclass, Status) VALUES(?, ?, ?, ?, ?, ?, ?)";
    $stmt = mysqli_prepare($dbh, $sql);

    if ($stmt) {
        mysqli_stmt_bind_param($stmt, 'sssssss', $userid, $username, $password, $email, $class, $subclass, $status);
        $result = mysqli_stmt_execute($stmt);

        if ($result) {
            $msg = "Class teacher has been added successfully";
        } else {
            $error = "ERROR: " . mysqli_error($dbh);
        }

        mysqli_stmt_close($stmt);
    } else {
        $error = "ERROR: Unable to prepare statement.";
    }
}
```

Fig. 12 *Code segment of the manage user process*

For class teachers, Fig. 13 shows the managed student interface. Class teachers must add student information such as name and identity card (IC) for their class. After successfully adding the students, the table will display the students' information by their class.

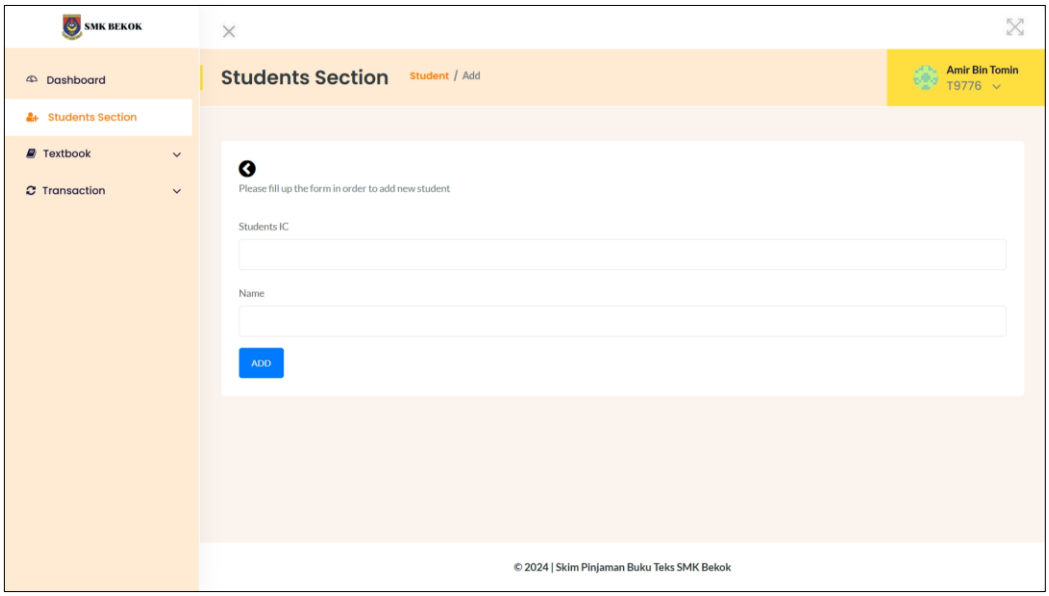

Fig. 13 *Manage student interface*

Fig. 14 shows the code segment of the manage student interface. The student data will be added and saved in the 'students' table database. The 'preg_match' function checks if the student's identification number ($ic) matches a specific pattern, ensuring it is 12 digits long and contains only numbers. This pattern prevents invalid entries. The SQL query then checks if the student's IC exists in the database. If it does, an error message indicates that a student with the same IC already exists.

```php
if (isset($_POST['add'])) {
    $empid = $_SESSION['eid'];
    $ic = $_POST['student_ic'];
    $username = $_POST['student_name'];

    if (!preg_match('/^\d{12}$/', $ic)) {
        $error = "Error adding Student: IC must be exactly 12 digits.";
    } else {
        $check_sql = "SELECT id FROM students WHERE student_ic = ?";
        $check_query = mysqli_prepare($dbh, $check_sql);
        mysqli_stmt_bind_param($check_query, "s", $ic);
        mysqli_stmt_execute($check_query);
        mysqli_stmt_store_result($check_query);
        $rows = mysqli_stmt_num_rows($check_query);
        mysqli_stmt_close($check_query);
        if ($rows > 0) {
            $error = "Error adding Student: IC already exists.";
```

Fig. 14 *Code segment of the manage student interface*

Fig. 15 shows the interface of borrowed textbooks textbooks, while Fig. 16 shows the code segment of the borrowed textbooks. The class teacher must enter the student's name and IC and select the textbooks the student needs to borrow. After successfully adding the students, the table will display the information on the borrowed textbook. Form Fig. 16 shows that when the student issues a textbook, the 'student_id' and 'book_id' are retrieved from the table transaction database. It then checks if there's an ongoing transaction (return_status is 0) for that student. If there is, it updates the existing record by adding the new books, ensuring no duplicates. If not, it creates a new transaction. The book quantity is updated in the database by decrements of the quantity borrowed for each book.

