TRIBHUVAN UNIVERSITY

Faculty of Institute of Science and Technology

Madan Bhandari Memorial College

Bijulibazar , Kathmandu

A Project Report On

“Library Management System”

For

System Analysis and Design (CSC 315)

Submitted By:

Yunish Luitel (T.U. Exam Roll No : 29176)

Submitted to :

Laxmi Prasad Yadav

Department of Computer Science and Information Technology

Table Of Contents

Chapter 0 : Abstract

0.1 Introduction : The increasing volume of information and the evolving needs of library users have necessitated the development of efficient, modern solutions for library management. This project focuses on the design and implementation of a comprehensive Library Management System (LMS) aimed at enhancing the efficiency of library operations and improving user experience.

The LMS provides an automated platform for managing library resources, including cataloging, user management, and circulation control. It addresses the key challenges faced by traditional library systems, such as manual record-keeping, inefficient book tracking, and limited user interaction capabilities. By integrating various functionalities into a unified system, the LMS streamlines administrative tasks, facilitates real-time updates, and offers an intuitive interface for both librarians and patrons.

0.2 Key features : Key features of the LMS include a robust catalog management system, user account management, book borrowing and returning functionalities, and the ability to generate detailed reports. The system is designed to be scalable, secure, and user-friendly, ensuring that it can handle growing amounts of data and users while maintaining high performance and data privacy.

0.3 Project Approach : This project involved analyzing existing library management practices, designing the system architecture, developing core functionalities, and evaluating system performance through rigorous testing. The resulting LMS aims to provide a reliable and efficient solution for modern libraries, enhancing operational efficiency and user satisfaction.

0.4 Summary : The successful implementation of the LMS demonstrates its potential as a valuable tool for libraries seeking to modernize their operations and better serve their communitie.

CHAPTER 1: INTRODUCTION

1.1 Background

In the digital age, efficient management of library resources has become increasingly important. Traditional library systems, while functional, often struggle with the demands of modern users and the volume of information they need to handle. The development of a Library Management System (LMS) aims to address these challenges by offering a comprehensive, automated solution for managing library collections, user interactions, and administrative tasks.

A well-designed LMS enhances the efficiency of library operations, improves the user experience, and facilitates better resource management. With the rise of digital libraries and online catalogs, there is a growing need for systems that can integrate various functionalities such as cataloging, borrowing, and reservation of books in a seamless and user-friendly manner.

1.2 Problem Definition

Despite the advantages of automated Library Management Systems, several challenges need to be addressed to optimize their effectiveness and usability.

One significant issue is the integration of various modules, such as catalog management, user management, and circulation control. Ensuring that these modules work together smoothly and provide accurate information is crucial for the system's reliability.

Another challenge is the usability of the system. Users, including librarians and patrons, should find the LMS intuitive and easy to navigate. A complex or poorly designed interface can hinder the adoption and effectiveness of the system.

Data security and privacy are also critical concerns. An LMS must protect sensitive user information and manage data access appropriately. This includes securing user accounts, transaction records, and personal details from unauthorized access.

Additionally, system performance and scalability are important. The LMS should be able to handle a growing number of users and data without performance degradation. Efficient database management and system optimization are essential to address these concerns.

1.3 Objectives

The primary goal of this project is to develop a robust and user-friendly Library Management System that addresses the current challenges in library management. The specific objectives are:

- To design a comprehensive system that integrates catalog management, user management, and circulation control.
- To develop an intuitive user interface that enhances the user experience for both librarians and patrons.
- To ensure the security and privacy of user data and system transactions.
- To implement scalable system architecture that can handle increasing amounts of data and users efficiently.

By achieving these objectives, the project aims to provide a reliable and efficient solution for modern library management, improving operational efficiency and user satisfaction.

1.4 Scope

This project focuses on the design, development, and evaluation of a Library Management System. The scope includes:

- Analyzing the requirements for catalog management, user management, and circulation control.
- Developing the system components, including the user interface, database management, and system integration.
- Implementing features for efficient book cataloging, user account management, borrowing and returning books, and generating reports.
- Evaluating the system's performance, usability, and security through testing and user feedback.

