

Design And Build Web-Base Application Library Management System

Li Tengfei

*School of Information Technology
Unitar International University
Petaling Jaya, Kuala Lumpur
fly020880@gmail.com*

Zhang Jie

*School of Information Technology
Unitar International University
Petaling Jaya, Kuala Lumpur
adat13803@gmail.com*

Hasanul Fahmi

*Faculty of Computer Science
President University
Bekasi, West Java
h.fahmizuhri@gmail.com*

Abstract— At present, most of China's higher education institutions have completed the computerization of library management, and library management is more scientific and efficient. However, many university library management systems have some problems, such as outdated functions, aging systems and unstable operation. In this paper, the authors designed a library management system. Under the Javaweb environment, the whole system is built using SpringMVC framework, MySQL database as the data storage support of the system, Windows 10 as the operating system, and Tomcat 8.0 as the web server. With the help of this technology stack, the problem of unstable operation of the previous system is solved, and a more efficient and stable library management information system is constructed, which not only reduces the workload of librarians, but also provides students with a more convenient book borrowing and lending experience, so that the library can play the most useful role in the campus life.

Keywords—*Library management system, SpringMVC, JavaWeb, Tomcat 8.0, User experience, System stability*

I. INTRODUCTION

With the arrival of the digital era, people's demand for information is increasing. As the center of information dissemination and the place of knowledge management, libraries bear very important social responsibilities. However, the traditional manual management method has been difficult to meet the needs of modern library management, which requires the use of computer technology to automate and intelligentize library management and improve the efficiency and quality of library management. Therefore, the development of a library management system has high practical significance and practical value. The system can not only facilitate the library managers to maintain the information of books and efficiently process the retrieval, borrowing and returning of books, but also provide readers with more convenient and fast services, and at the same time, it can also realize the enhancement of the utilization rate of the library's resources and management efficiency.

II. LITERATURE REVIEW

Based on SpringMVC framework, this paper proposes a new design scheme for library management information system, and through the review of related literature, it discusses in depth the current challenges, technological

advances, and future development direction of university library management system.

A. Current Challenges

Many university library management systems (LMS) suffer from a more serious aging problem. Most of the traditional library management systems are based on outdated technical architectures that cannot effectively support the development of modern information technology. For example, the system's functionality is difficult to expand, the interface is not user-friendly, and the user experience is poor. In addition, many of the older systems have high maintenance costs, poor stability, and are unable to meet the rapidly growing demand for lending and borrowing. These problems seriously affect the management efficiency and service quality of libraries [1] [2]. Therefore, how to update the technical architecture and improve the performance of the library management system has become an important issue that needs to be solved.

B. System Architecture

SpringMVC is a mature and efficient Java web framework that is widely favored for its modularity, high decoupling, and flexibility. By using the SpringMVC framework, developers can quickly build a library management system that meets modern needs. The framework has a clear MVC (Model-View-Controller) structure that effectively separates the business logic, data access, and view layers, improving the maintainability and scalability of the system. The system uses MySQL for data storage and Tomcat as the web server [3]. By solving the instability of the previous system, the Spring-based solution provides a more reliable library management experience [4]. At the same time, the system is designed to achieve fast performance and high security, ensuring fast and secure operations [5]. In the library management system, the application of the SpringMVC framework not only improves the operational efficiency of the system, but also makes the system more modular, facilitating expansion and upgrades. For example, SpringMVC allows for the creation of intuitive interfaces for both library staff and patrons, improving user engagement and satisfaction [6] [7].

Compared to traditional databases, MySQL has become the database of choice for more and more library information systems due to its efficient query performance, low-cost deployment and strong scalability. The advantages of the

MySQL database lie not only in its efficient processing power, but also in its good integration with the Java development environment, especially its compatibility with the Spring framework, which makes system development and maintenance much easier. MySQL's architecture allows for easy scaling, accommodating growing data needs as libraries expand their collections [8]. Libraries can implement web-based systems that leverage MySQL to manage user interactions and data processing efficiently [9]. By using the MySQL database, the library management system can realize functions such as quick queries, data backup and recovery, thereby ensuring the integrity and security of the data.

C. User Experience

As user needs change over time, systems need to be optimized to improve the user experience for both students and administrators. Automation of library processes through modern software significantly boosts staff productivity and improves service delivery to users [10]. For example, the system can enable students to borrow the books they need more efficiently by simplifying the process of borrowing, reserving and returning books. At the same time, the system should provide a user-friendly interface design that allows administrators to easily manage books, check inventory and generate statistics. Through these optimizations, the system can effectively improve the quality of library services and user satisfaction..

