FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

Diploma in Computer Science Year 1 Semester 3

Assignment

AMCS1003 SOFTWARE ENGINEERING

No.	Name	Registration No.	Tutorial Group	Signature	Marks
1.	Loh Yu Khai	23PMD06169	2	YuKhai	
2.	Calven Phnuah Kah Hong	23PMD05915	2	Calven	
3.	Ng Pui Yin	23PMD06254	2	PuiYin	
4.	Megan Yeoh Tze Xuan	23PMD11530	3	meg	
5.	Tan Ji Xian	23PMD09601	3	JMax	
6.	Sharwin Nair A/L Ramakrishnan	23PMD11528	2	Sharwin	

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

Plagiarism Statement and Guideline for Late Submission of Coursework

Read, complete, and sign this statement to be submitted with the written report.

We confirm that the submitted works are all our own work and are in our own words.

Name	Registration Number	Signature	Date
1. LOH YU KHAI	23PMD06169	YuKhai	12/5/2024
2. CALVEN PHNUAH KAH HONG	23PMD05915	Calven	12/5/2024
3. NG PUI YIN	23PMD06254	PuiYin	12/5/2024
4. MEGAN YEOH TZE XUAN	23PMD11530	meg	12/5/2024
5. TAN JI XIAN	23PMD09601	JMax	12/5/2024
6. SHARWIN NAIR A/L RAMAKRISHNAN	23PMD11528	Sharwin	12/5/2024

Table of content

Part 1

1.1 Problems of Existing System

The existing system which is studied in this assignment is used to support a community library operation. The library is located in the city centre and is open to the public. In the past year, the library has had about 100 staff members to support the daily operations, at the same time serving about 1000 visitors daily. The library has over 100,000 resources and approximately 80% of them are open for the public to borrow, varying from literature works to thesis paper materials. Moreover, the library provides a discussion room rental service for visitors to hold discussions or study sessions in groups. Additional services such as printing, photocopying and laminating also bring convenience to the visitors to deal with paper documents.

However, the old-fashioned operating method has been leading the organisation to a state of decline, as its inefficiency and disorganisation have been bringing loss and excessive workload to the organisation. As a result, the library management requested to develop an advanced and efficient digital information system to replace the existing operating method. The information system is expected to help the library increase income and fulfil the customer's satisfaction, therefore elevating the population of the community library among the public.

Before being able to develop the information system, a thorough analysis towards the existing system is crucial to specify the requirements which come in handy during development. Some of the main problems are being spotted from the current manual system and are listed as follows.

1.1.1 Human Error

Human errors here refer to mistakes or inaccuracies made by the library's staff while performing tasks. These errors may occur due to the staff's lack of attention, fatigue or simply just unintentional oversight.

The library's current manual system requires the staff to manually enter a substantial amount of daily transaction records. This manual input exposes the current system to a high risk of encountering human errors, particularly in the form of manual data entry errors. Such errors can result in data inconsistency, leading to many different types of issues such as inaccuracies in members' records, false status of availability of books and billing inaccuracies. A digital system can automate data entry tasks, thereby diminishing the concern about the occurrence of human errors. For instance, technologies like barcodes can be utilised by affixing barcode tags to each book, each containing its unique ISBN code, which can then be scanned for the entry of book data.

Moreover, one of the main sources of human errors is that manual data entries pose difficulty in validating their accuracy. Validate manual data requires additional library staff to manually cross-reference records and double-check the entries. For the same reason, a digital system will be able to automate data validation checks during the data entry process, hence decreasing the probability of human error. For example, if the alphabet is entered into the members' phone number section, a real-time error notification will be used to alert the staff.

1.1.2 High Time Cost

Time is a cost that the library cannot afford to lose, but using a manual system requires longer time to accomplish tasks. As the library operates, its vast collection of books will grow alongside it to the extent that it becomes complex and costly to handle with a manual system. Tasks such as recording those books and their respective transactions with paper and pen would require the library more labour to manage all that work. Besides, backing up these physical records brings challenges that make it less practical than digital backups. For instance, the process of photocopying the documents for backups requires significant resources that cause it to be very costly, space-consuming, and time-intensive.

A manual system will also cost time when a customer enquires about a book's availability. The staff has to go through shelves of paper records, with the risk of accidentally leaving out the book they are looking for. Consequently, not only will the library staff's workload increase, but customers might also get impatient and unsatisfied due to the long wait. In short, the situations mentioned above show how a manual system is inefficient and time-consuming for the library and how changes are urgently needed.

Introducing a new automated library system would disseminate the problems mentioned above. Adding new book records becomes a simple task for the library staff, as they can input book information using the keyboard, such as the author and book description. Additionally, a search function enables librarians to swiftly locate book details by entering relevant information into the search bar, eliminating the need to search through records manually.

1.1.3 Staff and Customers' Dissatisfaction

The library system is used by both the library staff and customers who visit the library. The existing system is unable to meet user's demands. The staff have a heavy workload due to reliance on physical handwritten records in data management. They are required to manually search and update the catalogues as well as the books' records. The process is time-consuming and often results in inaccurate data. Moreover, the customers of the library may experience frustration as they cannot access resources and services to meet their needs. The customers need to walk into the library to reserve a discussion room. Customers will experience longer wait times when checking out books during the book-borrowing process. This is because the staff will have to manually record transactions and stamp due dates on index cards.

Additionally, the existing system has low acceptability as it is hard for staff to learn and accept. When the new staff is onboarding, they are required to undergo training to learn how to effectively manage and familiarise themselves with the complex library manual system.

The manual system is perceived as low reliability. This is because the staff need to avoid data entry mistakes that could potentially result in providing inaccurate information such as incorrect due dates and book availability. This can result in customer frustration. The incorrect due dates of returning books will cause unforeseen fines to the customer. Furthermore, the customers cannot check the book availability remotely and visit the library physically to check if the book they are looking for is available. The dissatisfaction and inconvenience towards customers are the reasons contributing to the low reliability of the manual system.

To solve the problems mentioned above, the new library system can streamline the borrow and return books process by acquiring IC cards from customers and then scanning the barcode attached to the book. This solution enables faster checkouts and returns of books, as well as the accurate tracking of due dates, and also prevents inaccurate data from being input into the system. Other than that, customers can access the library system remotely to check the availability of the book without assistance from the library staff.

1.1.4 Risk of Data Leakage and Data Loss

A manual library system can face various challenges, including security issues. The confidentiality of patron information including borrowing history may not be adequately protected leading to privacy concerns. Likewise, the manual system often lacks systematic controls to ensure that only authorised personnel have access to patron information such as contact details, address and other identifiable information. The manual handling of records may increase the risk of unauthorised access and compromise the privacy of users. Similarly, physical records are susceptible to loss, theft or damage, risking the exposure of sensitive information. Users may lose trust in the library's ability to safeguard their information, leading to a decline in library usage. To address this, it is crucial to implement a controlled access system, ensure that only authorised users can access the sensitive patron information and introduce secure storage solutions such as encryption discipline and firewalls. A secure system ensures the protection of patron privacy, maintains the confidentiality of personal information and builds trust among library users, assuring that their sensitive data is protected.

In addition, physical storage infrastructure is prone to disaster. Physical storage locations, especially those within the library premises, may be at risk during natural disasters. For instance, floods can damage paper-based materials, and fires can destroy entire collections. Similarly, the vulnerability of physical storage can also be influenced by the geographical location of the library such as coastal areas may face higher risks. To address this issue, we allocate resources towards digital preservation solutions to protect collections against physical risk. This involves converting materials into digital forms and storing them securely such as cloud-based storage. By taking this solution, we can decrease our dependence on physical storage facilities.

1.2 Software Quality Attributes

To justify the quality of the software developed, the software must be assessed for its adherence to essential software quality attributes with the following aspects.

1.2.1 Acceptability

Acceptability refers to the level of system acceptance that a system needs to meet the requirements for a user to perceive it as usable and satisfactory. Achieving acceptability involves carrying out a user acceptance test to ensure the system achieves the specified requirements. For example, the functionality of the system is performed as it was designed to, with user-friendliness, and compatibility in different settings.

The new library system shall have the attribute of good software which is acceptability. The new system streamlines and simplifies library tasks meeting the users' needs and expectations. The staff will not need to manually track the books records and the customers can check the book availability remotely as well as online reservation. Moreover, the new system shall be designed with intuitive interfaces which can minimise the training requirements for staff. The staff can quickly adapt to the system and perform their tasks efficiently.

In addition, the new system shall have good usability. The new system shall have a user-friendly interface with clear navigation and consistent design elements. The interfaces for staff and customers will be designed to perform different tasks. As for the staff, the interfaces allow them to add, delete and update the library's catalogues. As for the customers, they can search for the books' information, borrow and return books online as well as make reservations for the discussion room online. Besides that, these features shall be learnable as the users can easily understand and become proficient in using the new system.

The new library system shall be understandable by using familiar terms for navigation and labelling. This can help users quickly understand the functions and features of the system. Additionally, meaningful error messages shall be implemented in the system to ensure optimal escapability. The errors made by the users shall be reported and clear options and guidelines shall be provided to correct the issue. For example, before the staff delete a book record, a confirmation message will be prompted before proceeding to the next process.

1.2.2 Dependability and Security

The dependability of a new library system is a critical aspect that encompasses its reliability, consistency and trustworthiness in delivering services. This is significant with the system uptime which reflects its ability to remain operational without disruptions and ensuring continuous access to resources. For example, the system shall have an uptime of 999 hours out of 1000 hours.

In addition, the implementation of the new library system must have a strong focus on key database management principles which are data integrity to ensure the dependability of the new system. The assurance of data integrity is the most important, whereby the system maintains accurate and consistent data to prevent misinformation. To maintain consistent data, error checking and validation play a crucial role in ensuring the accuracy and reliability of data within the system. It helps identify and correct mistakes or inconsistencies in the data which contributes to the database integrity. By implementing error-checking mechanisms, the system can promptly identify and address these errors to prevent the propagation of misinformation throughout the system.

Likewise, data redundancy may impact efficiency and data consistency. In a library system, avoiding excessive duplication of information is crucial to maintain efficiency and reduce the risk of inconsistencies. Reducing data redundancy saves database size, optimises data retrieval and reduces processing times. Similarly, minimising redundancy contributes to data consistency which ensures that updates or changes made to one instance of information are reflected equally across the system.

Moreover, system downtime will directly affect the availability of a system. Downtime will interrupt the essential library services, such as searching catalogues, borrowing books and returning books. This disruption not only inconveniences users but also creates a cascading effect, impacting the efficiency of the library staff. To prevent system downtime in our library system, we can employ several solutions. Initially, we can schedule regular software updates and maintenance to address weaknesses and ensure system stability. Additionally, by implementing backup and disaster recovery, we can conduct a backup routine towards the library system's data, providing a safeguard against potential data loss during unforeseen events.

On the other hand, having outstanding security contributes to the system's reliability. System failure or information leak through SQL injection and cross-site scripting attacks shall be fully prevented. It doesn't only cause losses to the organisation, confidential information leakage also will cause public distrust and therefore undermine the library's credibility. Certain measures should be taken to cope with the risks above, such as updating the system to be equipped with the latest technology to leverage the system's security to the maximum extent possible.

1.2.3 Performance

A performant information system is capable of reacting to the user's response quickly. The low response time results in a better user experience, customer satisfaction, and hence better business outcomes. To achieve the aspect mentioned above, the system should address the issue of server withstanding high network traffic. Other than purchasing server computers with better specifications, a more cost-effective approach is implementing a load-balancing strategy. Load balancing is an approach to distribute the incoming network traffic across multiple server workers. This effectively reduces the possibility of an imbalance in server workload which could cause the server with a high workload to throttle the performance of the system. In the end, the server could respond to the clients more invariably.

Besides, the new library system should be able to utilise every second by deducing unnecessary and tedious work for the staff. It means the system has to be automated, with each task done quickly without delay. For instance, when a customer wishes to borrow a book, the staff should be able to obtain the information about the book by scanning the book's barcode instead of entering the book's code by hand. Such changes will increase the staff's work speed and reduce the time required for the borrowing process, ensuring an efficient workflow. The new library system could also shorten the time for various tasks, such as searching the availability of a book, updating existing book records or even adding new books to the library's collection.

