Applied Information Technology And Computer Science Vol. 4 No. 1 (2023) 486-504

AITCS

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

Development of Library Management Mobile Application using Object-Oriented Approach

Lau Sian En[1], **Noryusliza Abdullah**[1]*

[1]Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/aitcs.2023.04.01.028
Received 19 June 2022; Accepted 26 September 2022; Available online 30 June 2023

Abstract: The Library Management Mobile Application is proposed to replace the manual library management system at Sekolah Kebangsaan (Perempuan) Bandar Kluang, SKPBK. This application is developed to enable portability in daily library tasks among the students, teachers and administrator. Furthermore, this application also allows the NILAM Program to be performed digitally. Additionally, the implementation of the payment penalty module grants the users a cashless environment by using a payment gateway. The project is conducted based on the Iterative Software Development Methodology. Android Studio software and Firebase database are used to build the application. Generally, with the support of the Library Management Mobile Application, the flow of work in terms of library-related subjects will be more effective, and efficiency will progressively grow.

Keywords: Library Management, NILAM, Mobile Application

1. Introduction

Library management is the application of management principles and techniques to the context of a library. The word management advocates for an approach that requires one to continue improving services through management in line with the ever-changing needs of the customers [1]. The head of the library of Sekolah Kebangsaan (Perempuan) Bandar Kluang, SKPBK, Mr. Saifulnizam Saari plays a crucial role in managing and monitoring all the operations carried out in the library. The library of SKPBK will be using the Library Management Mobile Application for students, teachers and also administrator.

A library management programme allows them to effortlessly manage their local libraries while also contributing to the central database's information richness [2]. The development of a Library Management Mobile Application as the main objective and benefit is portability will help in solving the problems. Students, teachers and administrator will be allowed to use this mobile application anywhere and anytime. The proposed mobile application will be equipped with digital books, book borrowing and returning status, online payment of fines for due books, add/edit/delete student's data,

*Corresponding author: yusliza@uthm.edu.my

NILAM Reporting support and a better user interface and user experience. Correspondingly, the proposed mobile application is much more versatile and consists of more functions than the current system used in the library which could solve and overcome the problems being faced by the users.

The current manual system which is in a non-portable device like a PC does give some basic functionalities like book borrowing and returning and also book record. However, it does not have any sort of portability like a mobile application on handheld devices. Due to the long stretch period of MCO, students and teachers are not able to go to school to proceed with the NILAM Program, hence students and teachers need to perform the NILAM Program through online platforms. Through the application development of mobile devices, students, teachers and the administrator will be able to conduct and monitor the NILAM Program with ease.

Furthermore, the current system does keep records of those who are late on returning books. Nevertheless, the library only implemented cash payment for the penalty. Hence, an online payment penalty for late return of books will be one of the features of the mobile application. Students that do not bring any cash to school could not pay the fines for the late return of books. Therefore, parents or guardians could pay the penalty through a mobile application with a payment gateway that will be implemented into the mobile application.

To attain the goal of this project, the following objectives have been set:

(i) To design a library management mobile application using an object-oriented approach.
(ii) To develop a mobile application for the library of SKPBK.
(iii) To perform tests on the developed mobile application.

In addition, the Library Management Mobile Application will focus on the matter related to the library whether it is inside or outside of a library. The language of the mobile application is English. The case study location will be in the library of SKPBK. System user lists are students, teachers and administrator. System Function Modules: User Administration Module, Book Administration Module, NILAM Program Module, eBook Module and Payment Penalty Module.

2. Related Work

2.1 Library Management Mobile Application

The latest trend is that "information is at our fingertips." A user's mobile device can be turned into a digital library [3]. Library Management Mobile Application will provide a seamless user interface equipped with features like user administration so that teachers and administrator will have full control over the students' data in terms of creating, reading, updating, and deleting students' information.

Manual systems at libraries are failing to keep up with the recent surge in information requests, and borrowers' requests for book and study information are overwhelming [4]. To keep track of the book records and status, students and administrator could use the book administration feature in the mobile application to review and check on the books whether the books have been borrowed or lost. This feature will save a lot of energy and time on finding the books compared to the manual system that is currently used by SKPBK.

2.2 Study of the Existing Systems

Glibrary-Library Management System [5], AutoLib [6], and Libero [7] are the three existing programs that are similar to Library Management Mobile Application. These three applications are available for download from the Google Play Store and the applications' official website. On the basis of the specific features, a comparison is made between the existing application and the proposed application. The comparison between the existing application and the proposed application is shown in Table 1.

Table 1: Comparison between the Existing Application and the Proposed Application

Features	Glibrary	AutoLib	Libero	Proposed Application
Android Based	✓	✓	✓	✓
iOS Based	×	×	✓	×
Web-Based	✓	✓	✓	×
Simple Navigation	✓	✓	✓	✓
High-Performance Design	✓	×	✓	✓
Clear Hierarchy	✓	×	✓	✓
User Accessibility	✓	✓	✓	✓
Tailored User Experience	×	×	✓	✓
Member Management	✓	✓	✓	✓
Staff Management	✓	×	✓	✓
Book Issue/Return/Reissue Management	✓	✓	✓	✓
Book Administration Module	✓	✓	✓	✓
Reports and Records Module	✓	✓	✓	✓
Barcode Reader	✓	✓	✓	×
QR Code Reader	×	×	×	×
eBook Module	×	✓	×	✓
NILAM Program Module	×	×	×	✓
Payment Penalty Module (Payment Gateway)	×	×	×	✓

3. Methodology/Framework

The iterative model is an iterative strategy for developing a system that starts with a list of requirements and continually improves the changing versions until the full system is built. Each phase, referred to as an "iteration," comprises the entire development process and includes steps such as planning, analysis & design, implementation, and testing. It is not necessary to start with a comprehensive requirement because software enhancement begins with identifying and executing sections of the software. This technique is then repeated to build a new version of the software at the end of each model cycle [8].

3.1 Planning & Requirement Phase

The planning & requirement phase is the initial stage of the iterative model. Tasks such as designing the project plan, identifying the development framework, and constructing a project timetable are all part of the planning process for this project. A Gantt Chart is used to guarantee that the project runs smoothly and efficiently. This will help ensure that the tasks are completed on time. Apart from that,

the collection of requirements are gathered from the client. An interview session with the client, Mr Saifulnizam Saari, the head of the library of SKPBK is carried out to acquire the user requirements.

3.2 Analysis Phase

Following the completion of the planning & requirement phase, an analysis is carried out to determine the suitable business logic, database models, and user interfaces that will be required at this point of the project. The requirements collected from the client are being analysed and processed to form a design. This is the context for the system's function in the typical system attribute of specified requirements, surroundings and strategy. There are two types of requirements: functional requirements and non-functional requirements. The functional and non-functional requirements of the Library Management Mobile Application are included in Table 2 and Table 3 respectively.