III. SYSTEM ANALYSIS

A. System Overview

The system provides support for book management and lending, there are two roles of readers and librarians, librarians can carry out unified management of books through the project, it is convenient to do the book entry and quick search, while also providing readers with borrowing and returning services, the system makes a clear record for borrowing, it is convenient for librarians to make statistical inquiries, in addition to readers can also be managed. Readers only need an account to use the system to borrow and return books.

This library management system provides different functions according to the roles, the administrator enters the system by logging in, and in the home page, you can navigate through the system to enter into the different functions for books and readers to borrow and manage. First of all, it is the function related to library management. In the library management function, the administrator can view the information of all the books, add books and view and maintain the information of the books. In order to provide borrowing and returning functions, first use the reader management function to add and maintain reader information, add new readers and maintain other reader information. After borrowing management in the borrowing and returning log function to query the borrowing and returning log, while you can delete the log.

Readers enter the system by logging in, and then in the home page of the system, they can search for the books they are interested in in the Find Books function, and in the search results, they can further view the details of the books to understand the books they are interested in. If they need to borrow, they only need to use the borrowing and returning functions on the books they are interested in.

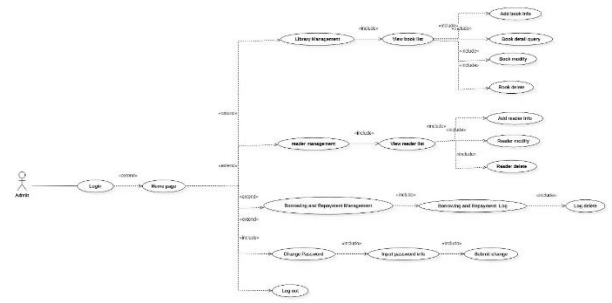


Figure 1: Use case diagram for administrator

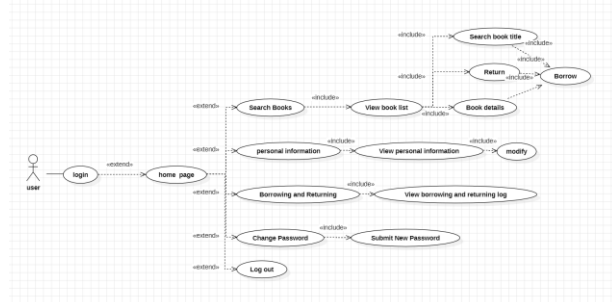


Figure 2: Use case diagram for readers

B. Hardware and Software Requirements

The system in the hardware does not have special requirements, but in order to ensure the stability and performance of the system, it is recommended to use the following basic configuration to meet the implementation of the deployment. First of all, the system supports Windows and Linux platforms, there is no major difference in the hardware requirements, multi-core CPU frequency of at least 2.0GHZ, at least 4GB of RAM, recommended 8GB or more, in order to ensure that the smooth operation of the system and data processing capabilities. The hard disk has at least 10GB of available space, and SSD is recommended to improve the read/write performance of the system. Enough storage space can store more book information and reader data.

The system uses the MVC design pattern, in the data layer using the Mysql database to store data information and MyBatis to maintain the data, this combination of technologies not only ensures the performance of the database storage, but also because of the mature technology framework for the follow-up maintenance has brought flexibility and convenience. In the view layer using JSP to develop the user interface, this technology is simple and easy to use to meet the user's functional requirements. In the control layer using Spring MVC technology framework, this technical solution is very suitable for the design of web layer development. Library management system in the overall technology based on the JAVA language Spring + Spring MVC + MyBatis + Mysql classic technology solutions, and the use of Maven for package management, this technology not only supports the implementation of cross-platform deployment on Windows and Linux and because of the clear architecture and module independence in scalability and maintenance cost advantages.

IV. SYSTEM DESIGN

A. Design Overview

Users are divided into two categories: readers and librarians. Librarians can modify readers' information, modify bibliographic information, and view all borrowing and returning logs, etc. Readers can only modify their personal information, borrow or return books, and view their own borrowing and returning logs.

