**LIBRARY MANAGEMENT SYSTEM**

**Table of Contents**

[Introduction 3](#_Toc201764267)

[Justification of Selected Data Structures and Algorithms 3](#_Toc201764268)

[Time Complexity Analysis with Pseudo-Code 4](#_Toc201764269)

[Testing Approach and Test Cases 6](#_Toc201764270)

[Conclusion 7](#_Toc201764271)

[References 9](#_Toc201764272)

[Appendices 10](#_Toc201764273)

[Appendix 1: Github Repository Link 10](#_Toc201764274)

[Appendix 2: Software Images 11](#_Toc201764275)

# Introduction

The project focuses on developing and establishing an effective method of resolving a particular problem that requires a calculation. The major goal is to understand and best the speed of various data structures and algorithms compared to the task. In the project, it is possible to emphasize the construction of proper data structures that will take less time and will allow a high level of efficiency of the system in general. In an effort to evaluate the effect of such selections, a mixture of theory and practice will be employed.

***Description of the Project***

In this project, the available problem necessitates utilising and manipulating a huge amount of data and thus, error-free algorithms to fetch and sort data and update data are mandatory. The implementation will aim at streamlining the performance of operation like search, sorting, and insertion operation using data structures such as arrays, linked lists, trees, among others, depending on the given task requirements.

The main aim of this project does not lie solely in the development of a solution that would work but in the evaluation of the effective use of the algorithms selected. The analysis of the time complexity will be made so that the chosen method performs well with larger data sets. Also, testing will be conducted so that the solution can be robust, workable, and reliable in different circumstances, with the key focus on edge cases. The project outcomes will help obtain a better comprehension of algorithmic efficiency in practice.

# Justification of Selected Data Structures and Algorithms

Data structures and algorithms are also important; a successful performance is largely dependent upon the correctness of their choice. Several data structures will be selected according to the problem characteristics and the computational complexity introduced by them, in this project (Anshori *et al.* 2022). The competent and efficient selection of data structures and algorithms is supposed to offer the best performance relative to the time and space requirements of a particular problem.

**Data Structures**

* ***Arrays:*** Arrays are selected due to their simplicity and direct access to array elements through indices. They are most convenient in situations when the fixed-size data is processed, and fast access to elements is required. It can have lots of computational applications in that the data elements can be updated and retrieved efficiently in the array structure.
* ***Linked Lists:*** Linked lists are chosen due to their desirable performance of processing dynamic data when the size of the data set is dynamic. Linked lists also record efficient insertion and deletion of data as compared to arrays, which would be suitable when the operation needs regular updates of the data on the dataset.
* ***Binary Search Trees (BST):*** BSTs are selected because of the effective search, insert, and delete operations. These operations are O(log n) in a balanced BST, and so it is appropriate in an application that must store sorted data or often has to find particular values.

**Algorithms**

* ***Quick Sort:*** The choice of Quick Sort is its O(n log n) average time complexity, which makes the algorithm one of the fastest sorting solutions when dealing with large amounts of data (Verma, 2023). It uses a divide-and-conquer strategy and thus it can sort data efficiently in large sets, compared to bubble sort, insertion sort, etc.
* ***Binary Search:*** In search of sorted data, a binary search is used because of its logarithmic search time of O(log n) making it orders of magnitude faster than linear search on large datasets.

These data structures and algorithms have been selected so as to find a compromise between simplicity of implementation and efficiency in the sense that the solution should not only satisfy needs of the problem but also should be of optimal and scalable form.

# Time Complexity Analysis with Pseudo-Code

***1. Login System***

|  |
| --- |
| BEGIN Login Form  Initialize database connection  Display login screen  ON clicking Login button:  IF username or password is empty:  SHOW error "Invalid Input"  ELSE:  Open database connection  Check if user exists with provided credentials:  IF user exists:  Check if user is admin:  IF user is admin:  Open Admin Menu  ELSE:  Open Student Menu  ELSE:  SHOW error "Invalid Credentials"  Close database connection  END IF  END Login Form |

***2. Student Registration***

|  |
| --- |
| BEGIN Student Registration Form  Initialize database connection  Display registration screen  ON clicking Register button:  IF required fields are empty:  SHOW error "Missing Values"  ELSE:  Open database connection  INSERT new student data into database  Close connection  SHOW success message "Registered Successfully!"  END IF  END Student Registration Form |