The project will not cover aspects such as physical library infrastructure or external integrations beyond the LMS itself.

1.5 Limitations

The project may encounter several limitations, including:

- **User Familiarity:** The LMS requires users to be familiar with basic computer and internet operations. Users with limited technical skills may face difficulties.
- **System Performance:** High traffic volumes or simultaneous user access may affect system performance. Optimization and load testing are necessary to address potential issues.
- **Customization:** The system may not offer extensive customization options for specific library needs or workflows.
- **Cost Constraints:** Developing and maintaining the system may involve costs, which could be a limitation for smaller libraries or institutions with limited budgets.
- **Data Privacy:** Implementing stringent security measures to protect user data may raise privacy concerns that need to be carefully managed.

CHAPTER 2: SYSTEM ANALYSIS

2.1 INTRODUCTION

System analysis involves studying a system to understand its objectives and improve its functionality. It is a critical phase in the development of a Library Management System (LMS), ensuring that all components work efficiently to meet user needs. Key tasks in system analysis include:

- **Requirement Gathering:** Collecting requirements from stakeholders such as library staff, patrons, and administrators to understand their needs and expectations for the LMS.
- **Functional Analysis:** Identifying the necessary functions and features of the LMS based on gathered requirements.
- **Data Analysis:** Examining the data requirements, including catalog information, user profiles, and transaction records.
- **Process Modeling:** Creating models to illustrate how the LMS will operate, including book borrowing, returns, and catalog searches.
- **Use Case Analysis:** Analyzing various use cases such as book checkouts, renewals, and reservations to define system interactions.
- **Performance Analysis:** Assessing the system's performance, including response times, reliability, and its ability to handle multiple users.
- **Security Analysis:** Evaluating the system's security measures to protect user data and library records from unauthorized access or breaches.
- **User Interface Analysis:** Assessing the usability of the LMS interface for both library staff and patrons to ensure it is intuitive and accessible.
- **Testing:** Identifying and fixing any issues through comprehensive testing to ensure the system functions as intended.

2.2 FUNCTIONAL REQUIREMENTS

Functional requirements specify the capabilities needed for the LMS to serve its users effectively. For the LMS, the following requirements are essential:

- **User Registration and Authentication:** The system must allow users to register and log in securely with basic information such as name, email, and library card number.
- **Catalog Management:** The system must enable librarians to add, update, and delete catalog entries, including books, magazines, and other materials.
- **Borrowing and Returning:** The system must handle book checkouts and returns, updating availability status and due dates.
- **Reservation System:** The system must allow users to reserve books and notify them when the items are available.
- **Search and Discovery:** The system must offer robust search capabilities for users to find books and resources based on various criteria.
- **User Account Management:** The system must manage user accounts, including tracking borrowing history, fines, and personal details.
- **Reporting:** The system must generate reports on library usage, book inventory, and user activity.
- **Notifications:** The system must send reminders for due dates, reservations, and overdue books.
-

2.3 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements address the quality attributes of the LMS:

- **Performance:** The system must efficiently handle a high volume of transactions and user interactions without delays.
- **Availability:** The system must be accessible 24/7, minimizing downtime to ensure users can access library services at any time.
- **Reliability:** The system should be reliable, with minimal errors and the ability to recover quickly from failures.
- **Security:** The system must protect user data and library records with encryption and secure access controls.

- **Usability:** The interface must be user-friendly, allowing easy navigation and interaction for both library staff and patrons.
- **Compatibility:** The system must be compatible with various devices and operating systems, including desktops, tablets, and smartphones.
- **Scalability:** The system should scale to accommodate growing numbers of users and library resources without performance degradation.

2.4 FEASIBILITY STUDY

Evaluating the feasibility of the LMS project involves assessing various aspects:

- **Operational Feasibility:** Examines whether the system will effectively meet the operational needs of the library and its users. The LMS should be user-friendly and suitable for both tech-savvy and less experienced users.
- **Technical Feasibility:** Assesses whether the system can be technically implemented with available resources. This includes evaluating the technical requirements and ensuring the necessary infrastructure is in place.
- **Economic Feasibility:** Compares the development costs of the LMS with the benefits it will provide. The system should offer a favorable cost-benefit ratio, delivering value through improved library management and user satisfaction.