1.2.4 Maintainability

Maintainability poses as one of the most important software quality attributes, reflecting the library system's capabilities for effective upkeep, repair, and updates. The library with high maintainability system software can significantly reduce the total downtime required for maintenance. Thereby, reducing frustrations of users by ensuring smooth and uninterrupted access to the library services.

Furthermore, the new library software can achieve high maintainability through the implementation of a highly modular software architecture. High modularity defines that the software is divided into simpler, task-focused modules, hence facilitating code comprehension. Moreover, a high modularity software design encourages attributes such as high cohesion and loose coupling.

High cohesion ensures that all instructions within a module of the library software collectively contribute to a single task, building a well-structured and easily debugged system. This structural integrity helps to improve the maintainability of the new library system. On the other hand, loose coupling orientates more on minimising dependence between modules, offering flexibility and preventing issues on one specific module from affecting other modules. In case any of the modules fails to function, it will not hinder other processes of the library system.

Additionally, a high modularity library software can be achieved by defining clear boundaries between modules and isolating the functionality of each module to reduce the potential for unintended interdependencies between modules.

1.3 Software Process Model

The software process model is a series of specific tasks to be carried out to develop a software system. Choosing a suitable software process model helps the software development team to develop the software in a structured way, making sure the software can be produced with high quality in a minimal amount of time, at the same time meeting the client's expectations. For the proposed library system, the **Rapid Application Development model (RAD)** has been chosen to be the software process model used in developing it.

1.3.1 Methodology

RAD is a software development model that emphasises rapid prototyping and quick feedback with less specific planning. This development methodology is well-suited for building library systems due to being faster and more flexible than traditional waterfall models, and it is well-suited for projects with frequent changes in user requirements and a need for quick delivery.

The activities carried out are as follows. At the very beginning, requirements are gathered by having discussions with staff involved such as library management and librarians to understand their needs and what functionalities are essential for the library system. Then, a basic version of the system with the core functionalities shall be developed. The system will be broken down into multiple functional modules. Each module is assigned to a separate team of developers based on their expertise to work concurrently on their assigned module. As modules are completed, they are integrated into a system that is not a fully fleshed-out system, but rather a functional model to demonstrate the key features to the client. User involved will get to test the prototype and provide their feedback on usability, functionality, and any missing features. The development team then takes the user feedback and makes improvements to the prototype. This might involve adding functionalities, changing the interface, or fixing any bugs encountered during testing to run final updates and maintenance tasks before going live.

1.3.2 Suitability

RAD is fundamentally suitable for library systems operating in today's dynamic environments. Firstly, the emphasis on fast development translates into significant time-to-market advantages. This accelerated the speed which allowed the library system to promptly introduce new products, and features and gain a competitive advantage by being first to market and evolving customer needs more rapidly.

Furthermore, due to the tight schedule for the implementation of the new library system, additional resources will be allocated to guarantee the excellence of the final product. Given the library's long history of operation, it is well-equipped to seamlessly transition to the RAD model with significant financial resources to facilitate the development process.

Additionally, with a large developer team in the library's arsenal, the library can effectively utilise this software processing model. This is due to the library's capability to be able to distribute the task of developing components and features of the library system equally among team members. This not only prevents individual developers from being overwhelmed by their allocated tasks but also ensures optimal utilisation of resources.

Moreover, the availability of large teams also facilitates parallel development, allowing developers to work on different aspects of the library simultaneously. This parallel development approach of RAD greatly enhances the pace of progress by leveraging the expertise and resources of the developer team. Besides, parallel development also reduces inter-team dependencies, thus granting more flexibility throughout the development process.

1.3.3 Principles

Timeboxing

Rapid Application Development is a software process model prioritising timeboxing as a core principle. Timeboxing involves allocating specific periods, known as timeboxes, for activities within the development process. In RAD, strict adherence to deadlines is enforced to guarantee the tasks are completed within their designated timebox, thereby ensuring the development cycle remains on track.

This leads to software systems developed using the RAD model often requiring accelerated completion, typically within 90 days or less. This approach is plausible due to the relatively smaller scale and complexity of systems involved, which in this case is the library system, where it tends to have functionalities and modules that are relatively straightforward and smaller, making RAD an effective implementation strategy. While developing the system, several teams will also collaborate to work on different modules of the library system simultaneously. Hence, multiple modules can progress concurrently by dividing the workload across various teams, accelerating the overall development process.

Timeboxing presents several notable business benefits for the library, such as ensuring development processes follow predefined timeframes. This results in expedited delivery of the final product, allocating additional time for staff to adapt to the changes introduced by a new system.

Prototype

In Rapid Application Development (RAD), the prototype plays a crucial role in accelerating the development process and ensuring that the final project meets the user's needs and expectations. RAD prioritizes building functional prototypes quickly, these prototypes do not need to be feature-rich, but they should demonstrate the core functionality of each module. This allows for faster development cycles compared to traditional heavy planning approaches.

The RAD often works well with modular development, where the application or project is broken down into smaller, independent modules. Each module focuses on a specific subset of functionality. Each development team will take each module and produce prototypes for each. The evolutionary prototypes are presented to users early in the development process to get user feedback and validation. This early-stage involvement becomes a key component of the RAD process. After that, the iteration development approach where prototypes evolve based on user feedback to refine the design, requirements and functionality. Within each iteration, prototypes are refined to incorporate feedback, leading to continuous enhancement in design, requirement and functionality. This iteration process ensures that the final product meets the user needs and expectations.

Frequent User Involvement

On the other hand, enhanced user involvement is another principle of this software process model. Users are actively involved in gathering and validating the system's requirements which ensures that their expectations are accurately captured by the system analyst. In addition, system analysts and the development team can prioritise the most valuable functionalities in the prototyping of the library management system by actively engaging customers in the phase of development. This reduces the risk of misalignment with project and business goals, prevents the need for system rework, and saves time and resources.

The RAD model allows the changing of requirements for the underlying problems of the system to be solved at the early stage of development as the users and developers will have rapid communication. The development team will involve the users frequently throughout the development process to verify the completion of various functions. The continuous user feedback and testing will provide the development team with valuable insights into how most of the users will interact with the system. By analysing the interaction between users and the system, they can simplify the processes and improve the user experience.

Use of Reusable Components

Component-Based Software Engineering (CBSE) is an approach involving crafting and refining computer-based systems through the utilisation of reusable software components. The components of CBSE can be sourced from various places, including external libraries, third-party vendors, or even developed in-house. In the context of a library system, examples of components could include modules for cataloguing books, managing user accounts, and generating reports.

CBSE provides the developers with a repository of reusable components which diminishes the need for developers to build software components from the ground up. Thereby speeding up the system development process. Especially when the RAD model requires the developer to produce a prototype in the shortest period of time, CBSE is a perfect pair for the RAD model. Not to mention, the fact that reusable components from outside sources originally will have already undergone thorough testing, validation, and compliance checks, developers can have confidence in the reliability and stability of these components. This quality assurance of components sourced from external providers significantly reduces the time and effort needed for the developer to do validation checks for risks of errors and defects in the to-be-implemented system. Thus, it also greatly assists in reducing the time needed for the development process.

Part 2

2.1 Project Plan and Schedule

A total of 15 developers are involved in this library system's development. The development plan is as the Gantt Chart:

Activities	Details	Start Date	Task Duration (month)	Predecessors	Overlap	April								May								June							
1	**Project Planning**																												
1.1	Identify Stakeholder	0	1/8	None	None	🟥																							
1.2	Conduct Feasibility Study	1/8	1/4	1.1	None		🟥	🟥																					
1.3	Define project scope and plan project	3/8	1/4	1.2	None				🟥	🟥																			
1.4	Interview	5/8	1/4	1.3	None						🟥	🟥																	

1.5	Document Inspection	5/8	1/4	1.4	None						🟥	🟥																	
1.6	Allocate task and resources	7/8	1/8	1.5	None								🟥																
2	**<u>Module Development</u>**																												
2.1	**Member Module**																												
2.1.1	Initial requirement analysis	1	1/4	1	None									🟥	🟥														
2.1.2	Prototype	5/4	1/4	2.1.1	None											🟥	🟥												
2.1.3	Testing	11/8	1/4	2.1.2	1/8												🟥	🟥											
2.1.4	Feedback	13/8	1/8	2.1.3	None														🟥										
2.2	**Staff Module**																												

2.2.1	Initial requirement analysis	1	1/4	1	None									🟥	🟥														
2.2.2	Prototype	5/4	1/4	2.2.1	None											🟥	🟥												
2.2.3	Testing	11/8	1/4	2.2.2	1/8												🟥	🟥											
2.2.4	Feedback	13/8	1/8	2.2.3	None														🟥										
2.3	**Book Maintenance Module**																												
2.3.1	Initial requirement analysis	1	1/4	1	None									🟥	🟥														
2.3.2	Prototype	5/4	1/4	2.3.1	None											🟥	🟥												
2.3.3	Testing	11/8	1/4	2.3.2	1/8												🟥	🟥											
2.3.4	Feedback	13/8	1/8	2.3.3	None														🟥										

2.4	**Book Transaction Module**																												
2.4.1	Initial requirement analysis	1	1/4	1	None									🟥	🟥														
2.4.2	Prototype	5/4	1/4	2.4.1	None											🟥	🟥												
2.4.3	Testing	11/8	1/4	2.4.2	1/8												🟥	🟥											
2.4.4	Feedback	13/8	1/8	2.4.3	None														🟥										
3	**Module Integration and Link Test**	7/4	1/4	2	None															🟥	🟥								
4	**System Test and User Acceptance Testing**	2	1/2	3	None																	🟥	🟥	🟥	🟥				
5	**Implementation and Deployment**																												

5.1	Hardware and Configuration Setup	9/4	1/2	4	1/4																			🟥	🟥	🟥	🟥		
5.2	System Conversion	5/2	1/4	5.1	1/4																					🟥	🟥		

Table 2.1 Gantt Chart

2.2 Software Requirements Specification

Software Requirements Specification is a document that describes the requirements collected from the user. It clearly defines the capabilities of the system to be developed and serves as a guide for the development team to follow during the development stage. Within this document, two categories of requirements describe the expected system, which are functional requirements and non-functional requirements.

2.2.1 Functional Requirements

Functional requirements describe the expected module functionality that will be integrated into the final system. The modules are expected to support the business activities through automation therefore increasing the benefits towards the organisation. There are four essential modules which should be included in the system developed which are as follows.

Staff Module

The staff module serves as a centralised platform to authorise and manage staff members, which allows them to use the library system to perform various administrative tasks and manage collections, such as staff authentication and account activation by specific staff.

1. The system shall allow the manager to add and activate the new staff's account to access the system.
2. The system shall allow staff to log in using their assigned staff ID and password.
3. The system shall restrict access if the password input does not match the associated staff's account.
4. The system shall maintain an audit trail of staff activities, including login/logout timestamps, actions performed, and the staff member responsible for each action.
5. The system shall provide the permissions to the staff to access various functionalities.

Member Module

The member module is a centralised module for library customers who register as members to utilise the functions of the system. The member authentication and profile management will be included in the member module.

1. The system shall allow members to register their personal information to access the system.
2. The system shall allow members to log in using their IC number as username and password.
3. The system shall restrict access if the password input does not match the member's account.
4. The system shall allow users to choose the "forgot password" option to reset their password.
5. The system shall allow members to update their personal information.
6. The system shall allow members to view the available books.
7. The system shall allow members to track their borrowing history and handle fines for overdue book transactions.

Book Management Module

The book management module poses as one of the vital modules that entails multiple crucial functions that are required for the library's operation. The main focus of this module is keeping the book data of the library organised and authorising users to manage existing book records. Utilisation of this module allows the library to have better management such as add, modify, delete the library's book data.

1. The system shall only authorise staff to access the add new book records function.
2. The system shall provide an input form to allow staff to add new book information including input fields such as ISBN, title, author, publisher, publishing date, genre and any other relevant information about the book.
3. The system shall validate the format of ISBN and date entered into the add new book form before storing the information in the database.