Table 2: Functional Requirements Analysis

Num	Function Modules	Functionalities
1.	Login	This module grants users to log in with a valid username and password.
2.	Logout	This module enables users to log out from the application.
3.	Profile Edit	This module allows users to edit their profiles.
4.	Book Borrowing	This module allows students to borrow or return books from the library. Students and administrator can check their own and the other students' borrowing records respectively. Using this module, administrators can issue books to students.
5.	Book Records	This module grants the administrator to add/edit/delete the books and also check on the book records.
6.	NILAM Report	This module enables students to add or edit NILAM reports and teachers can review and evaluate the NILAM report done by students.
7.	eBook Databases	This module allows students to access eBook databases.
8.	Penalty Payment	This module grants students to pay the penalty payment using the payment gateway. Administrator can check the penalty on the student using this module.
9.	User Administration	This module allows administrator to manage users including students and teachers.

Table 3: Non-functional Requirements Analysis

Num	Requirements	Functionalities
1.	Performance	The application should be available 24 hours a day, seven days a week, with the exception of scheduled maintenance.

Table 3: Non-functional Requirements Analysis (cont)

Num	Requirements	Functionalities
2.	Security	The application can only be accessed if users enter the correct username and password.
3.	Operational	To support real-time database updates, the application will have an always-on data connection.

3.3 Design Phase

After the requirements are collected from the client, it is being analysed and processed to form a design. Unified Modelling Language (UML) provides a rich set of visual modelling concepts to describe the structural and behavioural aspects of software at different levels of abstraction [9].

3.3.1 Use-Case Diagram

Use-cases specify the desired behaviour rather than the method for accomplishing it. One of the most important concepts in use case modelling is that it helps us design a system from the perspective of the end-user. Figure 1 depicts the use-case diagram for students, Figure 2 depicts the use-case diagram for teacher, and Figure 3 depicts the use-case diagram for administrator.

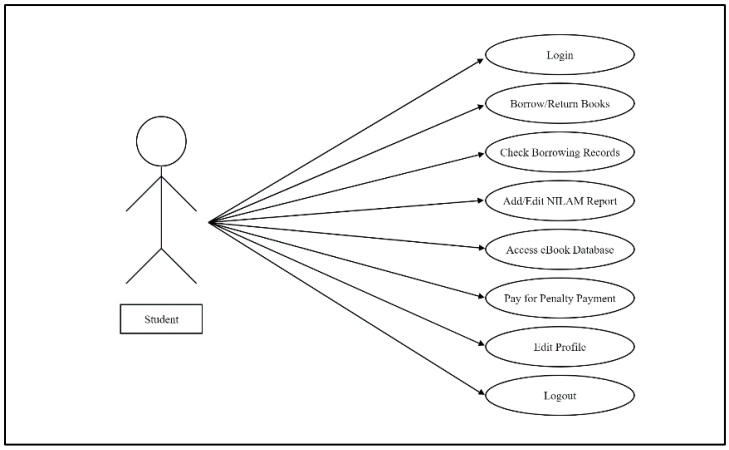


Figure 1: Use-Case Diagram for Student

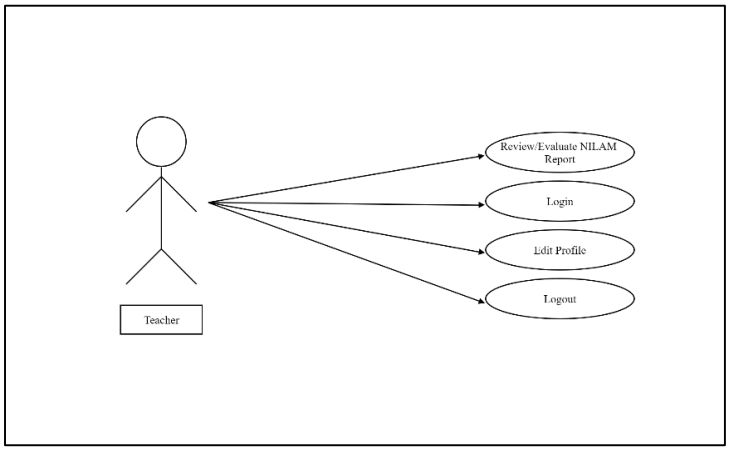


Figure 2: Use-Case Diagram for Teacher

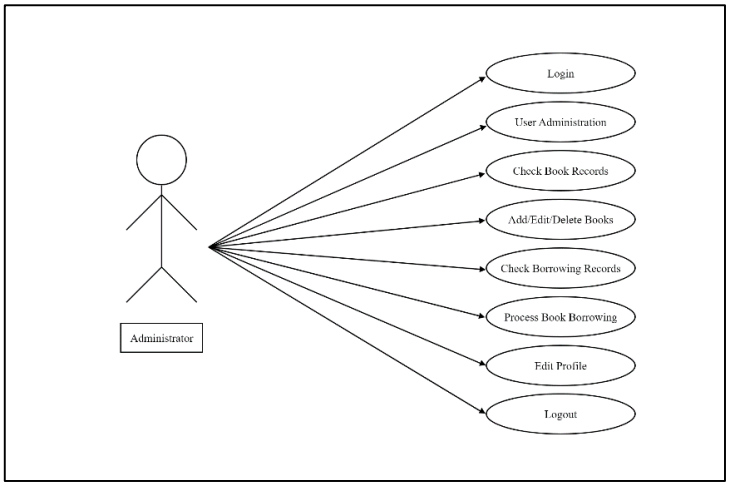


Figure 3: Use-Case Diagram for Administrator

3.3.2 Class Diagram

A class diagram is a form of a static structure diagram that depicts a system's structure by displaying the system's classes, properties, actions (or methods), and relationships between objects. The class diagram as shown in Figure 4 contains a total of nine classes which are student, teacher, administrator, eBook, penalty payment, book borrowing, NILAM, user administration and books.