2.5 Gantt Chart

Task Description	Start Date	End Date	Duration
Project Planning	01/05/2024	06/05/2024	5 days
Requirements Gathering	07/05/2024	17/05/2024	10 days
System Design	20/05/2024	30/05/2024	10 days
Front-End Development	01/06/2024	06/06/2024	5 days
Back-End Development	07/06/2024	22/06/2024	15 days
Testing and Bug Fixing	23/06/2024	04/07/2024	10 days
Deployment and Launch	05/07/2024	15/07/2024	10 days
Post-Launch Maintenance	16/07/2024	01/08/2024	15 days

Explanation:

- **Project Planning:** Define project goals, scope, and develop a plan.
- **Requirements Gathering:** Document functional and non-functional requirements for the LMS.
- **System Design:** Develop architecture, database schemas, and user interface designs.
- **Front-End Development:** Build the user interface and experience components.
- **Back-End Development:** Develop the database, and business logic.
- **Testing and Bug Fixing:** Conduct testing to identify and resolve issues.
- **Deployment and Launch:** Deploy the system and make it available to users.
- **Post-Launch Maintenance:** Provide ongoing support and updates.

2.6 Decision Table

Condition	Action 1	Action 2	Action 3
User is logged in	Access library catalog	View account details	Logout
User has borrowed books	View borrowing history	Renew books	Return books
Book is available	Reserve book	Notify user	Search for alternatives
Book is overdue	Pay fine	View overdue list	Return book

This decision table outlines actions based on different conditions. For example, if a user is logged in, they can access the catalog, view their account details, or log out. If a book is overdue, the user can pay the fine or view the overdue list.

CHAPTER 3: SYSTEM SPECIFICATIONS

3.1 SERVER REQUIREMENTS (Minimum Requirements)

- ❖ Asgiref = 3.8.1
- ❖ Django = 5.0.7
- ❖ Pillow = 10.4.0
- ❖ Sqlparse = 0.5.1
- ❖ Pip = 23.1.2
- ❖ Python = 3.x.x
- ❖ Virtual Environment = 20.x.x

3.2 DEVELOPMENT TOOLS USED

- Operating System : Linux 24.04
- IDE : Visual Studio 2023
- Front-End : HTML, CSS , Bootstrap
- Back-End : Python , Django
- Back-End Connectivity: MySQL

CHAPTER 4- MODULES AND THEIR DESCRIPTION

4.1.1 User Module

The User Module is a critical component of the Library Management System, as it allows users to interact with the system and access the services it provides. This module is designed to offer a seamless and user-friendly experience, enhancing user satisfaction and loyalty. The User Module typically includes the following features:

- **User Registration :** Allows users to create accounts and register with the library system.
- **Search and Borrow :** Enables users to search for books, reserve them, and borrow them from **the** library.
- **Return Management :** Facilitates the process of returning borrowed books and managing any associated fines.
- **Notifications and Alerts :** Sends reminders and alerts to users about due dates, reservations, and other important information.
- **User Feedback and Reviews :** Allows users to provide feedback and reviews on the books they have read.
- **Customer Support :** Provides assistance to users through various channels such as email, chat, or phone.

By implementing a robust User Module, the Library Management System can ensure that users have a smooth and efficient experience when interacting with the library's services.

4.1.2 Librarian Module

The Librarian Module is an essential part of the Library Management System, as it allows librarians to manage the library's operations effectively. This module is designed to streamline the workflow of librarians and enhance the overall efficiency of the library. The Librarian Module typically includes the following features:

- **Book Management:** Allows librarians to add, update, and delete book records in the system.
- **User Management :** Enables librarians to manage user accounts, including registration, updates, and deletions.
- **Loan Management :** Facilitates the process of issuing and returning books, including tracking due dates and fines.
- **Reservation Management :** Allows librarians to manage book reservations and notify users when reserved books are available.
- **Inventory Management :** Helps librarians keep track of the library's inventory, including stock levels and book conditions.
- **Reporting and Analytics :** Provides librarians with insights into library usage, user trends, and other relevant data.