4. The system shall store the new book information in the database at the end of the book-adding process.
5. The system shall allow staff to make modifications to book details except for the book's ISBN.
6. The system shall allow staff to remove book records from the database.
7. The system shall provide a request form for the users to fill out the book title, author, preferred format and a brief justification for the requested book.
8. The system shall check if a request form's required fields are filled out correctly before submitting it.
9. The system shall be able to check whether the requested book exists in the library system.
10. The system shall generate a monthly report on books requested sorted by the number of requests per book in descending order.
11. The system shall notify users about the book's arrival upon successfully adding the book requested by the users.

Book Transaction Module

The book transaction module is an essential part of the library system to allow members to borrow and return books. This module ensures that all of the transaction processes can be handled efficiently and the transaction records are well maintained to be referenced in the future.

1. The system shall accept the book borrower's identity card (IC) number, phone number, and the ISBN book they want to borrow as input during the book processing process.
2. The system shall check whether the borrowers have returned a borrowed book before they make a new borrow.
3. The system shall compute the due date for a borrow according to the book borrowed classification.
4. The system shall display the borrowed book details, return due dates, and customer information on the screen for confirmation purposes.
5. The system shall update the book availability status from 'available' to 'borrowed' upon a borrow has been made.

6. The system shall print a borrowing slip at the end of the book borrowing process, containing borrowers' IC numbers, phone numbers, borrowed book information, transaction timestamp, and due dates.
7. The system shall receive the ISBN of the book to be returned, the user’s IC and the contact number as input during the return process.
8. The system shall compute and display the overdue fine if the book is returned late depending on the returned book’s category and the total days overdue.
9. The system shall compute and display the damage fine when the librarian inputs the returned book’s damage to the system.
10. The system shall update the book availability status from ‘borrowed’ to ‘available’ during the book returning process, or ‘unavailable’ if the returned book’s damage is being specified.
11. The system shall print a return slip at the end of the return process which includes the information of the returned book, date of return and overdue or damage fine if exists.

2.2.2 Non-Functional Requirements

Product Requirements

1. The success rate of the system's functionalities and accomplishing the specific tasks in the library system is expected to achieve a rate of 100%.
2. The library system should achieve the principles of learnability whereby the users can familiarise themselves with the whole system within less than 45 minutes.
3. The library system shall be able to handle the errors occurring effectively by validating the user input such as checking the required fields and failed and the character limits with prompting meaningful error messages as well as allowing the user to retry.
4. The library system should allow at least 3,000 users to access and interact with the system concurrently without performance throttling.
5. The library system must be available to users for 99.98% per 1,000 hours of operation.
6. During peak hours, the library system is anticipated to handle 1,000 data transactions per minute which includes processes such as borrowing, returning and renewing books to ensure the smooth operation of the library.
7. The system shall be designed to minimise the potential impact of known vulnerabilities such as SQL injection by implementing security practices such as input validation and sanitization.
8. The library system shall be designed to achieve a mean-corrective-maintenance-time (MCMT) of no more than 30 minutes and a max-corrective-maintenance-time (MAXCMT) of 2 hours through practices such as having comprehensive system documentation and high modularity design

Organisational Requirements

1. The library system must implement data encryption to protect sensitive patron information.
2. The library system must implement regular backup and recovery procedures.
3. Infrastructure resources such as servers should be located suitably to meet the performance and scalability requirements of the system.
4. The library system shall be a closed-source software and provide the least possible privilege of source code section to the developer to minimise the occurrence of unauthorised access.
5. The library system shall be developed to emphasise reusability through high modularity so the components are reusable in other projects which could save time and effort.
6. The library system shall use a version control system to ease tracking changes and allow rollback.
7. The library system should be designed and operated in a way that avoids discrimination and bias in its services and interactions with users.

External Requirements

1. The library system must comply with the Personal Data Protection Act (PDPA) by implementing privacy policies that inform users about the collection, use, and disclosure of personal information during data practices.
2. The library system must adhere to copyright laws and regulations governing the use of copyrighted materials ensuring that the system does not violate the copyrights of authors, publishers, and other rights holders when providing access to library materials.

2.3 Architectural Design

The system organisation design of the library system is represented as shown in the box diagram below. The library system has 4 modules, which are member module, staff module, book management module and book transaction module. The users who log in as members will be given access to the functionality of other modules as mentioned, such as requesting new books and searching for books in the library. Meanwhile, the users who log in as staff will be provided more privileges towards the system functionality, such as maintaining the book records and conducting the book borrowing and returning process.

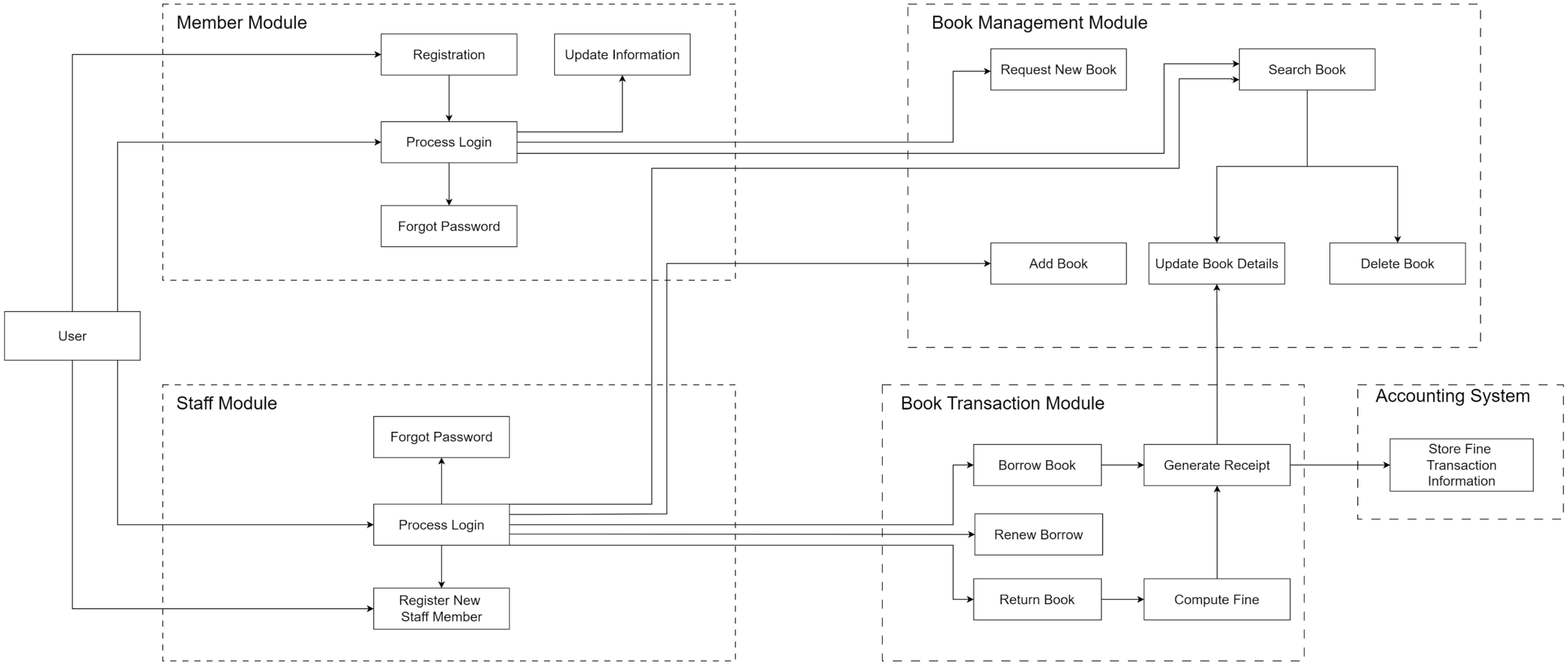


Figure 2.3 System Architecture Design (Box Diagram)

Part 3

3.1 Test Cases

3.1.1 Staff Registration Function

<table>
<tr><td colspan="6">Program Name: Staff Registration Function
Description: To test the staff registration function with valid staff data</td></tr>
<tr><td colspan="3">Test Date: 22/4/2024</td><td colspan="3">Tester: SHARWIN NAIR A/L RAMAKRISHNAN</td></tr>
<tr><th>No.</th><th>Test Cases</th><th>Test Data</th><th>Expected Result</th><th>Actual Result</th><th>Remarks/Comments</th></tr>
<tr><td>1.</td><td>Navigate to the staff registration page</td><td></td><td>The staff registration page should be displayed after navigation.</td><td></td><td></td></tr>
<tr><td>2.</td><td>Key in the Identification Card No Field</td><td>Valid Identification Card No value: 050520094556</td><td></td><td></td><td></td></tr>
<tr><td>3.</td><td>Key in the Name Field</td><td>Valid Name value: Mathew</td><td></td><td></td><td></td></tr>
<tr><td>4.</td><td>Key in the Phone No. Field</td><td>Valid Phone No. value: 0121234567</td><td></td><td></td><td></td></tr>
<tr><td>5.</td><td>Select the Role Field</td><td>Valid role value: Admin</td><td></td><td></td><td></td></tr>
</table>

6.	Select the Gender Field	Valid gender value: Male			
7.	Key in the Email Field	Valid Email: jai34@gmail.com			
9.	Key in the Password Field	Valid password: Lemons0990			
10.	Key in the Confirm Password Field	Valid password: Lemons0990			
11.	Click on Register Button		The system should alert the user with the message "Successfully registered," and display the Login Page.		

Program Name: Staff Registration Function
Description: To test the staff registration function with invalid staff data

Test Date: 22/4/2024

Tester: SHARWIN NAIR A/L RAMAKRISHNAN

No.	Test Cases	Test Data	Expected Result	Actual Result	Remarks/Comments
1.	Navigate to the staff registration page		The staff registration page should be displayed after navigation.		
2.	Key in the Identification Card No. Field	Invalid Identification Card	The input field bottom should show an error		

		No. value: 021088715	message of “IC No. must be at least [12] digits long”.		
3.	Key in the Phone No. Field	Invalid Phone No. value: 012-2312233	The input field bottom should show an error message of “Please enter without ‘-’.”		
4.	Key in the Email Field	Invalid email: hetds23@gmailcom	The input field bottom should show an error message of “Please enter a valid email.”.		
5.	Key in the Password Field	Invalid password: abcde	The input field bottom should show an error message of “Password must contain numeric value”.		
6.	Click on Clear Button		The system should be able to empty all the filled fields.		

3.1.2 Member Registration Function

Program Name: Member Registration Function Description: To test the member registration function with valid member registration data					
Test Date: 22/4/2024			Tester: NG PUI YIN		
No.	**Test Cases**	**Test Data**	**Expected Result**	**Actual Result**	**Remarks/Comments**
1.	Navigate to the member registration page		The member registration page should be displayed after navigation.		
2.	Key in the Identification Card No Field	Valid Identification Card No value: 051207070509			
3.	Key in the First Name Field	Valid First Name value: Adam			
4.	Select the Gender Field	Valid gender value: Male			
5.	Key in the Email Field	Valid Email: adamtan@gmail.com			
6.	Key in the Phone No. Field	Valid Phone No.: 0123456789			
7.	Key in the Password Field	Valid Password: abcd12345			
8.	Key in the Confirm	Valid Password:			

	Password Field	abcd12345			
9.	Click on Register Button		The system should alert the user with the message “Successfully registered,” and display the Login Page.		

Program Name: Member Registration Function
Description: To test the member registration function with invalid member registration data

Test Date: 22/4/2024

Tester: NG PUI YIN

No.	**Test Cases**	**Test Data**	**Expected Result**	**Actual Result**	**Remarks/Comments**
1.	Navigate to the member registration page		The member registration page should be displayed after navigation.		
2.	Key in the Identification Card No field	Invalid Identification Card No value: 051207-07-0509	The input field bottom should show an error message of “Please enter without ‘-’. ”.		
3.	Key in the Email Field	Invalid Email: adamtangmail	The input field bottom should show an error message of “Please		

			enter a valid email.”.		
4.	Key in the Password Field	Invalid Password: def098	The input field bottom should show an error message of “Please enter a password with minimum 8 characters.”.		
5.	Key in the Confirm Password Field	Invalid Password: abc1234 (The Password field is filled with the valid password of “abcd1234”)	The input field bottom should show an error message of “Passwords are not matching.”.		
6.	Click on Register Button		The system should alert the user with the message “Failed to register, please fix the errors according to the message displayed”.		
7.	Click on the Clear Button		When the Clear button is clicked, all the values filled in the input boxes should be cleared.		