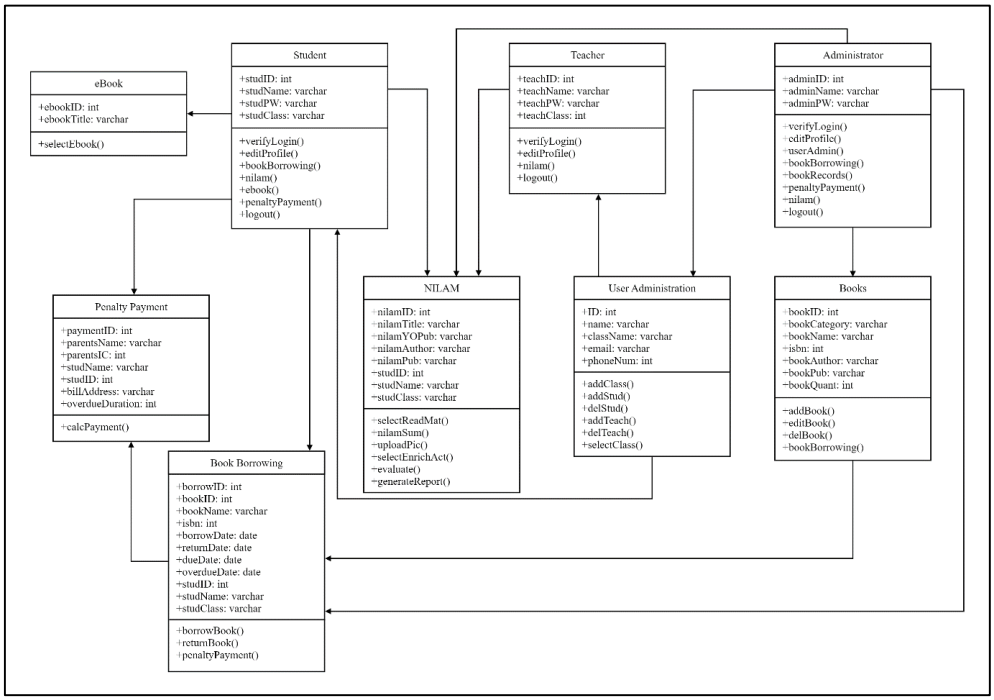


Figure 4: Class Diagram

3.3.3 Activity Diagram

Figure 5 illustrates the student, teacher and administrator activity diagram. The login page will be the starting point for the entire procedure. Users must enter a valid account ID and password. Students can choose whether they want to borrow/return books. Teachers can review/evaluate the NILAM report. Administrators can manage users in the user administration module.

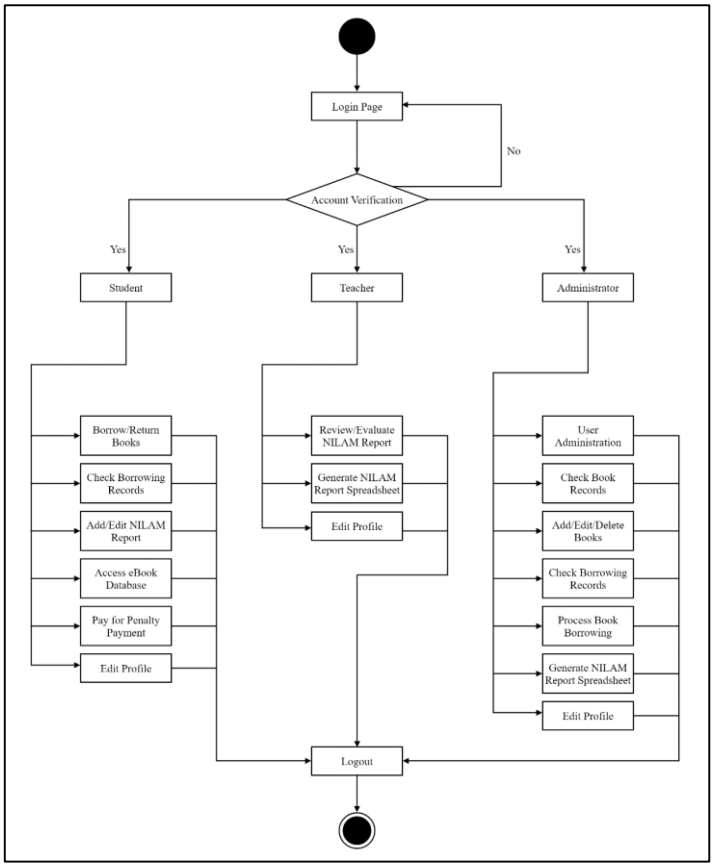


Figure 5: Activity Diagram

3.3.4 User Interface Design

The method of designing user interfaces in software or digital devices with a focus on appearance or style is known as user interface design. Designers strive to build user-friendly and engaging interfaces. Figure 6 illustrates the wireframes for the mobile application.

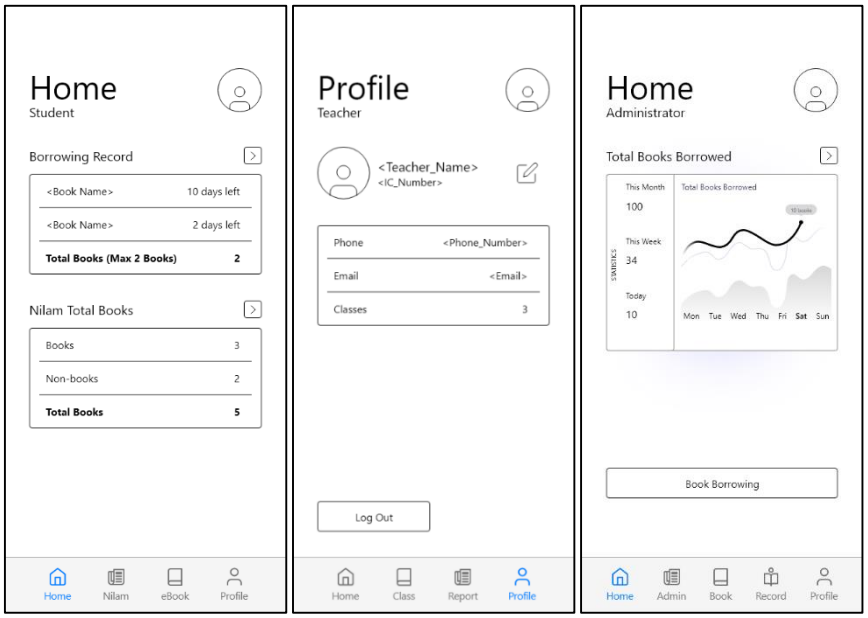


Figure 6: User's Wireframe

3.4 Implementation Phase

The real implementation and coding process begins when the analysis and design have been completed. Up to this point, all planning, specification, and design documents have been coded and incorporated into the project's first iteration. The Library Management Mobile Application is built with the Android Studio IDE and the Dart programming language (Flutter Framework). Each iteration's implementation phase requires one to two months to produce a succession of versions that are produced progressively.

3.5 Testing Phase

After this iteration of the build has been created and implemented, the next step is to run through a set of testing methods to discover and locate any potential flaws or issues that have arisen during the testing phase. To mitigate the risk of project uncertainty, a test strategy is offered for mobile application developers, as well as user acceptance testing for students, teachers and administrators. Following the completion of the testing phase, the subsequent iteration of implementation is carried out, utilising the tested results for improvements and alterations. The test plan shown in Table 4 is designed according to the nine function modules.