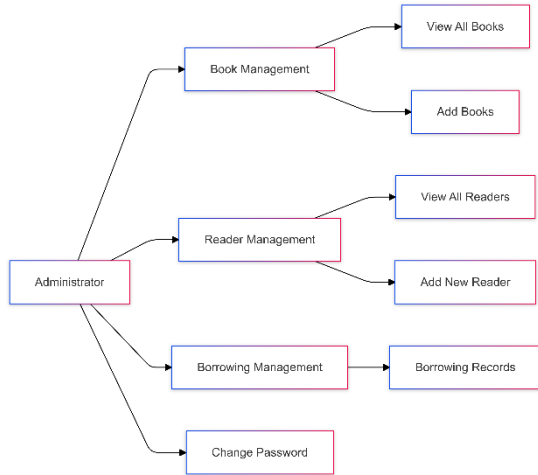


Figure 3: Project Module

B. User Interface Design

A simple and beautiful interface can enhance the user experience. So there is a need to explore how to design an effective library management system interface that prioritizes user experience while ensuring functionality. A minimalist approach was adopted in the registration and login interface, displaying only the necessary input fields. A concise system navigation function was adopted in the system home page, using three first-level menus to clearly prompt users on how to use the system. The data in the system interface is mainly laid out in tables, with concise data display to guide the user to focus on the information of the data itself, and through the highlighting feedback of the mouse movement to enhance the user experience in details, while focusing on the use of fonts and colors to make the whole system interface with a sense of hierarchy and the difference between the primary and secondary. For example, the book management interface displays all the books in a list, selectively displaying the main items of the book including title, author, publisher, price, remaining quantity, and using different color buttons at the back of the entries to prompt the user for further operations. The whole interface provides clear visual instructions to the user, and the selection process is simple and straightforward.

C. Database ER diagram

The database is designed with a book information table, a borrowing table, a reader table, a book categorization table, and an account management table.



Figure 4: ER diagram

D. Class Diagram

Book Management: contains all the information of the book, the attribute is the information of the book, the administrator and the user can view the list and details, the user can borrow and return the book, and only the administrator can edit, delete, and add the book.

Borrowing And Returning: contains the borrowing and returning information, the borrowing and returning attribute of the reader information, and both administrators and users can view the list, and only the administrator can delete the borrowing and returning records.

Reader Management: contains all the information of the reader, the attribute is the personal information of the reader, the administrator can view, edit, delete all the readers, and can add the reader, the reader can only view and edit the personal information.

Account Management: contains all the information of the account, the attribute is the basic information of the account, and both administrators and users can change their passwords and use the account to log in to the system.

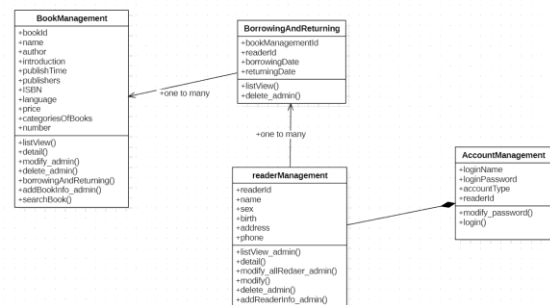


Figure 5: Class diagram

V. SYSTEM IMPLEMENTATION

A. User Interface

The interface of this system uses JSP as the page template, the page structure is designed using HTML, page interaction is implemented using JavaScript, and styles are designed using CSS. The login page uses HTML to design the page layout, form elements to store the account and password, JS to bind the login button to validate the input information, and Ajax to submit the form data to the server for login verification. The system homepage also uses the <div> tag for page layout, and CSS is used to add interactivity to the system navigation menu. The display of the function page is

embedded into the main interface of the current system homepage using the MVC view rendering method. This embedded method does not refresh the entire system homepage, which improves the user experience. On the add book screen, a form is used to record the book information entered by the user. The submit action triggers JS to verify the book information, and after the content validity is verified, it is submitted to the server using POST to complete the book information entry. The main interface of the book management uses an HTML table to display the book information by item, and uses JSP's EL tags to loop through the book data and bind it to the page table, and uses CSS to beautify the table to enhance the user experience.

The user interface as a whole uses JSP as the page template, HTML elements to complete the page layout, and JS and CSS to beautify the interface and achieve interaction to enhance the user experience. The data generated by the server is passed to the template, and EL expressions complete the filling of the page data. The technical solution of the system interface is mature and reliable, with clear responsibilities and functions of each component. While enhancing the user experience, it is also very convenient for subsequent maintenance.

B. Application Details

This system uses JSP for the front end and Spring and Spring MVC for the back end to provide services. A relational database such as MySQL is used to store book data and other related system information. Maven is used for project construction and dependency management, and Mybatis is used to manipulate database entity object data to complete storage and query functions. This technical solution is mature and can meet the requirements of this system.

After the system is deployed and running, the user's request to access the homepage will be judged by the server to see if the user is logged in, and the user will be redirected to the relevant page. Here, the MVC controller determines whether login information exists in the session and uses redirection technology to redirect the user's browser to the specified page. The user's login operation will be basically authenticated by the page JS, and the login interface of the server will be requested via Ajax. The server will receive the user's login information, call the login service to complete the authentication, and return the authentication result to the user. The login service will use Mybatis to call the database to query the user's information to determine whether it exists. On the homepage of the system, access the book list interface through the system navigation menu to initiate a request to the corresponding interface on the backend. The MVC controller calls the book service to query the relevant book information and passes the data to the view layer through the Model And View. The view layer calls the JSP template for the book list interface, uses EL expressions to obtain the book data and bind it to the table. After compiling the JSP template on the server side, the result is returned to the user's browser.