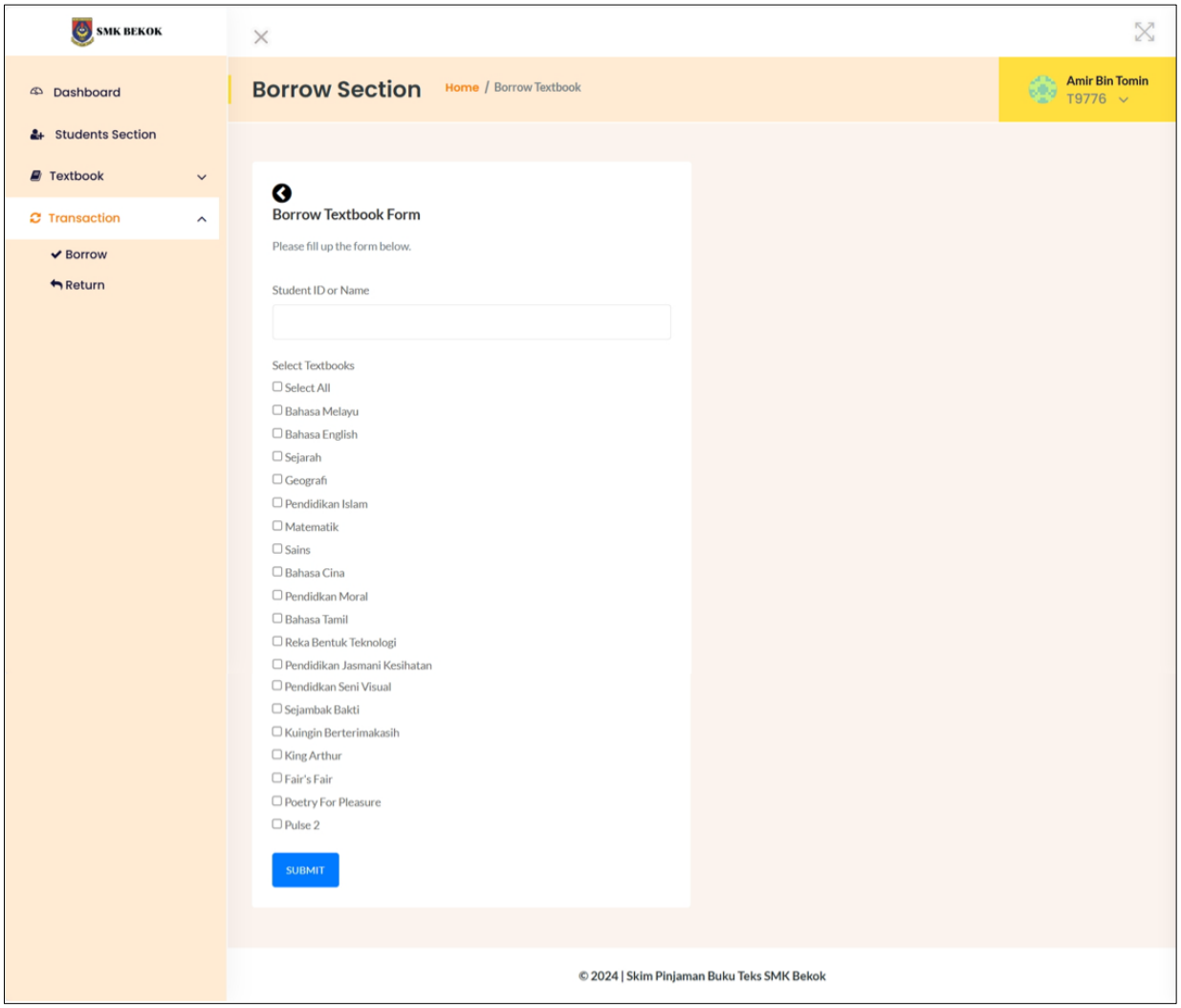

Fig. 15 *Borrow textbook interface*

```
} else {
    $insertSql = "INSERT INTO transaction (student_id, book_id, empid, borrow_date, return_status, return_date, book_qtyborrow)
                  VALUES (?, ?, ?, NOW(), 0, NULL, ?)";
    $insertQuery = mysqli_prepare($dbh, $insertSql);
    mysqli_stmt_bind_param($insertQuery, 'ssii', $studentic, $new_book_ids, $empid, $book_qtyborrow);
```

Fig. 16 *Code segment of borrow textbook process*

Fig. 17 shows the interface of returned textbooks, while Fig. 18 shows the code segment of the returned textbooks. Class teachers can distinguish between students who have not returned books based on their status. A "borrow" or "0" status indicates that the student has not returned all the books, while a "return" or "1" status indicates that the student has returned all the books. When a return is submitted, it retrieves the selected 'book_id' and 'borrow_id', then sets the return status to 1 if all the textbook is already returned; if not, the return status will still be 0. The textbook quantity is incremented for each returning textbook, and the transaction record is updated to reflect the return. If a book is a replacement, it follows the same process for replacement quantities.