Table 4: Test Plan

Function Modules	Expected Result	Actual Result	Feedback
Login	1. Users are able to log in with a valid username and password.	Pass/Fail	
	2. The login button is functioning.	Pass/Fail	
Logout	1. This module enables users to log out from the application.	Pass/Fail	
	2. The logout button is functioning.	Pass/Fail	
Profile Edit	1. Users are able to edit their profiles.	Pass/Fail	
	2. Users are able to update their passwords.	Pass/Fail	
Book Borrowing	1. Students are able to borrow or return books from the library.	Pass/Fail	
	2. Students are able to check their borrowing records.	Pass/Fail	
	3. Administrator is able to check students' borrowing records.	Pass/Fail	
Book Records	1. Administrator is able to add/edit/delete the books and also check on the book records.	Pass/Fail	

Table 4: Test Plan (cont)

Function Modules	Expected Result	Actual Result	Feedback
NILAM Report	1. Students are able to add or edit NILAM reports.	Pass/Fail	
	2. Teachers are able to review and evaluate the NILAM report done by students.	Pass/Fail	
eBook Databases	1. Students are able to access eBook databases.	Pass/Fail	
Penalty Payment	1. Students are able to pay the penalty payment using the payment gateway.	Pass/Fail	
	2. Administrator can check the penalty on the student using this module.	Pass/Fail	
User Administration	1. Administrator is able to view, add, edit or delete students and teachers.	Pass/Fail	

4. Implementation and Testing

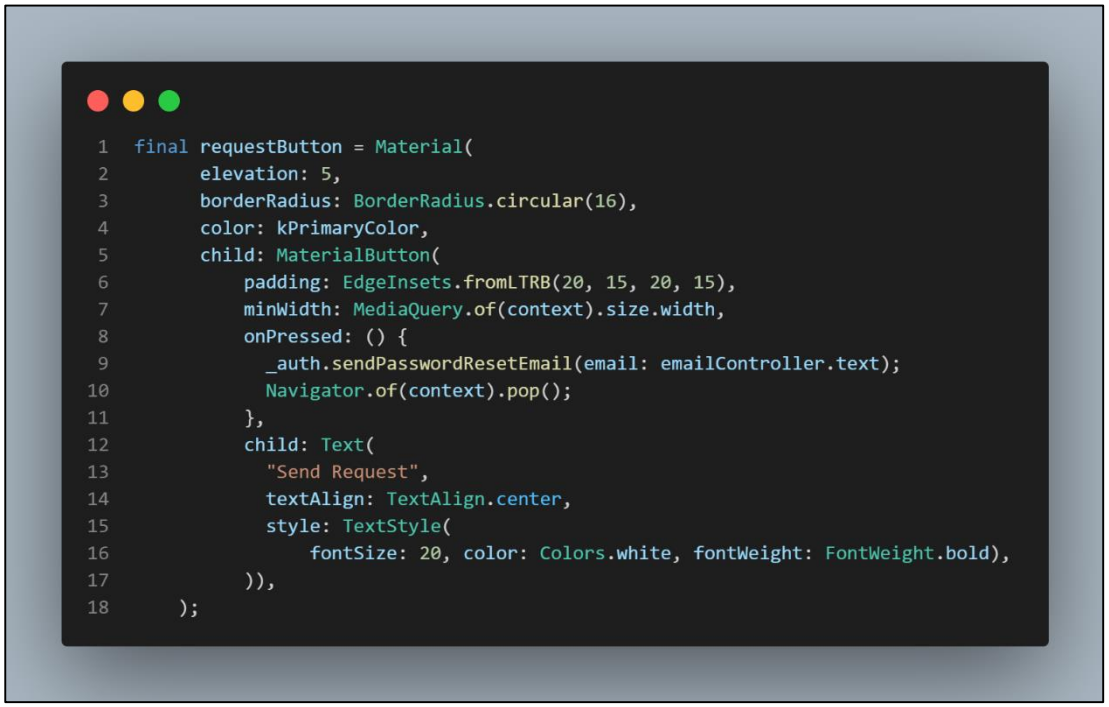
The implementation process guarantees that the finished programme meets defined standards, while the testing phase assures that the mobile application adhered to the standards.

4.1 Implementation

The implementation phase is the logical extension of the design. Dart Programming (Flutter Framework) is used to create the mobile application's front-end and back-end. Firebase Firestore Database, Firebase Storage, and Firebase SDKs have been added to connect the system to Firebase. The database for this proposed application is Firebase Firestore Database. Before the mobile applications may connect to it, it must be configured and set up. The mobile application's dependencies must be defined, as shown in Figure 7.

Figure 7: Dependency declared for Firebase Database Android

A password reset page is where users can send their reset password request. Users will receive a password reset email upon sending the request. The code segment forgets password function is shown in Figure 8.



```
final requestButton = Material(
      elevation: 5,
      borderRadius: BorderRadius.circular(16),
      color: kPrimaryColor,
      child: MaterialButton(
          padding: EdgeInsets.fromLTRB(20, 15, 20, 15),
          minWidth: MediaQuery.of(context).size.width,
          onPressed: () {
            _auth.sendPasswordResetEmail(email: emailController.text);
            Navigator.of(context).pop();
          },
          child: Text(
            "Send Request",
            textAlign: TextAlign.center,
            style: TextStyle(
                fontSize: 20, color: Colors.white, fontWeight: FontWeight.bold),
          )),
    );
```

Figure 8: A code segment for the password reset process.

The administrator can view all the book categories and books. Additionally, the administrator can add and edit or delete books. Figure 9 and Figure 10 illustrate the code segment for adding a book and the editing or deleting process of a book, respectively. The code used a book model to call all the aspects from the database.

```dart
void saveBook() async {
    // calling our firestore
    // calling our user model
    // sedning these values

    FirebaseFirestore firebaseFirestore = FirebaseFirestore.instance;
    DocumentReference books =
        FirebaseFirestore.instance.collection('books').doc();

    BookModel bookModel = BookModel();

    // writing all the values
    bookModel.bookID = books.id;
    bookModel.categoryName = catNameEditingController.text;
    bookModel.bookName = bookNameEditingController.text;
    bookModel.bookISBN1 = isbn1EditingController.text;
    bookModel.bookISBN2 = isbn2EditingController.text;
    bookModel.bookAuthor = authorEditingController.text;
    bookModel.bookQuantity = int.parse(bookQuantityEditingController.text);
    bookModel.currentQuantity =
        int.parse(currentQuantityEditingController.text);

    if (_formKey.currentState!.validate()) {
      books.set(bookModel.toMap());
      //FirebaseFirestore.instance.collection("books").add(bookModel.toMap
());
      Fluttertoast.showToast(msg: "Save successfully");

      Navigator.pushAndRemoveUntil(
          (context),
          MaterialPageRoute(builder: (context) => AdminMain()),
          (route) => false);
    }
  }
}
```