3.1.3 Book Borrowing Function

Program Name: Book Borrowing Function Description: To test the book borrowing function with valid book borrower data.					
Test Date: 22/4/2024			Tester: TAN JI XIAN		
No.	**Test Cases**	**Test Data**	**Expected Result**	**Actual Result**	**Remarks/Comments**
1.	Navigate to borrow book page		The system should display the borrow book page		
2.	Enter User NRIC	Valid User NRIC: 051207070509	The name and phone number of the returner should be auto-filled into their respective fields, which are Adam and 0123456789. The borrower eligibility status of the customer will also be displayed.		
3.	Add book into the borrow list	Valid Book ISBN: 9780241350966	The system should display the book's details, borrow date and also generate and display the return due date into the borrow list.		

4.	Click on the 'X' button		The system should remove the book where the 'X' button is clicked. A message should be displayed including the name of the removed book with an "Undo" button and an "OK" button.		
5.	Click on the "Undo" button		The "Undo" button should be able to reverse the book removal.		
6.	Click on the "Confirm" button		The message "Book Borrowed Successfully." should be displayed. The transaction receipt should be printed with the details of the book borrowed, borrow date and return due date.		

Program Name: Book Borrowing Function Description: To test the book borrowing function with invalid book borrower data					
Test Date: 22/4/2024			Tester: TAN JI XIAN		
No.	**Test Cases**	**Test Data**	**Expected Result**	**Actual Result**	**Remarks/Comments**
1.	Navigate to borrow book page		The system should display the borrow book page		
2.	Enter User NRIC	Invalid User NRIC: 051207-07-0509	The input box should turn red indicating the input NRIC is wrong.		
3.	Borrow Eligibility Status		The system should display "Not Eligible" in red text if the NRIC is invalid.		
4.	Add book into the borrow list	Invalid Book ISBN: 978024135	The system should disable the "Enter Book ISBN" field if the NRIC is invalid. No input should be allowed at the field.		
5.	Click on the "Confirm" Button		The system should disable the "Confirm" button if the NRIC is invalid.		

3.1.4 Book Returning Function

<table>
<tr><td colspan="6">Program Name: Book Returning Function
Description: To test the book returning function with valid book returner and defect data.</td></tr>
<tr><td colspan="3">Test Date: 22/4/2024</td><td colspan="3">Tester: LOH YU KHAI</td></tr>
<tr><th>No.</th><th>Test Cases</th><th>Test Data</th><th>Expected Result</th><th>Actual Result</th><th>Remarks/Comments</th></tr>
<tr><td>1.</td><td>Navigate to the book returning page</td><td></td><td>The book returning page should be displayed after navigation.</td><td></td><td></td></tr>
<tr><td>2.</td><td>Key in the NRIC field</td><td>Valid NRIC value: 051207070509</td><td>The name and phone number of the returner should be auto-filled into their respective fields, which are Adam and 0123456789.

At the same time, the book borrowed, Wizard of the Oz with the details, which is the author’s name and date of due should be displayed on the right-hand side panel.

The overdue day of 3 and an overdue fine of RM0.60 should be displayed.</td><td></td><td></td></tr>
</table>

3.	Add book defects through the pull-down menu and the quantity of said defects of the book returned	Valid Defection: Minor Tear Defection Quantity: 2	The defect selected and defection quantity should be appended to the defection list. The amount of defection fine should be displayed, which is RM4.00. The total fine amount should be increased from RM0.60 to RM4.60		
4.	Click on the return button		The message "Book is successfully returned." should be displayed. The transaction receipt should be printed with the details of the book returned and a list of various fines.		

<table>
<tr><td colspan="6">Program Name: Book Returning Function
Description: To test the book returning function with invalid book returner data.</td></tr>
<tr><td colspan="3">Test Date: 22/4/2024</td><td colspan="3">Tester: LOH YU KHAI</td></tr>
<tr><th>No.</th><th>Test Cases</th><th>Test Data</th><th>Expected Result</th><th>Actual Result</th><th>Remarks/Comments</th></tr>
<tr><td>1.</td><td>Navigate to the book returning page</td><td></td><td>The book returning page should be displayed after navigation.</td><td></td><td></td></tr>
<tr><td>2.</td><td>Key in the NRIC field</td><td>Invalid NRIC value: 051207-07-0509
(NRIC value should not have dashes)</td><td>The input box should turn red and display the error message of “Invalid NRIC value” at the bottom of input box.</td><td></td><td></td></tr>
<tr><td>3.</td><td>Key in the Name field</td><td>Invalid Name value: 123Hi
(Name value should not have number)</td><td>The input box should turn red and display the error message of “Invalid name value” at the bottom of input box.</td><td></td><td></td></tr>
<tr><td>4.</td><td>Key in the HP No. (phone number) field</td><td>Invalid HP No. value: 012-345 6789
(Phone number should not have dashes and spaces)</td><td>The input box should turn red and display the error message of “Invalid phone number value” at the bottom of input box.</td><td></td><td></td></tr>
</table>

5.	Click on Return button		Return button should be disabled, and clicking on it should not have effects.		

3.1.5 Add New Book Function

<table>
<tr><td colspan="6">Program Name: Add New Book Function
Description: To test the add new book function with valid book details.</td></tr>
<tr><td colspan="3">Test Date: 22/4/2024</td><td colspan="3">Tester: MEGAN YEOH TZE XUAN</td></tr>
<tr><th>No.</th><th>Test Cases</th><th>Test Data</th><th>Expected Result</th><th>Actual Result</th><th>Remarks/Comments</th></tr>
<tr><td>1.</td><td>Navigate to the add new book page</td><td></td><td>The system should display the ‘add new book’ page after navigation.</td><td></td><td></td></tr>
<tr><td>2.</td><td>Key in ISBN of book to add</td><td>Valid ISBN value:
9780192728029</td><td></td><td></td><td></td></tr>
<tr><td>3.</td><td>Key in Title of book to add</td><td>Valid Title value:
The Wizard of Oz</td><td></td><td></td><td></td></tr>
<tr><td>4.</td><td>Key in Author Name of book to add</td><td>Valid Author value:
L. Frank Baum</td><td></td><td></td><td></td></tr>
<tr><td>5.</td><td>Upload Image of Book’s Cover</td><td>Valid Image File:
wizardofoz.jpg</td><td>The system should display the image uploaded in the box after successful upload.</td><td></td><td></td></tr>
<tr><td>6.</td><td>Key in Publisher of the book to add</td><td>Valid Publisher value:
George M. Hill Company</td><td></td><td></td><td></td></tr>
</table>

7.	Key in Book Description of the book to add	Valid Book Description value: Dorothy ends up in the magical Land of Oz after she and her pet dog Toto are swept away from their home by a cyclone.			
8.	Select Genre of the book to add	Valid genre value: Fantasy			
9.	Select Book Category of the book to add	Valid category value: Physical Book			
10.	Select Publishing Date of the book to add	Valid			
11.	Select Language of the book to add	-	The system should be able to display a dropdown list of languages for the user to select from and display the selected language afterwards.		
12.	Key in Book Edition of the book to add	Valid Book Edition value: First Edition	The system should allow the user to continue on keying in the next input field.		

13.	Key in Number of Pages of the book to add	Valid Number of Pages value: 158	The system should validate if the entered value is in numbers before allowing the user to continue on keying in the next field.		
14.	Key in Location of the book to add	Valid Location value: Shelf C Row 3	The system should allow the user to continue on keying in the next input field.		
15.	Select if the book to add is Borrowable or not	-	The system should display and let the user select one radio button on whether the book’s borrowable status is ‘Yes’ or ‘No’.		
16.	Key in Supplier of the book to add	Valid Supplier value: TanMark Book Centre Sdn. Bhd	The system should allow the user to continue on keying in the next input field.		
17.	Key in the Price of the book to add	Valid Price value: 70.00	The system should validate if the entered values are numbers before allowing the user to continue on keying in the next input field.		

18.	Select the Date when the book is added to the library	-	The system should be able to display a calendar for the user to select the date when the book is added to the library.		
19.	Click on Add button	-	The system should display a “Book Successfully Added” message and redirect the user to the page to display the added book’s details.		

Program Name: Add New Book Function Description: To test the add new book function with invalid book details.					
Test Date: 22/4/2024			Tester: MEGAN YEOH TZE XUAN		
No.	**Test Cases**	**Test Data**	**Expected Result**	**Actual Result**	**Remarks/Comments**
1.	Navigate to the add new book page	-	The system should display the ‘add new book’ page after navigation.	The ‘add new book’ page is displayed after navigation.	Test Status: PASS
2.	Key in ISBN of book to add	Invalid ISBN value: Abcd (ISBN should be in ISBN-13 format and should not have any alphabets)	The input box will turn red and an error message of “Please fill in ISBN” will be displayed.	The system does not show any error message and allows the user to proceed onto the next input field.	Test Status: FAIL
3.	Enter empty value to Title field	-	The input box will turn red and an error message of “Please fill in Book Title” will be displayed.	The system does not show any error message and allows the user to proceed onto the next input field.	Test Status: FAIL
4.	Enter empty value to Author field	-	The input box will turn red and an error message of “Please fill in Author Name” will be displayed.	The system does not show any error message and allows the user to proceed onto the next input	Test Status: FAIL

				field.	
5.	Upload Image of Book’s Cover	Invalid Image File Format: wizardofoz.mp4 (Image file format should be only in JPG, PNG, BMP, SVG, DXF formats)	The system will show an error message of “Invalid image format uploaded”.	The system shows the error message of “Invalid image format uploaded”.	Test Status: PASS
6.	Key in Publisher of the book to add	-	The system should disable the Publisher field due to invalid input from previous fields.	The system does not disable the Publisher field and the user can still enter values into the input box.	Test Status: FAIL
7.	Key in Book Description of the book to add	-	The system should disable the Book Description field due to invalid input from previous fields.	The system does not disable the Book Description field and the user can still enter values into the input box.	Test Status: FAIL
8.	Select Genre of the book to add	-	The system should disable the Genre dropdown and the user would not be able to make any selections.	The drop-down selection is shown active and can be used to select the book's genre.	Test Status: FAIL
9.	Select Book Category of the book to add	-	The system should disable the Book Category dropdown	The drop-down selection is disabled and cannot be	Test Status: PASS

			and the user would not be able to make any selections.	accessed.	
10.	Select Publishing Date of the book to add	-	The system should disable this field by not displaying the calendar for the user to select the Publishing Date.	The system does not disable the Publishing Date field and the user can still select the Publishing Date of the book.	Test Status: FAIL
11.	Select Language of the book to add	-	The system should disable the Language dropdown and the user would not be able to make any selections.	The system disables the Language dropdown and the user cannot make any selections.	Test Status: PASS
12.	Key in Book Edition of the book to add	-	The system should disable the Book Edition field due to invalid input from previous fields.	The system disables the Book Edition field and the user cannot enter values into the input box.	Test Status: PASS
13.	Key in Number of Pages of the book to add	Invalid Number of Pages value: abc (Number of Pages entered should only be in integers)	The input box will turn red and an error message of “Invalid Number of Pages entered” will be displayed.	The system does not display the error message and allows the user to proceed on to the next input field.	Test Status: FAIL
14.	Key in Location of the book to add	-	The system should disable the Book	The system disables the Book Location	Test Status: PASS

			Location field due to invalid input from previous fields.	field and the user cannot enter values into the input box.	
15.	Radio Button on book Borrowable status is not selected	-	The system will display an error message of “Please select the book’s borrowable status.”	The system does not show any error message and allows the user to proceed onto the next input field.	Test Status: FAIL
16.	Key in Supplier of the book to add	-	The system should disable the Supplier Name field due to invalid input from previous fields.	The system disables the Supplier Name field and the user cannot enter values into the input box.	Test Status: PASS
17.	Key in the Price of the book to add	Invalid Price value: abc (Price entered should only be in numbers)	The input box will turn red and an error message of “Invalid Price entered” will be displayed.	The system does not display the error message and allows the user to proceed on to the next input field.	Test Status: FAIL
18.	Select the Date when the book is added to the library	-	The system should disable this field by not displaying the calendar for the user to select the Date Added to Library.	The system disables the Date Added to Library field and the user cannot select the date when the book is added to the library.	Test Status: PASS
19.	Click on Add button	-	The system would be disabled due to	The system proceeds to submit the form	Test Status: FAIL

			invalid inputs and the user should not be able to submit the form.	despite of invalid input.	