By implementing a robust Librarian Module, the Library Management System can ensure that librarians have the tools they need to manage the library efficiently and effectively.

4.1.3 Administration Module

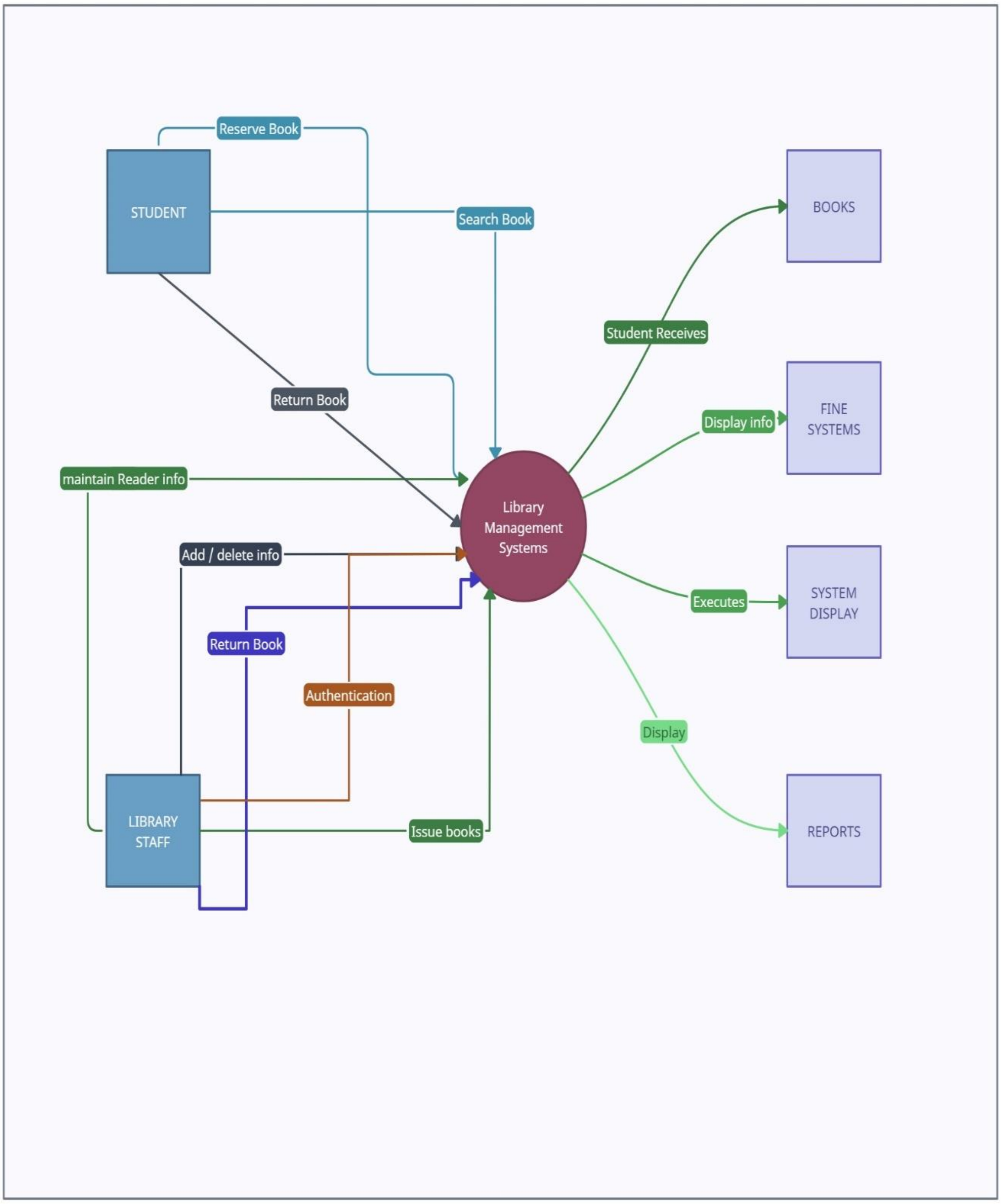
The Administration Module is an important component of the Library Management System, as it allows administrators to oversee and manage the system's operations. This module is designed to provide administrators with the necessary tools to ensure the smooth functioning of the library. The Administration Module typically includes the following features:

- **Dashboard :** Provides a comprehensive overview of the library's operations, including key metrics and statistics.
- **User Management :** Allows administrators to manage user accounts, including librarians and patrons.

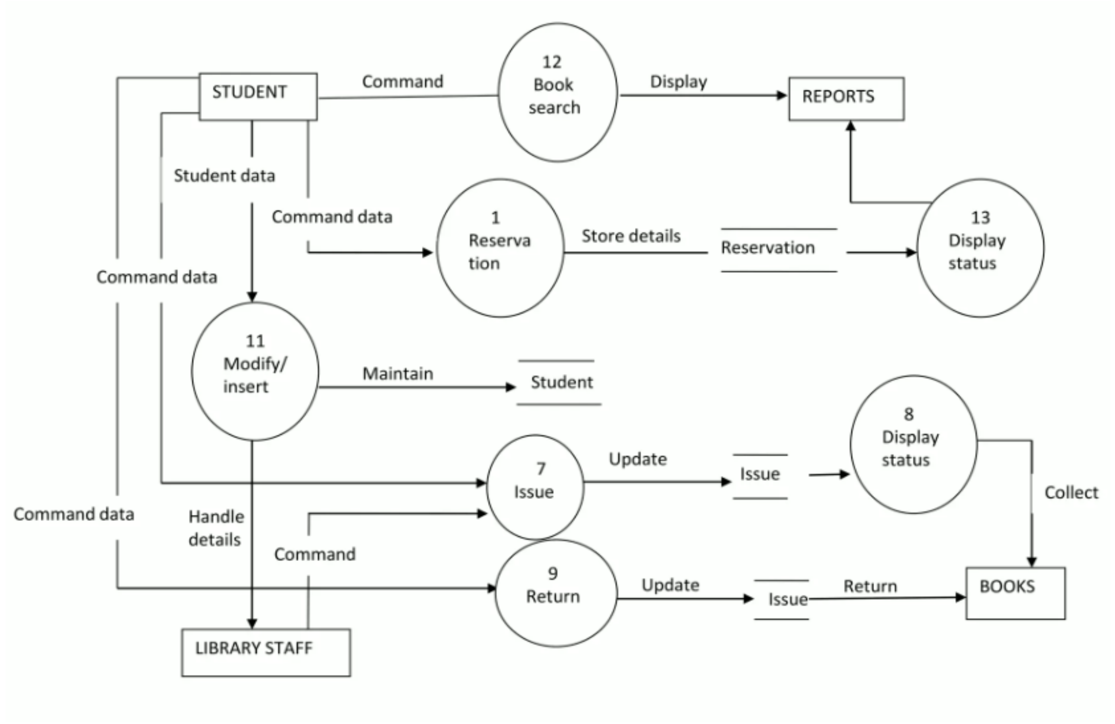
- **Book Management :** Enables administrators to oversee the addition, updating, and deletion of book records.
- **Fine Management :** Facilitates the management of book loans, including tracking due dates and fines.
- **System Configuration :** Allows administrators to configure system settings, such as loan periods, fine rates, and other parameters.
- **Reporting and Analytics :** Provides administrators with detailed reports and analytics on library usage, user trends, and other relevant data.
- **Security and Access Control :** Ensures that the system is secure and that access is controlled appropriately.

By implementing a robust Administration Module, the Library Management System can be efficiently managed and maintained, leading to increased operational efficiency and improved service delivery.This module provides administrator-related functionalities. The administrator manages the entire application and maintains the profiles of users. The Administration Module has all privileges related to the entire library system, allowing the administrator to oversee and manage all aspects of the library's operations.