Figure 9: A code segment for the book adding process.

```dart
final deleteUser = FirebaseFirestore
    .instance
    .collection('books')
    .doc(widget.bookModel.bookID);

deleteUser.delete().whenComplete(() {
  Navigator.of(context).pop();
});

final updateUser = FirebaseFirestore
    .instance
    .collection('books')
    .doc(widget.bookModel.bookID);

updateBook.update({
  'bookName': bookName.text,
  'bookAuthor': bookAuthor.text,
  'bookISBN1': bookISBN1.text,
  'bookISBN2': bookISBN2.text,
  'bookQuantity': bookQuantity.text,
  'currentQuantity': currentQuantity.text,
}).whenComplete(() {
  Fluttertoast.showToast(
      msg: 'Updated successfully.');
  Navigator.pushAndRemoveUntil(
      (context),
      MaterialPageRoute(
          builder: (context) =>
              AdminMain()),
      (route) => false);
});
```

Figure 10: A code segment for editing and deleting book process.

The application's main login interface is shown in Figure 11. The students, teachers and administrator can choose their roles before login into the page. After selecting, the user will be directed to the appropriate login page for their role. Figure 12 depicts the login page for three types of users: students, teachers, and administrators.

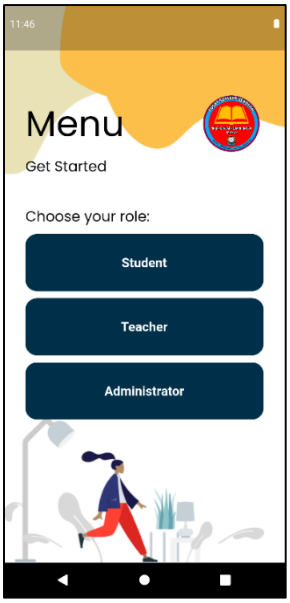


Figure 11: Main menu page.

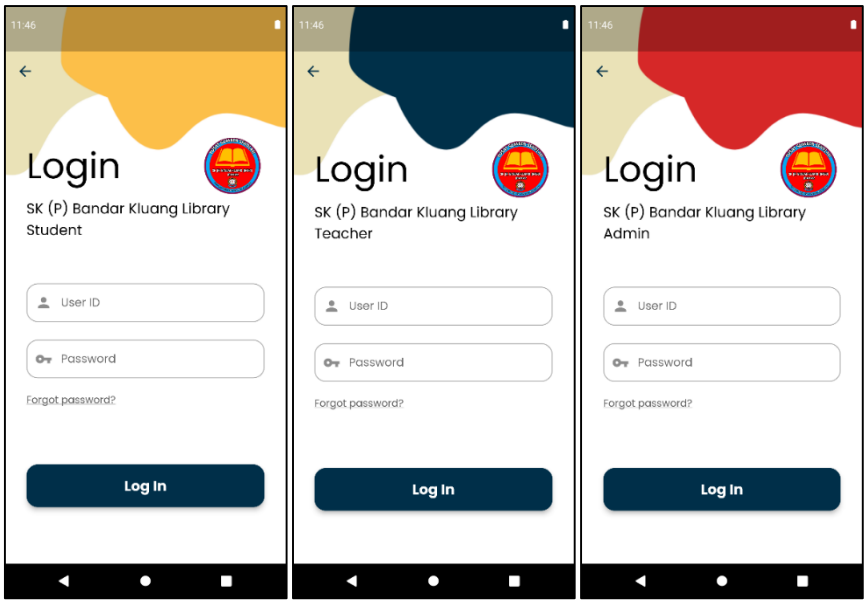


Figure 12: Student, Teacher and Administrator login page.

The administrator can view all the book categories and books. Books can be added by clicking the add icon on the top right corner of the page and it will navigate to an "Add Book" page where the administrator has to input all the related book information and the book edit and delete page are shown in Figure 13.

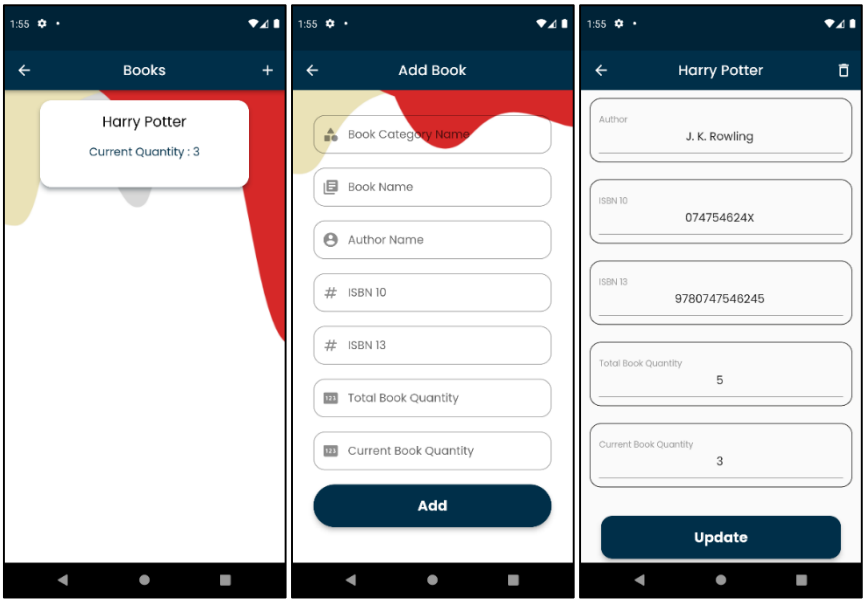


Figure 13: Book administration pages.

Figure 14 depicts the book borrowing page for the administrator. The administrator has access to all of the students who are borrowing books. On the add borrowing record page, where the administrator can enter the relevant borrowing information and dates to complete the process. Aside from that, when the student returns the books, the administrator can enter the return date. Figure 15 depicts a student's book borrowing record page. In addition, students can view the details of their borrowing records that have been updated by the administrator.