3.1.6 Request New Book Function

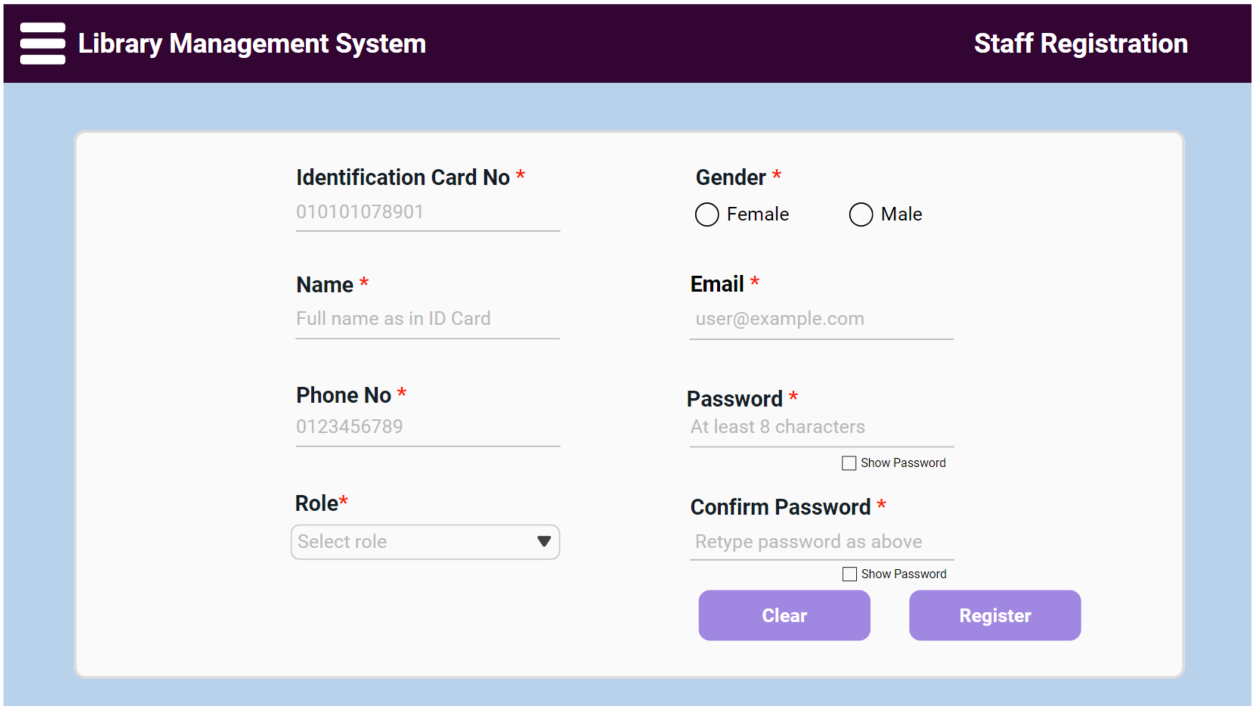
<table>
<tr><td colspan="6">Program Name: Request New Book Function
Description: To test the request new book function with valid request new book data.</td></tr>
<tr><td colspan="3">Test Date: 22/4/2024</td><td colspan="3">Tester: CALVEN PHNUAH KAH HONG</td></tr>
<tr><th>No.</th><th>Test Cases</th><th>Test Data</th><th>Expected Result</th><th>Actual Result</th><th>Remarks/Comments</th></tr>
<tr><td>1.</td><td>Navigate to Request New Book Page</td><td>-</td><td>The system should display the request new book page.</td><td>The new book page is displayed.</td><td>Test Status: PASS</td></tr>
<tr><td>2.</td><td>Key in Book Title</td><td>Valid Book Title: Absalom,Absalom</td><td>The input field should turn green.</td><td>The input field is turn green.</td><td>Test Status: PASS</td></tr>
<tr><td>3.</td><td>Key in Author</td><td>Valid Author: William Faulkne</td><td>The input field shouldturn green.</td><td>The input field is turn green.</td><td>Test Status: PASS</td></tr>
<tr><td>4.</td><td>Click on request button</td><td>-</td><td>The system should update the status into “Requested” and display the data into history column.</td><td>The system is able to update the status into “Requested” and display the data into history column.</td><td>Test Status: PASS</td></tr>
</table>

<table>
<tr><td colspan="6">Program Name: Request New Book Function
Description: To test the request new book function with invalid request new book data.</td></tr>
<tr><td colspan="3">Test Date: 22/4/2024</td><td colspan="3">Tester: CALVEN PHNUAH KAH HONG</td></tr>
<tr><th>No.</th><th>Test Cases</th><th>Test Data</th><th>Expected Result</th><th>Actual Result</th><th>Remarks/Comments</th></tr>
<tr><td>1.</td><td>Navigate to Request New Book Page</td><td>-</td><td>The system should display the request new book page</td><td>The system is able to display the request new book page.</td><td>Test Status: PASS</td></tr>
<tr><td>2.</td><td>Key in Book Title</td><td>Invalid Book Title: Absalom,Absalom33</td><td>The input field should turn red and prompt an error message “Please enter book title without digit”</td><td>The input field is turn red and error message is displayed.</td><td>Test Status: PASS</td></tr>
<tr><td>3.</td><td>Key in Author</td><td>Invalid Author: William Faulkne123</td><td>The input field should turn red and prompt an error message “Please enter author name with character only.”</td><td>The input field does not turn red and error message does not display.</td><td>Test Status: FAIL</td></tr>
<tr><td>4.</td><td>Click on request button</td><td>-</td><td>The system should be to update the status into “Requested” and display the data into history column.</td><td>The system doesn't update the status into “Requested” and display the data into history column.</td><td>Test Status: FAIL</td></tr>
</table>

3.2 User Interface Design

3.2.1 Ease Of Data Entry

The ease of data is important across various aspects of a user interface. The staff register module is designed to prioritise ease of data entry so that it benefits the user and increases data accuracy.



Identification Card No *
010101078901

Gender *
○ Female ○ Male

Name *
Full name as in ID Card

Email *
user@example.com

Phone No *
0123456789

Password *
At least 8 characters
□ Show Password

Role*
Select role ▼

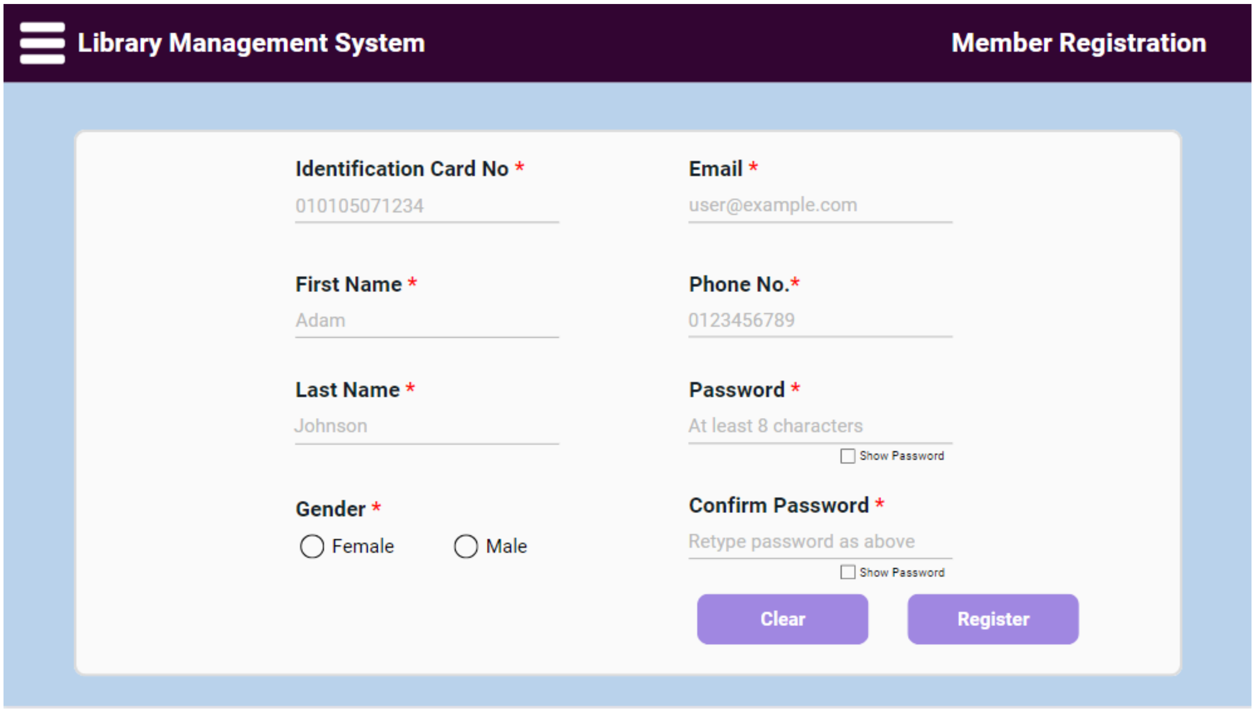
Confirm Password *
Retype password as above
□ Show Password

Figure 3.2.1 Library System Book Staff Registration Page

The input field of the identification card number, name, phone number, email and password can be typed by the user. Each input field has been labelled with clear instructions about the expected format to minimise invalid input errors. The role field can be chosen by the given dropdown button that lists the selectable roles for the staff. The gender field which has limited choice can be chosen with the radio button. This type of button is also used to ensure data entered is complete and accurate, and aligned with the system's expectations, thereby enhancing the overall user experience.

3.2.2 Validation Check

The user interface was designed with the validation check. The validation check is performed when the “Register” button is clicked. Null value validation check is one of the validation checks in user interface design.