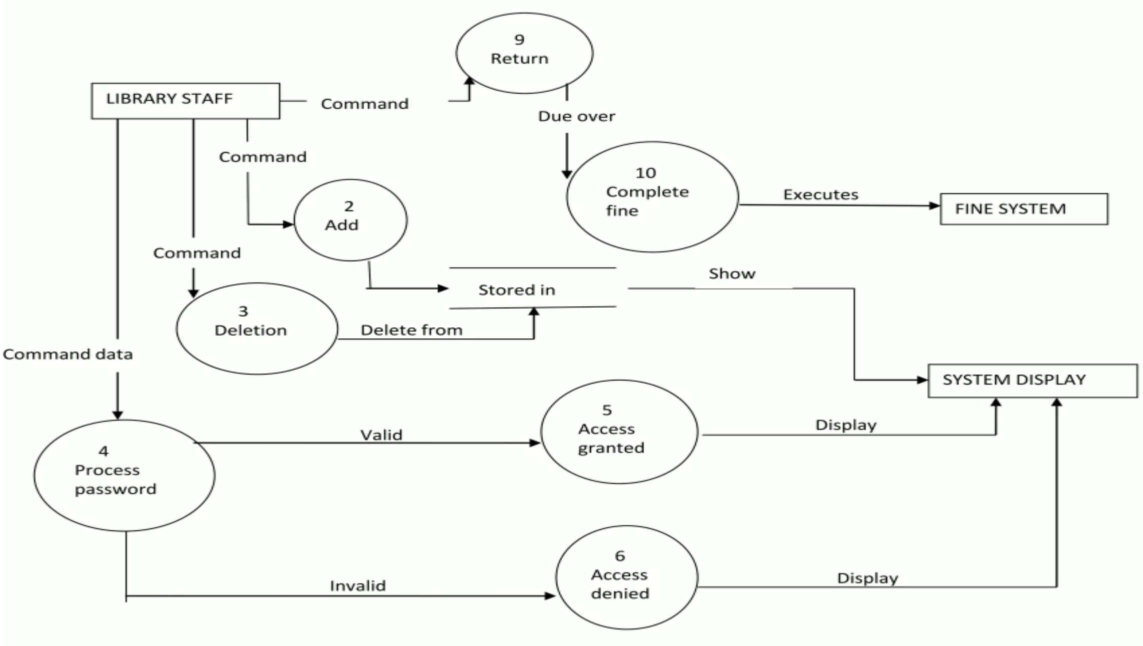
4.2 Context Diagram / Level 0 Data Flow Diagram