After the user accesses a function in the system interface, the request will be passed to the MVC controller on the server. The controller will call different services according to the interface to process and generate data. The service will then call the database to query and process the results. Finally, the view layer will complete the rendering of the data and return the results to the user's browser. At the same time, we also use

JS and CSS to beautify the interface and interaction to enhance the user experience.

VI. SYSTEM TESTING

The system mainly completes unit testing. The testing environment is deployed on a personal laptop, the configuration is Windows 10 Pro, the CPU is an i5 quad-core, and the memory is 8GB. This environment meets the basic requirements for system deployment. The system's operating environment is Java 8, MySQL 5.7, Spring 4.2.2, and Maven 3 is used to build and deploy.

Test cases	Results
User registration and login	Pass
Book management, adding books and searching	Pass
Reader entry and editing	Pass
Book borrowing and returning	Pass

Table 1: Test results

The above tests were performed using the Chrome browser, and unit tests were carried out on the core functions of the system. This testing method covers the overall processes and functions of the system and can therefore be used as a reference. The test results verify the usability of the system functions.

This test satisfies the requirements of the hardware environment and software environment, and the results of the unit tests verify that the system meets the requirements for release and launch.

VII. CONCLUSION AND FUTURE WORKS

A. Conclusion

This web-based library management system, developed using the Spring Framework in Java, provides a powerful solution for library management. The system helps library staff manage books and reader information more efficiently and provides readers with a more convenient and efficient borrowing service. Overall, the system solves common problems faced by traditional systems while improving the efficiency of management personnel, the stability of the system, and the user experience.

In addition, developing the library management system has also made me aware of some important management principles and practices:

1. User-centered

The core of a library management system is to serve readers. Therefore, during the system development process, we must be user-centric, fully consider user needs and the user experience, and ensure that the system can provide users with convenient, efficient, and comfortable services.

2.The importance of data management

A library management system involves a large amount of data management work, such as book management, reader information, borrowing records, etc. Therefore, data management is very important, and a standardized and

effective data management mechanism must be established to ensure the security, integrity, and reliability of the data.

3. Technology updates and upgrades

With the development of technology and changes in society, the library management system also needs to be constantly updated and upgraded to meet user needs and the requirements of the times. Therefore, we must pay close attention to technological trends and changes and carry out timely upgrades and improvements.

4. Equal emphasis on management and service

The library management system is not only a management tool, but also a service platform. Therefore, in the development and operation process, it is necessary to consider the needs of both management and service, not only to manage library resources and reader information, but also to provide efficient and high-quality borrowing services.

VIII. REFERENCES

- [1] Haziri, M. (2019). User experience and its importance in banking software: A study of user experience during development.
- [2] Paiva, J. O. V., Borges, B., Duarte, P., Santos, I. S., Andrade, R. M. C., & Lelli, V. (2022). PRO-UX: System redesign process for improving the UX. Proceedings of the ACM Symposium on Applied Computing, doi: 10.1145/3571473.3571501.
- [3] Zhang, J., & Sun, Y. (2022). Design and implementation of library management information system based on SpringMVC. Proceedings of the 2022 International Conference on Intelligent Systems and Engineering (ICISE), 41-45.
- [4] Shaw, B. K., Prajapati, P., Ghosh, A., Sarkar, B., Giri, H., & Basak, S. (2022). Library management system (LMS) using JAVA. International Journal for Research in Applied Science and Engineering Technology (IJRASET), 10(4), 328-333.
- [5] Wei, L., & Zhang, J. (2022). Research on Java-based library management system. Proceedings of the 2022 Conference on Intelligent Programming and Artificial Engineering (CIPAE), 304-307.
- [6] Priyanka, S. (2024). Library management system. Journal of Universal Bibliography, doi: 10.69758/jubi7297.
- [7] Seema, J. (2023). Library management system. International Journal for Research in Applied Science and Engineering Technology, doi: 10.22214/ijraset.2023.55279.
- [8] Ziang, W. (2024). The design of library database management system based on MySQL. Applied and Computational Engineering, doi: 10.54254/2755-2721/38/20230527.
- [9] Irsyad, P., Irwansyah, I., & Dian, A. W. (2018). Sistem informasi perpustakaan berbasis web pada STMIK Nusa Mandiri menggunakan PHP dan MySQL. Jurnal Sistem Informasi, doi: 10.51998/JSI.V7I1.204.
- [10] (2023). 4. Automating library processes using the oss. Vestnik Torajgyrov universiteta, doi: 10.48081/qkoz7457.