Identification Card No *
010105071234

Email *
user@example.com

First Name *
Adam

Phone No.*
0123456789

Last Name *
Johnson

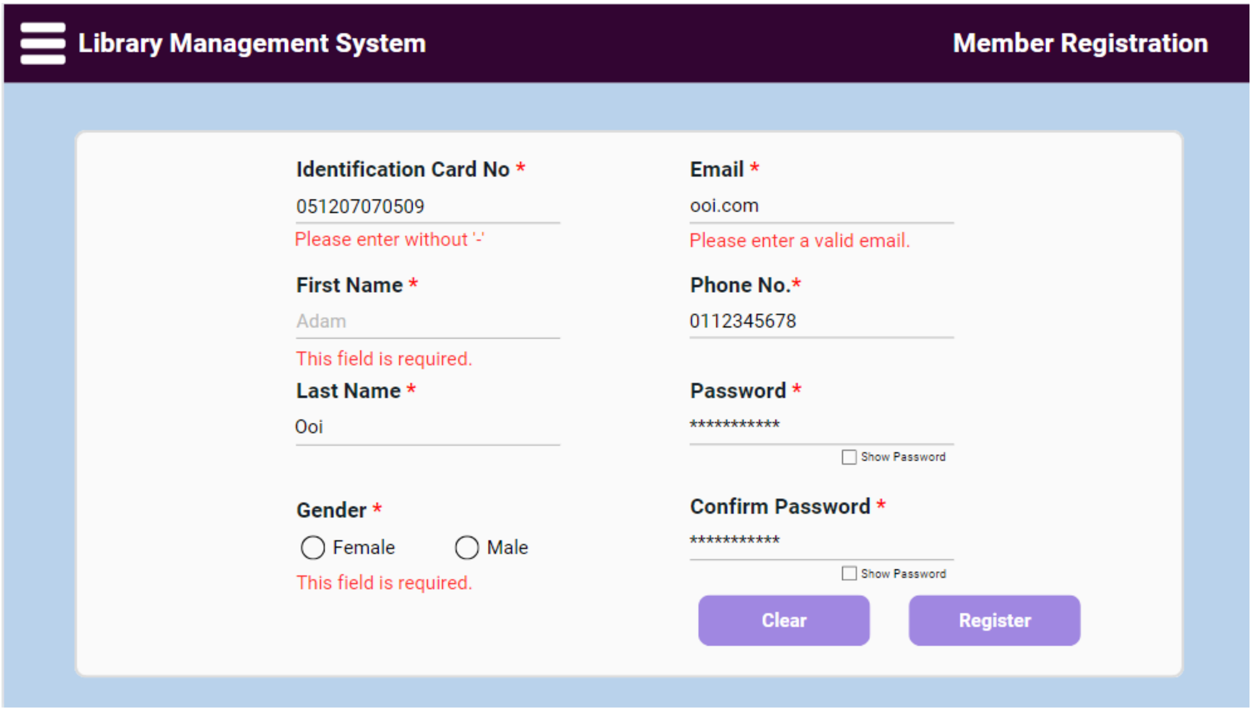
Password *
At least 8 characters
□ Show Password

Gender *
○ Female ○ Male

Confirm Password *
Retype password as above
□ Show Password

Figure 3.2.2 Library System Book Member Registration Page

The field of identification card numbers, names, genders, phone numbers, and passwords are verified to confirm to be filled. The system will ensure the required fields are not left empty or null before submission. The mandatory fields are indicated with a red asterisk. If the user misses the field, an error message will be prompted to remind the user about it and prevent the form submission.



Identification Card No *
051207070509
Please enter without '-'

Email *
ooi.com
Please enter a valid email.

First Name *
Adam
This field is required.

Phone No.*
0112345678

Last Name *
Ooi

Password *
****
□ Show Password

Gender *
○ Female ○ Male
This field is required.

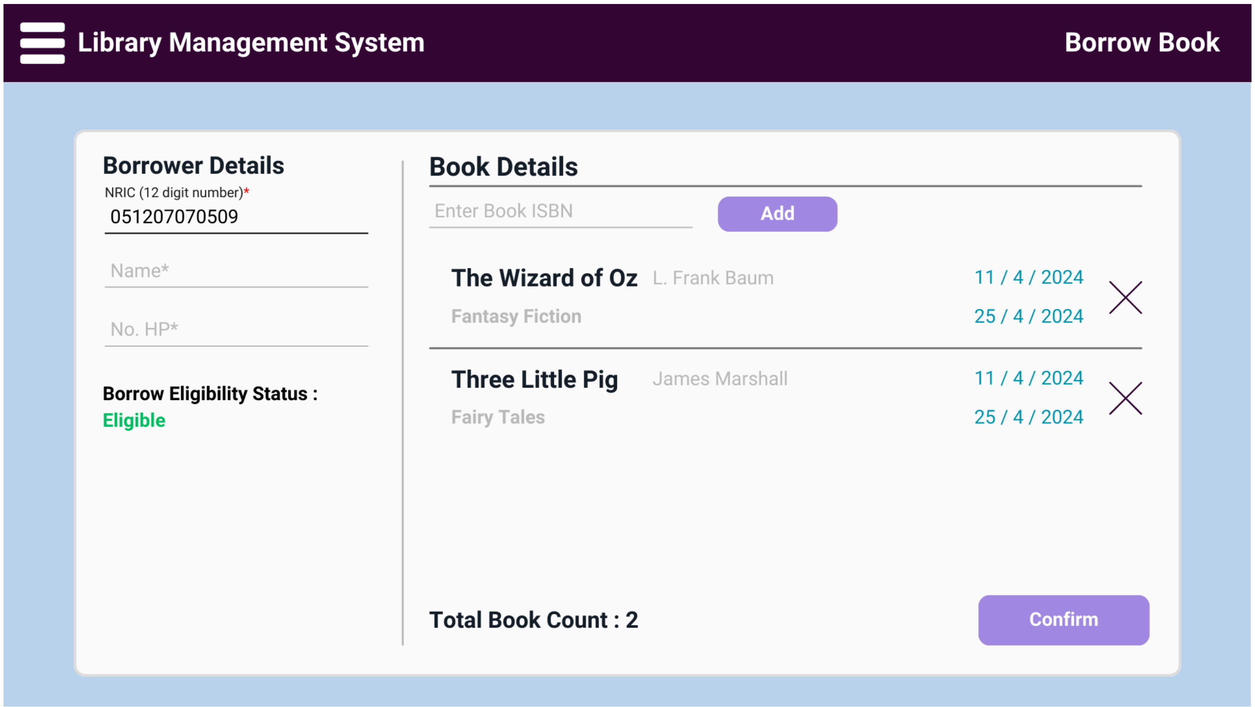
Confirm Password *
****
□ Show Password

Figure 3.2.3 Library System Book Member Registration Page with Validation Error Message

Moreover, users are required to fill in the information following the format required. The format validation check ensures the data entered into specific fields adhere to predetermined formats and criteria. Format validation ensures the data consistency and accuracy. The guidance for entering the correct format of fields is shown. For example, users should enter 12-digit numbers into the ID card field, instead of including dashes within the value. This same goes for the validation check of the phone number. In addition, the validation check of name input is carried out by verifying the inexistence of digit value. Lastly, the password is validated by ensuring a minimum of 8 characters in the password input. After entering the password, users are required to confirm the password by entering the password again. The validation check of both passwords and confirm passwords must be the same to pass the verification. If the users enter the wrong format of information, a validation error message will be displayed to alert the user.

3.2.3 Consistency

Consistency is one of the important principles when considering the user interface design. This principle ensures all the interfaces share identical elements and looks, increasing the application interface's cohesiveness. As a result, the users will not feel a sense of dissonance when navigating from page to page within the application, as well as increase the learnability of the application.



Borrower Details

NRIC (12 digit number)*
051207070509

Name*

No. HP*

Borrow Eligibility Status :
Eligible

Book Details

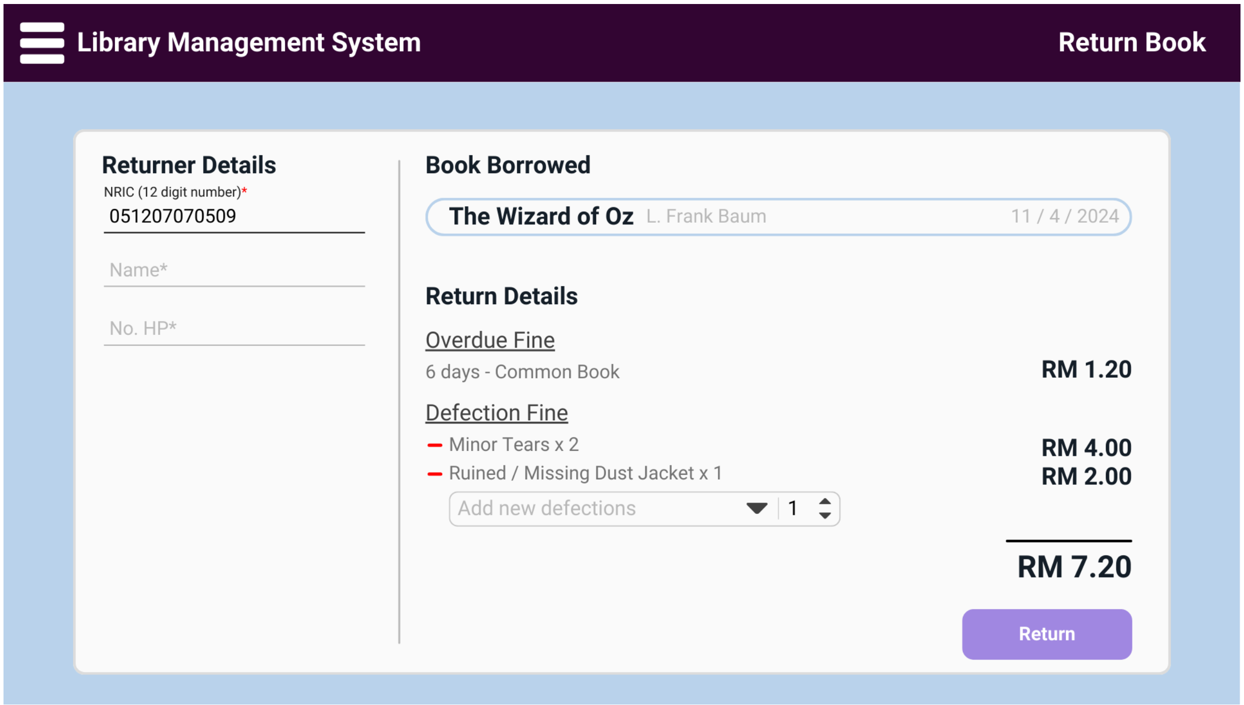
Enter Book ISBN

The Wizard of Oz L. Frank Baum 11 / 4 / 2024
Fantasy Fiction 25 / 4 / 2024

Three Little Pig James Marshall 11 / 4 / 2024
Fairy Tales 25 / 4 / 2024

Total Book Count : 2

Figure 3.2.4 Library System Book Borrowing Page



Returner Details

NRIC (12 digit number)*: 051207070509

Name*

No. HP*

Book Borrowed

The Wizard of Oz L. Frank Baum 11 / 4 / 2024

Return Details

Overdue Fine

6 days - Common Book RM 1.20

Defection Fine

– Minor Tears x 2 RM 4.00

– Ruined / Missing Dust Jacket x 1 RM 2.00

Add new defections 1

RM 7.20

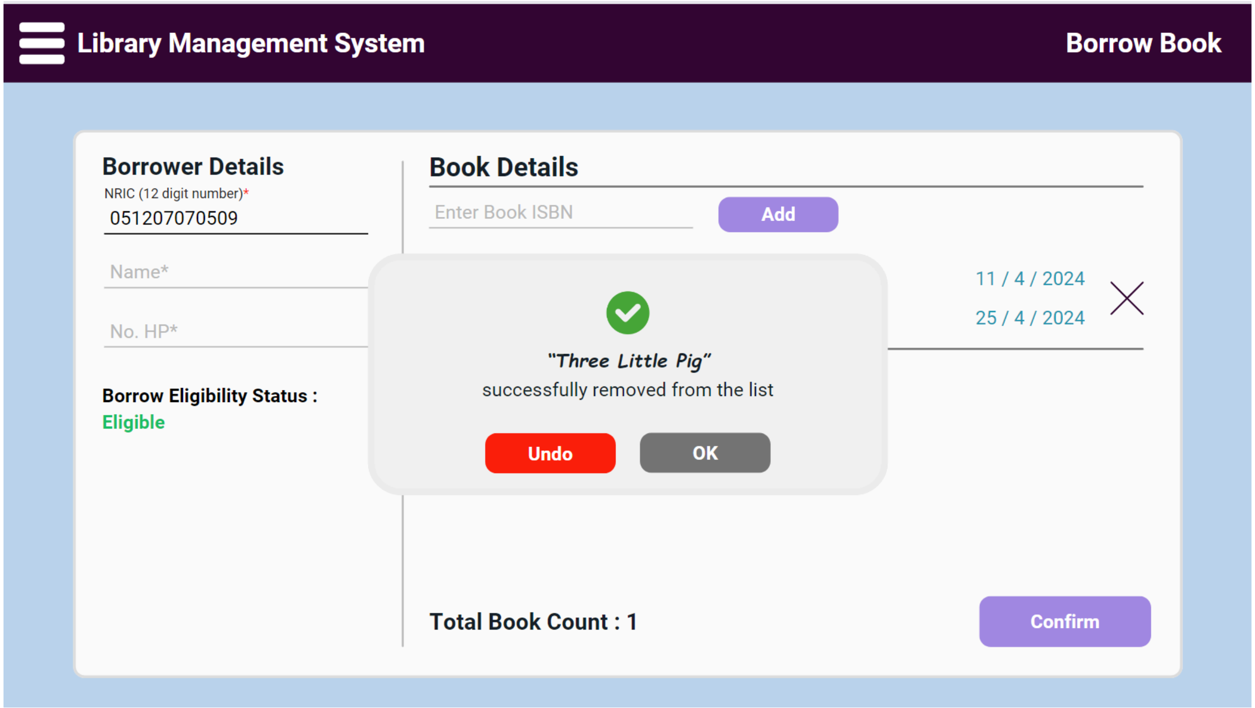
Figure 3.2.5 Library System Book Returning Page

An example of applying this principle within this library system can be identified from the book borrowing and returning page. Both of the pages have an identical layout, whereby the side panel is used to read and display the borrower or returner details, and the main panel shows the book transaction information.