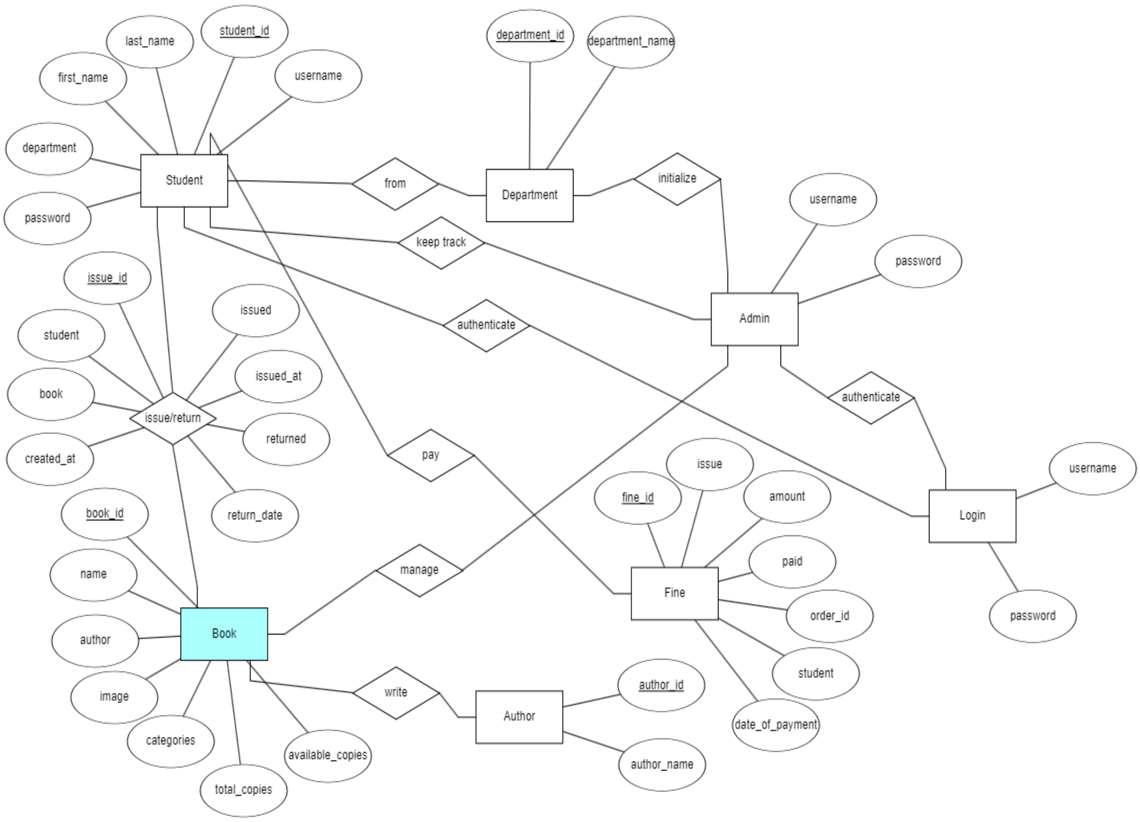
4.2.1: Level 1 Data Flow Diagram of Student