Applied Information Technology and Computer Science Vol. 5 No. 2 (YEAR) p. 371 - 387

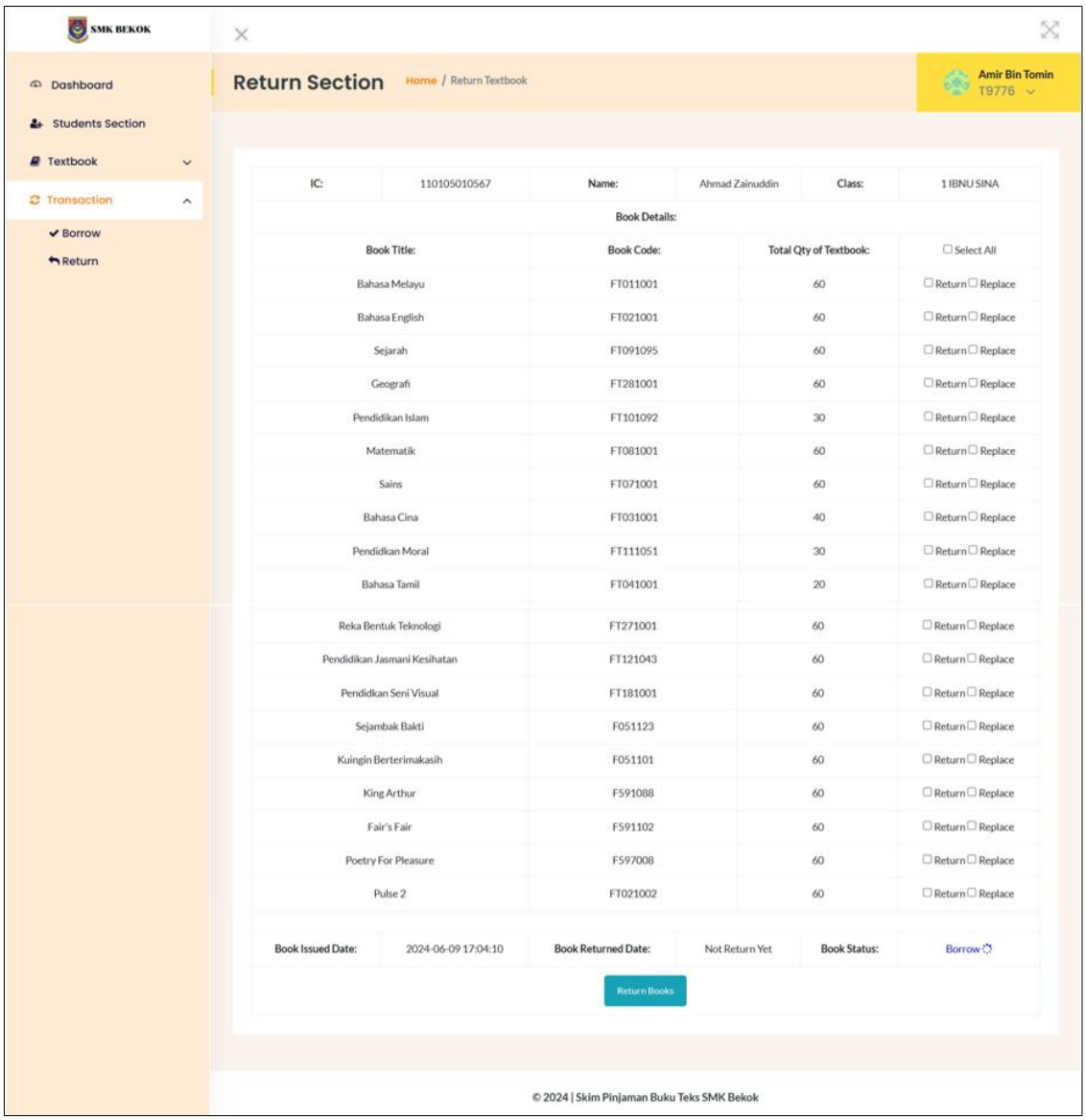

Fig. 17 *Return textbook interface*

```
$return_date = date('Y-m-d H:i:s');
$sql = "UPDATE transaction SET return_status = ?, return_date = ?, book_qtyreturn = ?, book_idReturn = ?