Besides the general looks, the process is also very similar. Both processes are started by entering the borrowers’ or returners’ NRIC numbers. The system will display their name and phone number if their information exists within the system, otherwise, the librarian needs to enter them manually. Other than that, the button to commit the transaction which is placed at the bottom right corner of the page will be clicked to end the transaction. The similarity in the looks and processes will shorten the librarians’ time to learn and operate the system.

3.2.4 Recoverability

Recoverability is a fundamental concept in user interface design primarily referring to a system's capacity to smoothly guide users through error rectification or reversing unintended actions. By providing users with ways to recover from mistakes, recoverability helps foster a sense of control and confidence in the system. Furthermore, it will be able to greatly reduce user frustration when using the system and increase the efficiency of workflow by allowing users to swiftly recover from errors.



NRIC (12 digit number)*
051207070509

Borrow Eligibility Status :
Eligible

11 / 4 / 2024
25 / 4 / 2024

"Three Little Pig"
successfully removed from the list

Total Book Count : 1

Figure 3.2.6 Library System Book Borrowing Page with Alert Message

The above diagram shows an example of the integration of the recoverability concept within the library's borrow page. If users mistakenly input the wrong book ISBN or scan an incorrect barcode, they can easily rectify this by utilising the remove button positioned alongside each book entry. Upon pressing the remove button, the system eliminates the selected book from the borrow list. Additionally, a pop-up message notifies users of the successful removal. If the users want to undo the action, the 'Undo' button is provided for the amendment. This approach not only provides users with greater control over their interactions but also promotes confidence in the system's usability.

3.2.5 Meaningful Error Message

Whenever an error occurs while a user uses the system, an error message is shown to users to inform them what's gone wrong. This error message should be meaningful and concise with its information to clarify the cause of errors for users. It can be done by using clear and simple language without including any technical jargon. Error messages should also offer solutions for users to try and resolve the problem. With error messages that can pinpoint the exact issues encountered, users can have a clear guide on solving problems faced with the system.

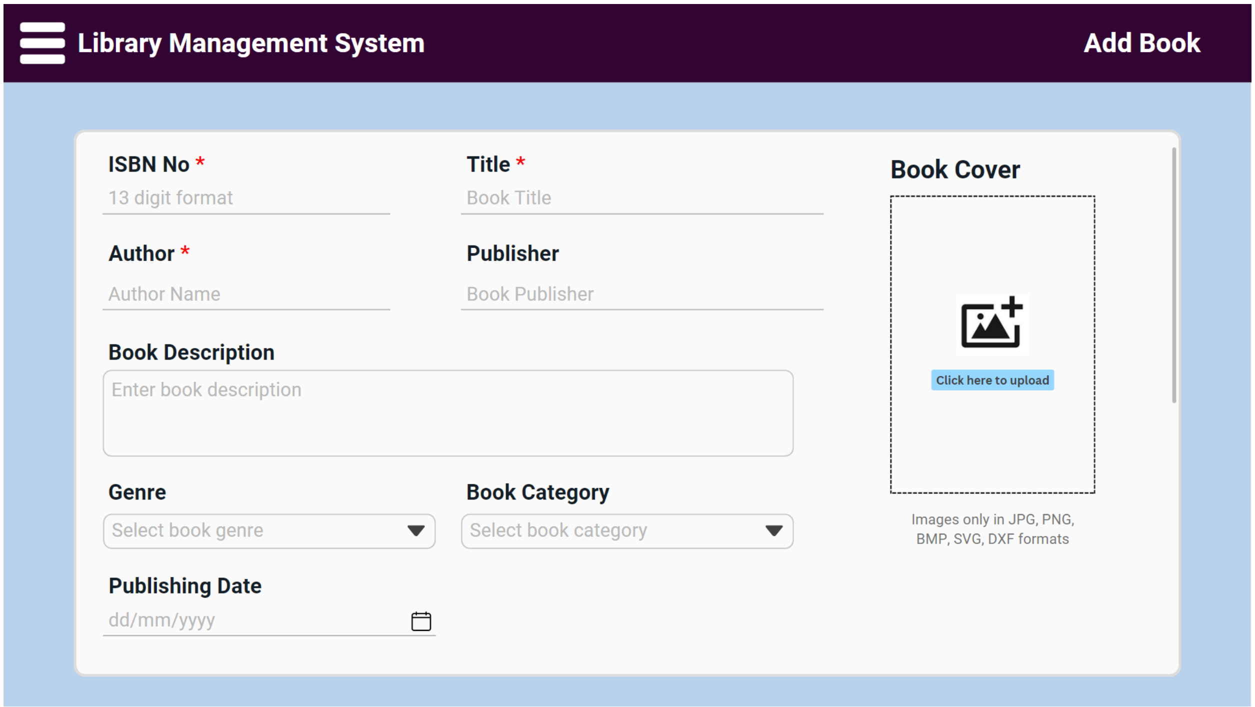
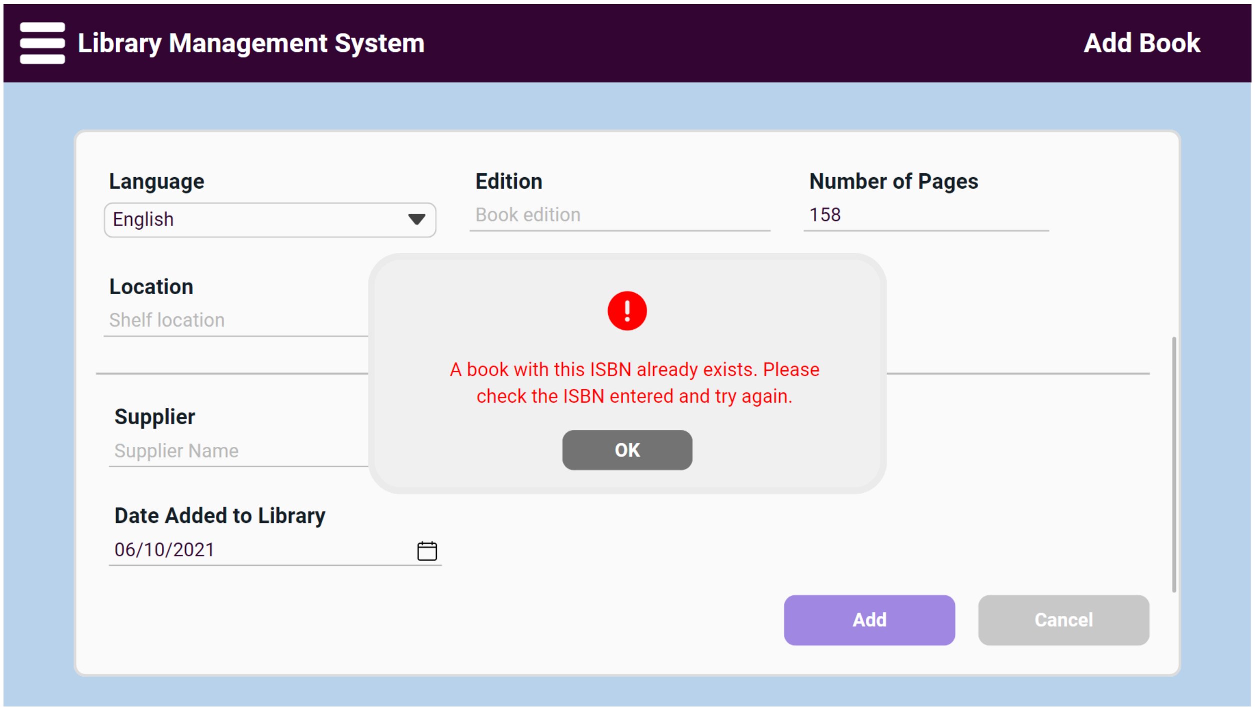


Figure 3.2.7 Library System Add Book Page

The image above shows the interface for library staff to add new book records. It is a form with multiple fields to store relevant information for each book in the library.



Language: English

Edition: Book edition

Number of Pages: 158

Location: Shelf location

Supplier: Supplier Name

Date Added to Library: 06/10/2021

A book with this ISBN already exists. Please check the ISBN entered and try again.

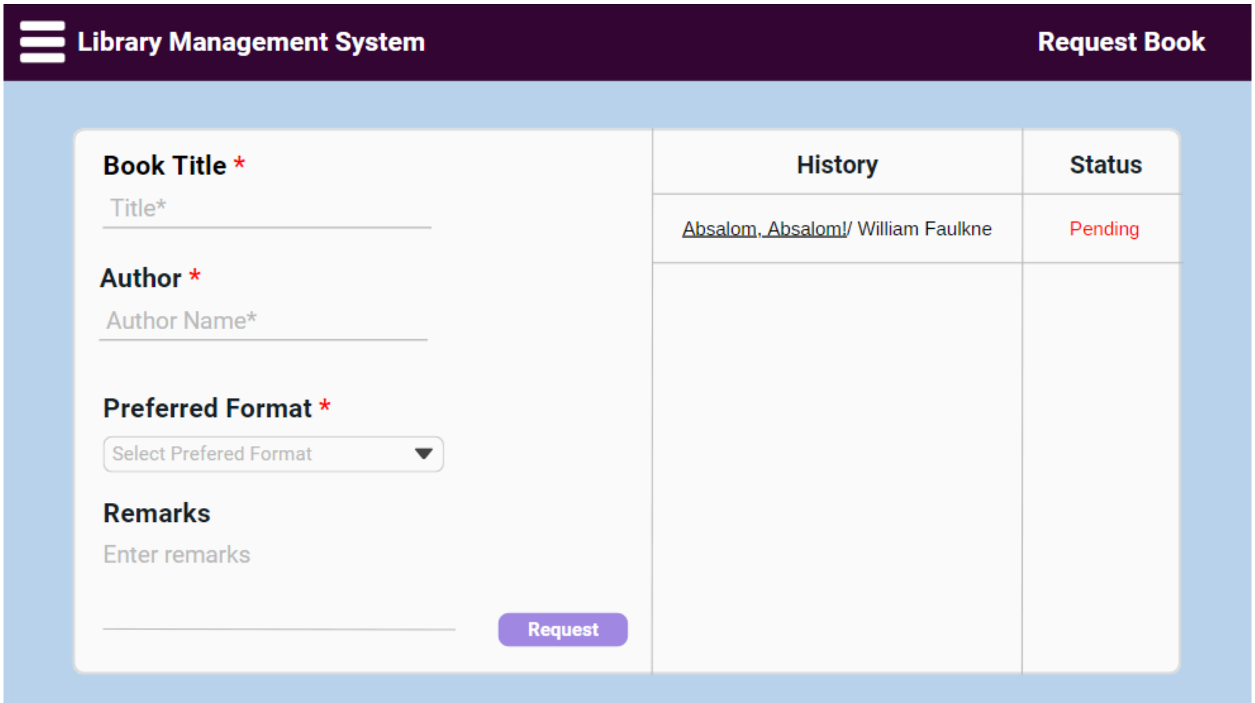
Figure 3.2.8 Library System Add Book Page with Error Message

Before submitting the form, the system would check if the ISBN entered by the user already exists in the library database. This is a crucial step to prevent duplicate data where one book is added twice into the database. If the ISBN entered already exists, the popup message would be shown as in the image to inform the user about the error they have encountered, alerting them that the ISBN they entered already exists. Therefore, they should check again to ensure a correct ISBN is entered. With the meaningful error message as displayed above implemented in the library system, it would allow users to identify and solve errors immediately.