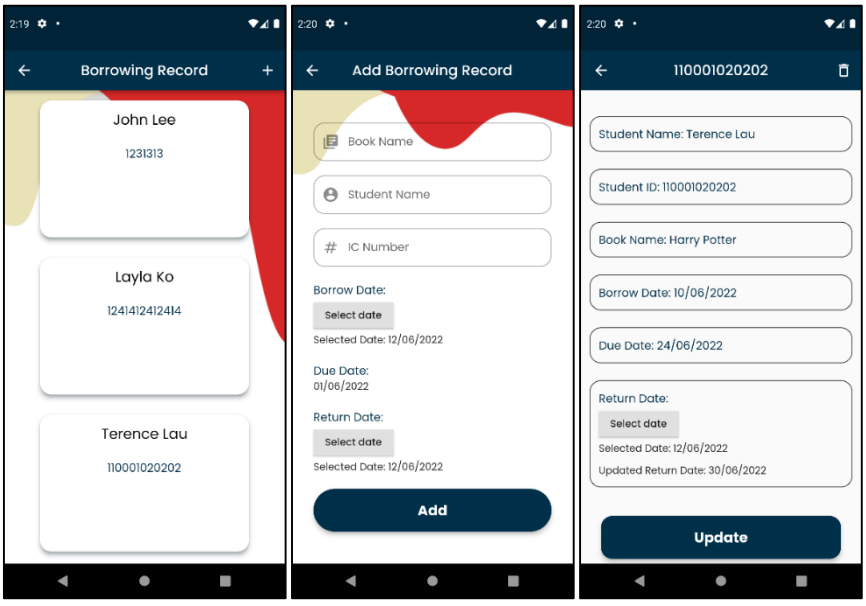


Figure 14: Book borrowing pages for the administrator.

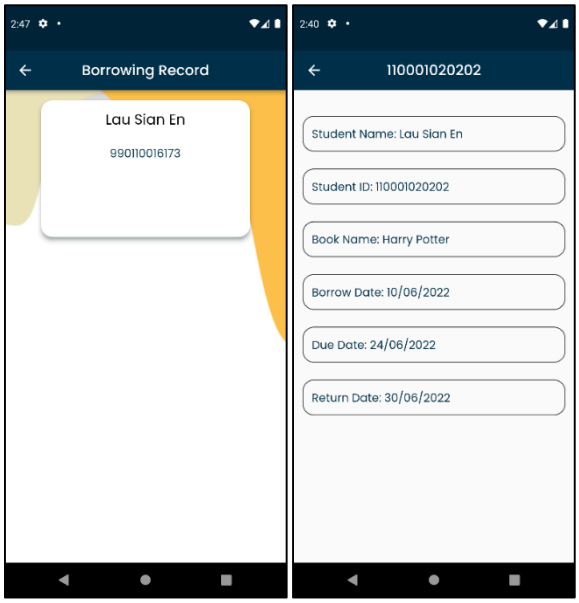


Figure 15: Book borrowing pages for students.

The NILAM report page for students is depicted in Figure 16. Students can find a list of NILAM reports that they have completed. Moreover, students can add a new NILAM report. The NILAM report details for students to edit or delete. Students are able to view the teacher's NILAM evaluation. Furthermore, teachers are able to access this module as well. Figure 17 shows the NILAM report page for teachers. Aside from that, teachers can access all of the NILAM reports completed by their students. Teachers can also review and give an evaluation of the NILAM report done by students.

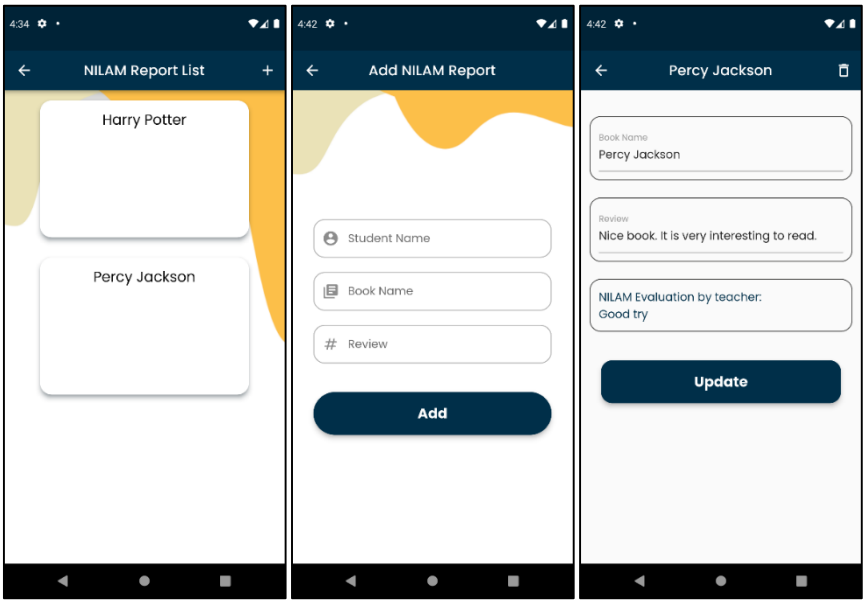


Figure 16: Students NILAM report page.

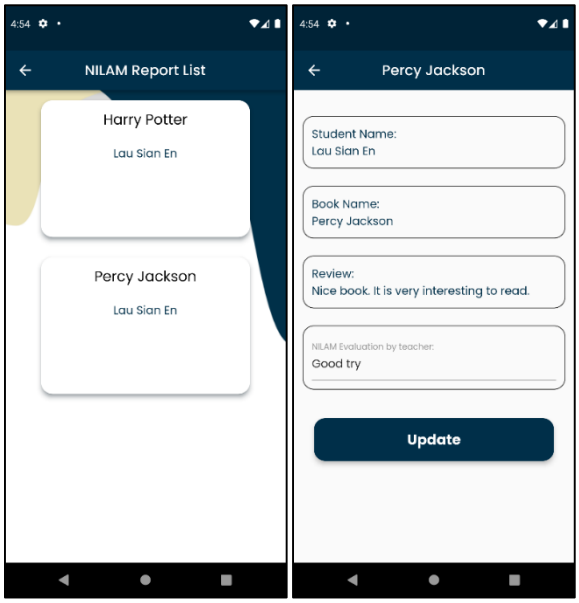


Figure 17: Teachers NILAM report page.

4.2 Functional Testing

Functional testing ensures that a software system meets its functional requirements. By giving appropriate input and comparing it to functional requirements, functional testing is used to check the output of a software program.

4.2.1 Test Plan

Table 5 shows the test plan for this mobile application. Login, Logout, Profile Edit, Book Borrowing, Book Records, NILAM Report, eBook Databases, Penalty Payment, and User Administration are the nine function modules that make up the test plan presented in Table 5.