        WHERE borrow_id = ? AND FIND_IN_SET(?, book_id)";
$stmt = mysqli_prepare($dbh, $sql);
mysqli_stmt_bind_param($stmt, "isssii", $rstatus, $return_date, $book_qtyreturn, $book_idReturn, $rid, $book_id);
if (!mysqli_stmt_execute($stmt)) {
    throw new Exception("Failed to update transaction record");
}
```

Fig. 18 *Code segment of return textbook process*

4.4.2 Testing Plan

The purpose of a test plan for a system is to outline the strategy and procedures for verifying that the system meets its requirements, functions correctly, and is free of defects. It ensures thorough testing coverage, identifies test objectives, and provides a roadmap for executing and evaluating tests to guarantee the system's reliability and performance.

The functional testing result of the Textbook Borrowing System for SMK Bekok for the coordinator teacher site is shown in Table 5, while the class teacher site is shown in Table 6. The results of the non-functional testing are shown in Table 7.

Table 5 *Functional test of the proposed system (Coordinator Teacher)*

Section	Test Cases	Expected Result	Actual Pass
Login	Insert invalid data, either an invalid username or invalid password	Messages window prompt out: Invalid Details	Pass
	Submit the form without the data input	Messages prompt out: Please fill out these fields	Pass
	Insert the correct username and password	Directly go to the dashboard page	Pass
Dashboard	View all the data like a total textbook, class teacher, borrow, and return	Show all total data	Pass
Manage User (Class Teacher)	Create the user details by inserting the valid email and username	Messages info prompt out: Class teacher has been added successfully	Pass
	Create the user details by inserting the invalid email and username	Messages prompt out: Please fill out these fields Messages prompt out: Please include an '@' in the email address.	Pass
	View the user details	Show all the successfully added user details	Pass
	Edit all user details, including the username, class, and subclass	Messages info prompt out: Teacher record updated successfully	Pass
	Delete the user account	Messages window prompt out 'Do you want to delete' and successfully a messages info prompt out 'The selected teacher has been deleted'	Pass
Manage Textbook	Create the textbook details by inserting the book code, ISBN, title, author, price, quantity of textbook	Messages info prompt out: Textbook added successfully	Pass
	Create the textbook details by inserting the invalid ISBN	Messages prompt out: Please enter a 13-digit number	Pass
	View the textbook's details	Show all the successfully added textbook details	Pass
	Edit all textbook details	Messages info prompt out: Textbook updated successfully	Pass
	Delete the textbook	Messages window prompt out 'Do you want to delete' and successfully a messages info prompt out 'Textbook record deleted'	Pass
	Selected class category	Only show selected classes category textbook	Pass
Transaction (Manage Borrowing and Returning)	Selected class category	Only show selected class category students	Pass
	Issue the textbook for each specific student	Messages info prompt out: Book(s) issued successfully	Pass
	Return the textbook for each specific student	Messages info prompt out: All books returned successfully	Pass
	View the borrow and return details for each borrower student	View the all-textbook details borrow and return by the student	Pass
Report	Selected class category	Only show the selected class category textbook report	Pass

Table 6 *Functional test of the proposed system (Class Teacher)*

Section	Test Cases	Expected Result	Actual Pass
Login	Insert invalid data, either an invalid email or an invalid password	Messages prompt out: Sorry, Invalid Details	Pass
	Submit the form without the data input	Messages prompt out: Please fill out these fields	Pass
	Insert the correct email and password	Directly go to the dashboard page	Pass
Dashboard	View all the data like a total textbook, students, borrow, and return	Show all total data	Pass
Manage Students	Create the student's details by inserting the IC and name	Messages info prompt out: Student created successfully	Pass
	Enter the same IC repeatedly	Messages prompt out: Error adding Student: IC already exists	Pass
	View the student details	Show all the successfully added student details	Pass
	Edit all student details	Messages info prompt out: Student updated successfully	Pass
	Delete the user account	Messages window prompt out 'Do you want to delete' and successfully a messages info prompt out 'The selected student has been deleted'	Pass
View Textbook	View the textbook's details	Only show textbooks that belong to the teacher's class	Pass