***3. Book Management***

|  |
| --- |
| BEGIN Add/Edit Book Form  Initialize database connection  Display book management screen  ON clicking Add Book button:  IF required fields are empty:  SHOW error "Missing Inputs"  ELSE:  Open database connection  INSERT new book data into Books table  Close connection  SHOW success message "Added Book"  END IF  ON clicking Edit Book button:  IF required fields are empty:  SHOW error "Missing Inputs"  ELSE:  Open database connection  UPDATE existing book data in Books table  Close connection  SHOW success message "Edit Saved"  END IF  END Add/Edit Book Form |

***4. Book Supply Management***

|  |
| --- |
| BEGIN Add Book Supply Form  Initialize database connection  Display available books for supply  ON selecting book and clicking Add Supply:  IF quantity is valid:  Open database connection  INSERT supply transaction record into BookSupplyTransaction table  Update BorrowingTransaction table  Close connection  SHOW success message "Added Supply"  ELSE:  SHOW error "Input Error"  END IF  END Add Book Supply Form |

# Testing Approach and Test Cases

The testing strategy of the Library Management System consists of testing system, integration testing, and unit testing.

* ***Unit Testing:*** Individual forms (e.g., login and student registration, book management, etc.) are to be tested separately to guarantee that their functionality is performed according to the expectations.
* ***Integration Testing:*** Once individual components were proven, integration testing will be applied to make sure the data passes between the forms and the database normally (Yusuf *et al.* 2022). As an example, once a student has registered, the new log in credentials should be identified by the log in form.
* ***System Testing:*** The full system will be with all tests done to confirm the end-to-end activities. This will involve testing the connection among various modules like the borrowing of books, returns of books and a change in the stock.

**Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Input** | **Expected Outcome** |
| TC\_01 | Test valid login credentials | Username: 'user1', Password: 'pass1' | Login successful and redirects to correct menu |
| TC\_02 | Test invalid login credentials | Username: 'user1', Password: 'wrongpass' | Error message: "Invalid Credentials" |
| TC\_03 | Test empty login fields | Username: '', Password: '' | Error message: "Invalid Input!" |
| TC\_04 | Test student registration with all fields filled | First Name: 'John', Last Name: 'Doe', etc. | Successful registration and success message |
| TC\_05 | Test student registration with missing fields | First Name: 'John', Last Name: '', etc. | Error message: "Missing Values!" |

**Table 1: Test Cases**

# Conclusion

Implementation of the Library Management System in user authentication, borrowing, returning and book supply transactions has been effected successfully. The system has the features of both the students and the administrative team members where book borrowing and update involve the student and book management, user and stock updating involves the administrative staff. Using SQL database integration, the system guarantees that real time data processed through the system is updated accordingly and this makes it easier to operate with the users and the library system.

***Limitations***

There are limitations in the system. Plaintext passwords should be hashed and encrypted so that they are not deciphered easily in the login process, which ideally is not the case. Also, a better approach to errors should be provided in the system, especially in cases such as a network shutdown or a database being unplugged (Gusvita and Alon, 2021). The interface would be made more user-friendly, and it would help the user to make it easier and especially with new users to the library system. SQL queries also are also intensively used in the system and there might exist performance problems when dealing with larger data or more complicated queries.

***Reflection***

Reflectively, albeit serving the minimum needs of the management of library transactions, in the future, enhancements may be made to the system, including maxims on security, performance optimization, and improvement on the user experience. There is also the need to include extra features like notifications using emails regarding overdue books, live updates of the stocks, and a mobile-friendly system which will make this system more user-friendly and reliable.

# References

Anshori, M., Elynawati, E. and Alfatchussadiqin, A., 2022. Implementation of Library Management in Improving Excellent Service in Vocational High Schools. *Nazhruna: Jurnal Pendidikan Islam*, *5*(2), pp.830-842.

Verma, M., 2023. Novel study on AI-based chatbot (ChatGPT) impacts on the traditional library management. *International Journal of Trend in Scientific Research and Development (IJTSRD)*, *7*(1), pp.961-964.

Yusuf, T.I., Adebayo, O.A., Bello, L.A. and Kayode, J.O., 2022. Adoption of artificial intelligence for effective library service delivery in academic libraries in Nigeria. *Library Philosophy and Practice (e-journal)*, *6804*, pp.1-13.

Gusvita, M. and Alon, F., 2021. Leadership Supervision in Improving Library Services at the Batanghari Regency Library and Archives Office. *At-Tasyrih: jurnal pendidikan dan hukum Islam*, *7*(2), pp.107-119.

# Appendices

## Appendix 1: Github Repository Link

<https://github.com/shiivampatell12/LibraryManagementSystem->

## Appendix 2: Software Images

|  |
| --- |
|  |
|  |
|  |