Table 5: Test Plan

Function Modules	Expected Result	Actual Result
Login	1. Users are able to log in with a valid username and password.	Pass
	2. The login button is functioning.	Pass
Logout	1. This module enables users to log out from the application.	Pass
	2. The logout button is functioning.	Pass
Profile Edit	1. Users are able to edit their profiles.	Pass
	2. Users are able to update their passwords.	Pass
Book Borrowing	1. Students are able to borrow or return books from the library.	Pass

Table 5: Test Plan (cont)

Function Modules	Expected Result	Actual Result
	2. Students are able to check their borrowing records.	Pass
	3. Administrator is able to check students' borrowing records.	Pass
Book Records	1. Administrator is able to add/edit/delete the books and also check on the book records.	Pass
NILAM Report	1. Students are able to add or edit NILAM reports.	Pass
	2. Teachers are able to review and evaluate the NILAM report done by students.	Pass
eBook Databases	1. Students are able to access eBook databases.	Pass
Penalty Payment	1. Students are able to pay the penalty payment using the payment gateway.	Pass
	2. Administrator can check the penalty on the student using this module.	Pass
User Administration	1. Administrator is able to view, add, edit or delete students and teachers.	Pass

4.2.2 User Acceptance Testing

User acceptance tests are carried out using Google Form questionnaire. There are a total of users involved in this testing which are 10 students, 10 teachers and one administrator. The results for user interface evaluation, result of function modules evaluation for students, result of function modules evaluation for teachers and the result of function modules evaluation for administrator are shown in Figure 18, Figure 19, Figure 20 and Figure 21, respectively.

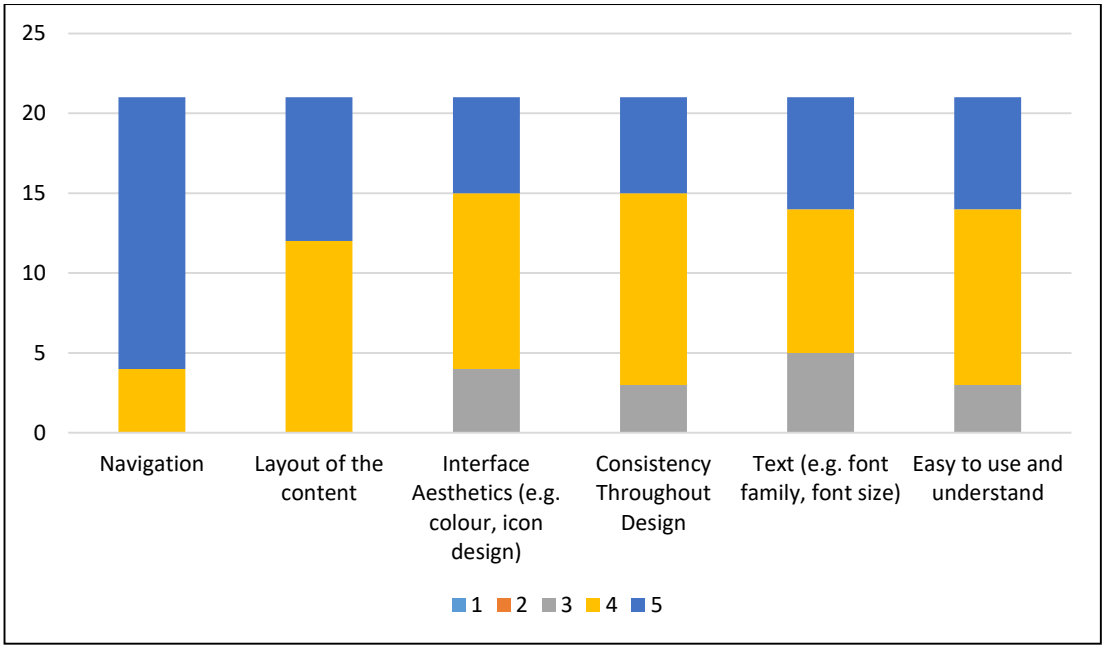


Figure 18: Result of user interface evaluation graph.

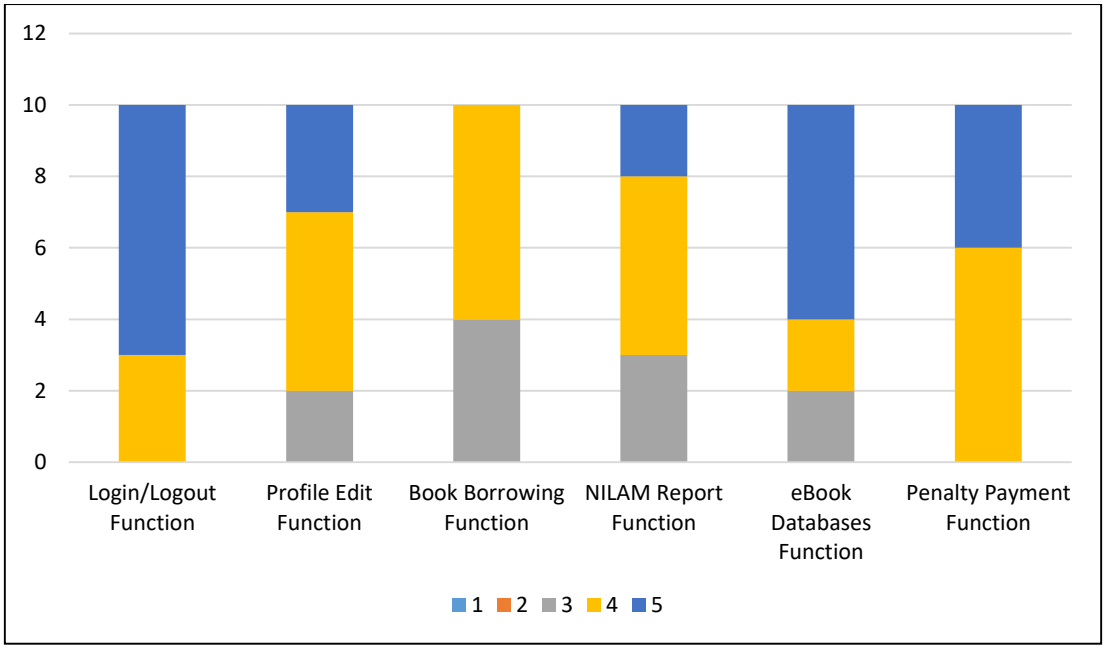


Figure 19: Result of function modules evaluation for students' graph.