Transaction (Manage Borrowing and Returning)	Issue the textbook for each specific student by teacher's class	Messages info prompt out: Book(s) issued successfully	Pass
	Issue the twice textbook for a student	Messages info prompt out: Cannot issue the same book(s) multiple times.	Pass
	Return the textbook for each specific student	Messages info prompt out: All books returned successfully	Pass
	View the borrow and return details for each borrower student	View the all-textbook details borrow and return by the student	Pass

Table 7 *Non-functional requirements of the proposed system*

Section	Expected Result	Actual Pass
Usability	The system should have an intuitive and user-friendly interface to facilitate ease of use.	Pass
Performance	The system should respond to user requests and have minimum loading time.	Pass
Reliability	The system should have at least 99% uptime after launch.	Pass
Operational	The system should work on any web browser with an internet connection.	Pass
Security	The system should have role-based access control to ensure data privacy.	Pass

5. Conclusion

In conclusion, the "Textbook Borrowing System for SMK Bekok" project has successfully developed a web-based system that addresses the inefficiencies of the manual textbook borrowing process. The system, created using PHP, JavaScript, CSS, phpMyAdmin, and MySQL, offers an organized and efficient solution, enabling coordinators to manage textbook inventories, oversee borrowing and returning processes, and generate detailed reports. This project has achieved its primary objectives and significantly improved the efficiency of textbook management at SMK Bekok. The system improves efficiency by simplifying the process of checking textbook availability and managing inventories, allowing coordinators to manage class-specific textbooks, automating the tracking of borrowing and returning textbooks, enabling class teachers to manage student accounts and specific inventories, and automatically updating inventory counts when textbooks are borrowed or returned.

Acknowledgment

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

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

The authors confirm contribution to the paper as follows: ***study conception and design:*** *LIAW WAN QING, Hairulnizam Bin Mahdin;* ***data collection:*** *Liaw Wan Qing;* ***analysis and interpretation of results:*** *LIAW WAN QING, Hairulnizam Bin Mahdin;* ***draft manuscript preparation:*** *LIAW WAN QING, Hairulnizam Bin Mahdin. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] N. S. N. Ismail, M. A. Bin Zakaria, N. A. B. Muhammad, and F. B. Yunus, "Smart Skim Pinjaman Buku Teks (SBPT) School Operation System in Malaysia," in *2021 IEEE International Conference on Automatic Control and Intelligent Systems, I2CACIS 2021 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Jun. 2021, pp. 259–264. doi: 10.1109/I2CACIS52118.2021.9495924.

[2] N. Sulaiman *et al.*, "Rekod baru spesies rama-rama penyebab alahan di Malaysia (Toxoproctis hemibathes) dan kajian kesannya terhadap penduduk di Mukim Labis dan Mukim Bekok, Johor," 2012.

[3] Suhaili Salleh, "Pengurusan fail SPBT.," Slide share. Accessed: Jun. 10, 2024. [Online]. Available: https://www.slideshare.net/slideshow/pengurusan-fail-spbt/47941123

[4] "SPBT SMK Tengku Ampuan Rahimah." Accessed: Jun. 10, 2024. [Online]. Available: https://spbtstarklang.wordpress.com/kemasukan-rekod-buku-teks-pelajar/

[5] "eSPBT System." Accessed: Jun. 10, 2024. [Online]. Available: https://espbt.moe.gov.my/

[6] M. Yuvaraj, "Library automation with cloud based ILMS Librarika: case study of Central University of South Bihar," *Library Hi Tech News*, vol. 33, no. 7, pp. 13–17, 2016, doi: 10.1108/LHTN-04-2016-0016.

[7] A. Gimba Library, "USABILITY STUDY OF LIBRARIKA LIBRARY MANAGEMENT SOFTWARE IN LIBRARY SERVICES OF A PRIVATE UNIVERSITY LIBRARY IN NIGERIA," *Lapai Journal of Science and Technology*, vol. 6, no. 1, 2020.

[8] H. K. Aroral, "Waterfall Process Operations in the Fast-paced World: Project Management Exploratory Analysis," *International Journal of Applied Business and Management Studies*, vol. 6, no. 1, p. 2021.

[9] L. Sherrell, "Waterfall Model," in *Encyclopedia of Sciences and Religions*, Dordrecht: Springer Netherlands, 2013, pp. 2343–2344. doi: 10.1007/978-1-4020-8265-8_200285.

[10] Clifford F. Gray and Erik W. Larson, *Project_Management_The_Managerial_Proces (1)*.

Applied Information Technology and Computer Science Vol. 5 No. 2 (YEAR) p. 371 - 387

Appendix A: The manual process flow of borrowing and returning textbooks