3.2.6 User Diversity

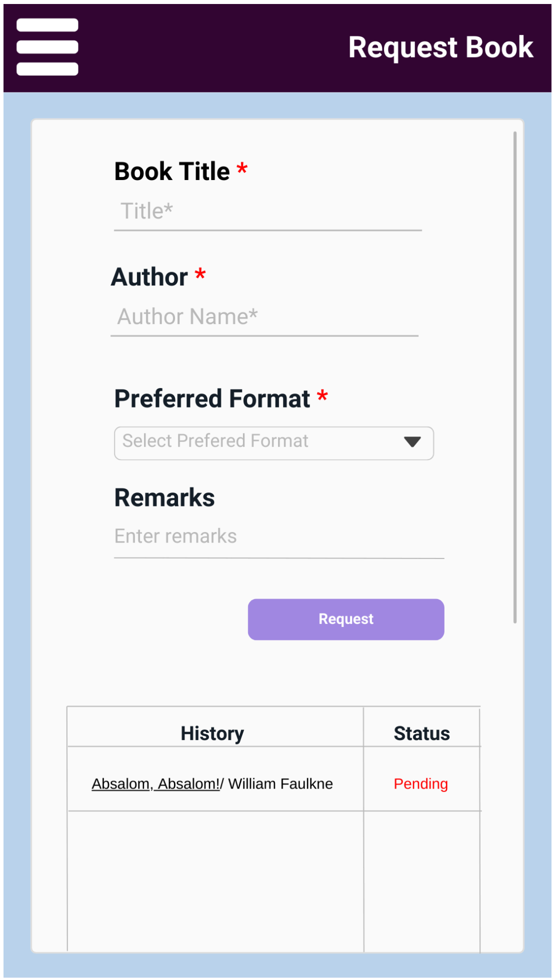
User diversity in a system refers to having a different range of users with different backgrounds and perspectives. It encompasses a range of needs, objectives and capabilities and aims to incorporate as many individuals as possible. Different users come with various browser environments, therefore, the system should be accessible by each user despite having different platforms to access the library system. By incorporating the user diversity principle, it ensures that the library system is accessible to a wide range of users, regardless of what platform they use.

Using the request book page as an example, since the page is accessed by all system members, the users are expected to view the page in various screen dimensions, such as 16:9 dimension on PC or 9:16 dimension on the mobile browser. Therefore, the interface is implemented with a responsive design to ensure the web page can consistently convey the information to every user.



History	Status
Absalom, Absalom!/ William Faulkne	Pending

Figure 3.2.9 Library System Request Book Page - PC dimension



Book Title *

Title*

Author *

Author Name*

Preferred Format *

Select Prefered Format

Remarks

Enter remarks

History	Status
Absalom, Absalom!/ William Faulkne	Pending

Figure 3.2.10 Library System RequestBook Page - Mobile dimension

3.3 System Maintenance

System maintenance is a process to sustain a software system's to ensure it is functioning smoothly and continuously support the business needs. In software engineering, there is different types of system maintenance whereas each of them plays an important role in contributing to extended system lifespan. The 4 types of system maintenance that are applicable to the library systems are adaptive maintenance, perfective maintenance, corrective maintenance and preventive maintenance.

3.3.1 Adaptive Maintenance

Adaptive maintenance focuses on making modifications to a software system to adapt to changes in the system environment and external regulation. Adaptive maintenance is crucial to adapt a software system to changes to ensure its continued functionality and effectiveness.

One of the aspects that is being considered in adaptive maintenance is the system environment compatibility. An operating system is the software that supports the communication of the library system and the infrastructure. Most of the time, the operating system will constantly receive updates to improve its invulnerability over time. To protect the system's server from attacks by hackers, we should update the operating system that runs on the server infrastructure and client devices at a constant rate. At the same time, we need to ensure that the library system can adapt to the changes in the operating system and resolve any conflicts in the system specifications to avoid the system's failure. However, as adaptive maintenance also requires resources and time to be allocated, the maintenance should be carried out wisely. The effective strategy is the software team should only be concerned about the critical security updates, instead of minor performance and feature updates which do not make significant changes to the library system which runs on it. This could decrease the rate of adaptive maintenance and save resources for better use.

Furthermore, changes in laws, regulations or industry standards may require modifications to the software system to ensure compliance. It involves updating the system to adhere to new legal or regulatory requirements. For example, new tax regulations such as Sales and Service Tax (SST) by the government are required to be implemented into the transaction module to ensure that the transaction amount are being calculated correctly.

3.3.2 Perfective Maintenance

Perfective maintenance is one of the four types of software maintenance. This type of maintenance aims to enhance the functionality, performance, and user-friendliness of the software over time. Perfective software maintenance focuses on the evolution of requirements and features that exist in our library system. As users interact with our applications, they may notice details previously overlooked or suggest new features that they would like as part of the software, which could contribute to future enhancements. Perfective software maintenance takes over some of the work, both adding features that can enhance user experience and removing features that are not effective and functional. This can include features that are not used or those that do not help you to meet your end goals.

The goal of perfective maintenance in a library system is to ensure the software remains relevant, useful, and competitive in the marketplace. By continuously refining and enhancing the software's features, performance, and usability, the library system is able to meet changing user demands and technological advancements. This approach will be able to extend the software's lifespan and maximise its value for the organisation.

Examples of perfective maintenance in a library system include implementing internationalisation features to support multiple languages. By offering content in multiple languages, libraries can reach a broader audience which improves accessibility to resources and services. Another example is integrating recommendation algorithms for personalised book suggestions, which helps to pique interest of customers, thereby promoting more frequent use of library resources. These perfective maintenance not only improve the software's functionality but also contribute to enhancing overall user experience and maintaining the library system's competitiveness in a dynamic environment.

3.3.3 Corrective Maintenance

The corrective maintenance encompasses the procedures undertaken to identify and rectify faults or failures in equipment or systems after they have occurred, with the primary objective of restoring them to their optimal operational state. This form of maintenance can be further categorised into planned and unplanned corrective maintenance based on the scheduling and execution of maintenance activities.

In contrast, unplanned corrective maintenance often referred to as emergency or reactive maintenance, involves responding immediately to unforeseen equipment failures or breakdowns that occur suddenly and unexpectedly. The primary aim of unplanned corrective maintenance is to address issues to prevent further damage or operational disruptions. In addition, if the issue escalated into more severe failures, it could lead to extended downtime or more costly repairs.

Some of the main activities are required to be carried out to conduct complete corrective maintenance regardless of the approaches mentioned above. The first one is gathering issues identified by the development team or by users. The problems will be analysed and prioritised to allocate proper resources and tasks for maintenance implementation. After the maintenance is implemented, testing will be conducted to ensure the bugs are fixed and the original functionalities are still working as expected. Lastly, the patch will be verified and deployed to the existing system. If the maintenance is unplanned, usually the patch is required to be applied as quickly as possible. Otherwise, the patch will be deployed in a more organised manner and distributed in fixed intervals.

3.3.4 Preventive Maintenance

Preventive maintenance is a type of maintenance that minimises the happening of potential errors when a system is being run in a real-life environment. As opposed to corrective maintenance, this maintenance aims to decrease the risk of unplanned downtime due to system failures, which could avoid inducing a large amount of cost towards the organisation.

This task is carried out through scheduled inspection and maintenance. The tasks to be performed during the maintenance are often listed in a checklist. Each of these tasks is required to be categorised by its mandatoriness. Mandatory maintenance tasks are required to be carried out as soon as they are due, and it is commonly tasks that ensure safety-critical aspects of the system. On the other hand, non-mandatory tasks are those which can be delayed without resulting in a critical system failure or security issue. Inspection will be carried out during preventive maintenance to monitor for indications of potential faults. After faults are identified from the inspection, the maintenance team can determine for execution of just-in-time maintenance during the inspection for minor fixes, or documentation of inspection results for further maintenance use.

As for scheduling the maintenance tasks above, the most commonly used methods to conduct the maintenance are calendar-based maintenance and usage-based maintenance. Calendar-based approach schedules the maintenance using calendar intervals, such as weekly, monthly or annually, whereas usage-based maintenance schedules the maintenance depending on the daily utilisation of the system itself. As usage-based maintenance is a more reactive approach, it is only considerable for non-critical systems to prevent unexpected failure. On the contrary, calendar-based maintenance fits better in the situation for critical systems where the failure could lead to safety concerns or significant loss. In this particular case, the library system could adopt usage-based maintenance as it is more cost-effective.

Preventive maintenance is a crucial task to be performed to ensure system quality in the long run. It helps the software team to anticipate potential faults within the system and allows proactive maintenance to be made as opposed to reactive maintenance. This not only helps to boost cost efficiency but, at the same time increases the system's reliability which contributes to better client satisfaction.

References

1. AltexSoft. (n.d.). Acceptance Criteria: Purposes, Types, Examples and Best Prac. [online] Available at: https://www.altexsoft.com/blog/acceptance-criteria-purposes-formats-and-best-practices/.
2. atlan.com. (n.d.). Data Validation: Types, Benefits, and Accuracy Process. [online] Available at: https://atlan.com/what-is-data-validation/.
3. Default. (n.d.). The 4 Types Of Software Maintenance & How They Help | CAST Software. [online] Available at: https://www.castsoftware.com/glossary/Four-Types-Of-Software-Maintenance-How-They-Help-Your-Organization-Preventive-Perfective-Adaptive-corrective.
4. Eurotechconseil. (n.d.). What is perfective maintenance ? [online] Available at: https://www.eurotechconseil.com/en/glossary/perfective-maintenance/.
5. GeeksforGeeks. (2018). Software Engineering | Software Maintenance - GeeksforGeeks. [online] Available at: https://www.geeksforgeeks.org/software-engineering-software-maintenance/.
6. GeeksforGeeks. (2020). Component Based Software Engineering. [online] Available at: https://www.geeksforgeeks.org/component-based-software-engineering/?ref=gcse.
7. Henson, M. (2018). 17 Essential Steps In The Software Upgrade Process. [online] eLearning Industry. Available at: https://elearningindustry.com/software-upgrade-process-essential-steps.
8. IBM (2023). What is preventive maintenance? | IBM. [online] www.ibm.com. Available at: https://www.ibm.com/topics/what-is-preventive-maintenance.
9. kissflow (2022). Rapid Application Development (RAD) | Definition, Steps & Full Guide. [online] kissflow.com. Available at: https://kissflow.com/application-development/rad/rapid-application-development/.
10. kissflow.com. (n.d.). RAD (Rapid Application Development): Definition, Meaning and Benefits. [online] Available at: https://kissflow.com/application-development/rad/rapid-application-development-rad-meaning/.

11. Limble CMMS. (n.d.). What is Corrective Maintenance? [online] Available at: https://limblecmms.com/maintenance-definitions/corrective-maintenance/.
12. OpenGenus IQ: Computing Expertise & Legacy. (2022). *System Design of Library Management System*. [online] Available at: https://iq.opengenus.org/system-design-of-library-management-system/.
13. Pruciak, M. (n.d.). *Key Principles of Rapid Application Development (RAD) and How to Apply Them to Your Projects*. [online] www.ideamotive.co. Available at: https://www.ideamotive.co/blog/key-principles-of-rapid-application-development.
14. PubNub. (n.d.). What is Load Balancing? [online] Available at: https://www.pubnub.com/guides/load-balancing/.
15. Team, T. (2023). Bad Reasons to Update Your Linux Kernel. [online] TuxCare. Available at: https://tuxcare.com/blog/5-bad-reasons-to-update-your-linux-kernel/.
16. Testsigma Blog. (2023). What Are The Software Quality Attributes? [online] Available at: https://testsigma.com/blog/software-quality-attributes/#Portability.
17. Veritas (2023). Data Backup & Recovery: The Essential Guide for Businesses | Veritas. [online] www.veritas.com. Available at: https://www.veritas.com/information-center/data-backup-and-recovery.
18. www.twi-global.com. (n.d.). What is Preventive Maintenance? [online] Available at: https://www.twi-global.com/technical-knowledge/faqs/what-is-preventive-maintenance.