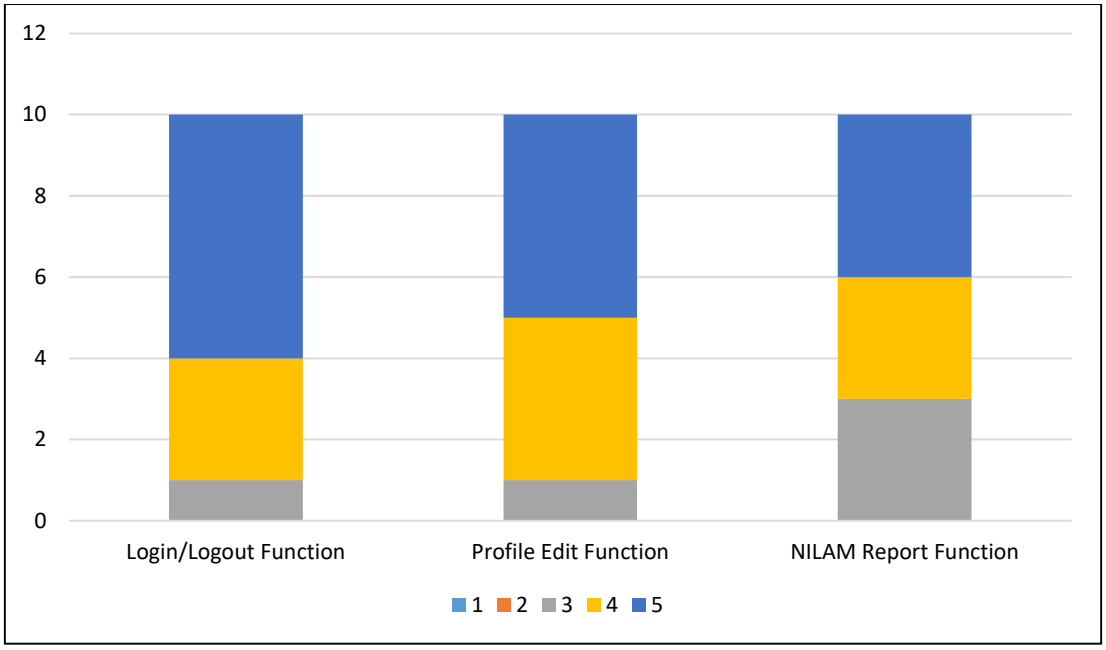


Figure 20: Result of function modules evaluation for teachers' graph.

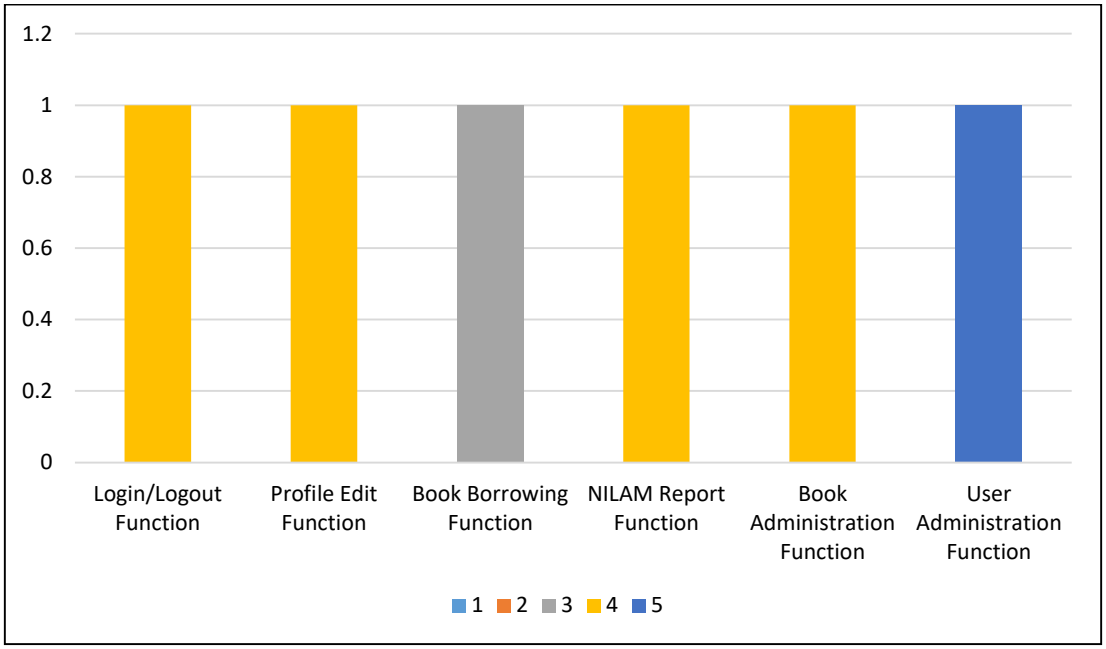


Figure 21: Result of function modules evaluation for the administrator graph.

5. Conclusion

The Library Management Mobile Application has been developed in order to benefit the library of SKPBK in library management areas including book borrowing, user administration and book administration replacing the old and manual system in the school. Students are able to add NILAM reports for the teachers to check without any difficulties. The mobile application is user-friendly in terms of book borrowing too. The administrator can perform the book borrowing process by using the mobile application which is very convenient.

Conclusively, the Library Management Mobile Application for the library of SKPBK has accomplished the objectives based on the project scope, system, and user requirements. The proposed mobile application has also helped the library operate more efficiently. Although the application may have some flaws, more work needs to be done to overcome these flaws and improve its functionality so that users in the library may receive a better experience in using the mobile application.

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] Gathoni, N. and Van der Walt, T., "Evaluating library service quality at the aga khan university library: Application of a total quality management approach". Journal of Librarianship and Information Science, 51(1), 123-136, 2019.

[2] Ambawat Yuvraj Patel, Divyanshu Sharma, Jayesh Sharma, Gourav Puri Goswami and Jitendra Sharma, Library Management System. International Research Journal of Modernization in Engineering, 2021.

[3] T, Manjula, "Library Mobile Apps: For Effective Services of Library". 5. 17-31, 2019.

[4] R. A. Juanatas and I. C. Juanatas, "RFID-Based Library Management System with Android Mobile Access Application," 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), 2019, pp. 270-274, 2019.

[5] (2021). Glibrary-Library Management System (Version 1.0.16) [Mobile app]. Retrieved from Google Play Store. https://play.google.com/store/apps/details?id=com.gayatrisoft.glibrary&hl=en&gl=US.

[6] (2021). AutoLib (Version 1.0.5) [Mobile app]. Retrieved from Google Play Store. https://play.google.com/store/apps/details?id=com.autolib.selfchecking&hl=en&gl=US.

[7] (2021). Libero [Mobile app]. Retrieved from Libero Official Website. https://libero.com.au/

[8] Okesola, Olatunji & Adebiyi, Ayodele & Owoade, Ayoade & Adeaga, Oyetunde & Adeyemi, Oluseyi & Odun-Ayo, Isaac, "Software Requirement in Iterative SDLC Model", 2020.

[9] Al Lail, M, "A unified modeling language framework for specifying and analyzing temporal properties" (Order No. 10840070). Available from ProQuest Dissertations & Theses Global. (2118688046), 2018.