4.2.2: Level 1 Data Flow Diagram of Admin



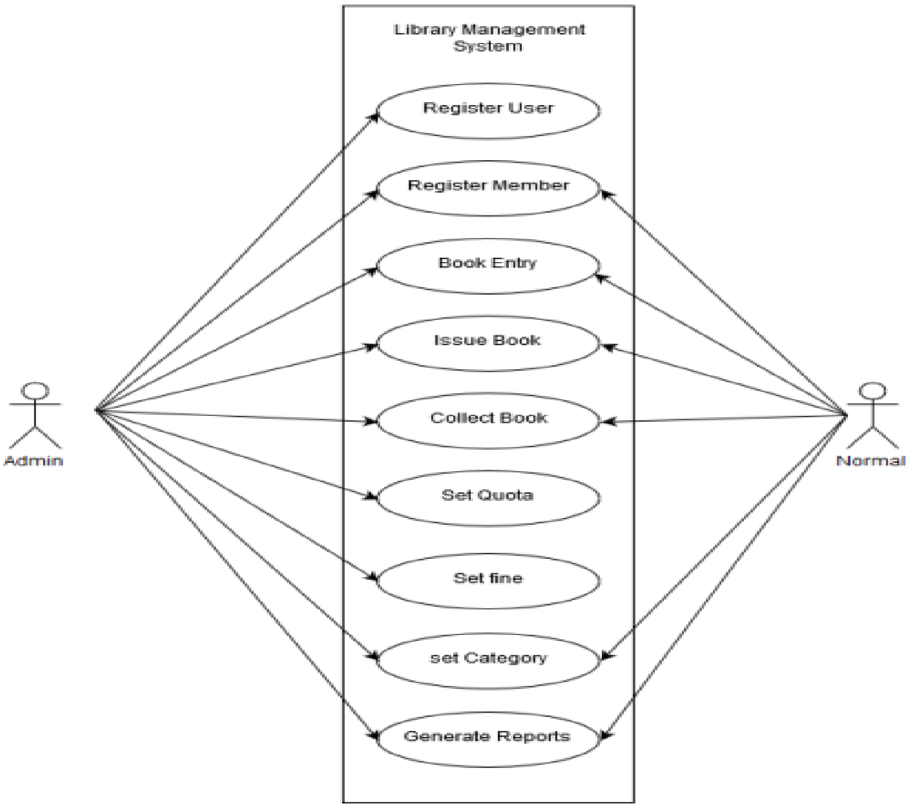
4.2.3: ER Diagram



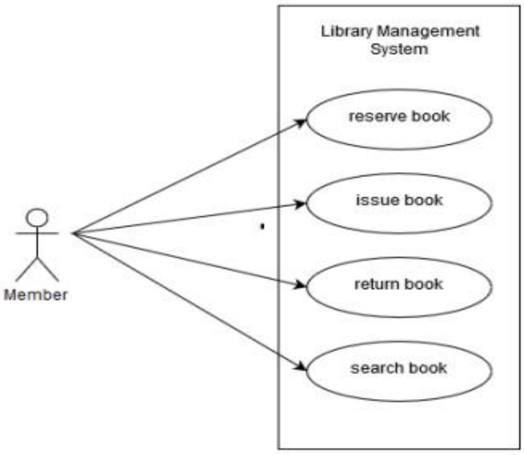
4.2.4 : Use Case

A use case is an approach used in system analysis to identify, clarify, and organize system requirements. The use case is built of a set of possible sequences of interactions between systems and users in a particular environment and relate to a particular goal. It consists of a group of elements (for example, classes and interfaces) that can be used together in a way that will have an effect larger than the sum of the separate elements combined. The use case should contain all system activities that have significance to the users. A use case can be thought of as a collection of possible scenarios related to a goal, indeed, the use case and goal are sometimes considered to be synonymous. Here, two use cases have been implemented. The first one, admin vs normal user, where the librarians are admins and have all the functions ready to be used, whereas students are normal users and have limited access. For the second one, the member use case, the students are the members and we focus more on defining rules of what a member can and cannot do.

4.2.5 : Use Case Admin vs Normal



4.2.6 : Use Case for member.



CHAPTER 5: SYSTEM TESTING

5.1 INTRODUCTION

Testing is very important for the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. Testing is running the program (or product) under various circumstances and conditions to find errors and bugs in it. [11] It is a critical part of software quality assurance and represents the ultimate review of specification, design, and coding. The testing phase involves testing a system using various test data. Testing involves the operation of the system or application under controlled conditions and evaluating the results. The controlled conditions should contain both normal and abnormal conditions.

5.1.1 Testing Objective

The test objectives are :

- To find out defects or issues in the application experienced by the end user
- To evaluate the performance of the application in order to improve it
- To ensure all the integrated modules are functioning correctly •
- To ensure the application is running smoothly on different software

To ensure that these objectives were fulfilled, some testing was conducted. This has been further discussed in the following section.

5.1.2 White Box

Testing White box testing, sometimes called glass-box testing, is a test case design philosophy that uses the control structure described as part of the componentlevel design to derive test cases. Using White box testing methods it can be confirmed that:

- All independent paths inside a module have been exercised at least once
- All logical decisions have been implemented.
- All loops at their boundaries and within their operational bounds are implemented.
- All internal data structures have been exercised to ensure their validity.

5.1.3 Black Box Testing

Black Box Test is testing without knowledge of the internal workings of the item being tested.When black box testing is practiced, the tester would only know the allowed inputs and what the expected outputs should be, but not how the program actually arrives at those outputs. It is because of this that black box testing can be regarded as testing with respect to the specifications, no other knowledge of the program is necessary

5.1.4 Test Phase

- **Unit Testing :** Unit testing is a software development process in which the smallest testable parts of an application, known as units, are individually and independently examined for proper operation. It refers to tests that verify the functionality of a specific section of code, usually at the function level. Unit testing was conducted to ascertain that the application functions properly and whether the function used in the application is operational or not. It is necessary to ensure the correct operation. Hence, each component/functionality has been tested and compared with the desired output. In this application, various units like book entry, check publisher entry, check book availability, check valid members, user creation, member registration with a valid username, check quota assigned to various members and fine calculation have been tested.

- **Integration Testing :** Integration testing is testing in which a group of components is combined to produce output. Also, the interaction between software and hardware is tested in this testing to check if software and hardware components have any relation.

- **Module Testing :** There are different modules in the application like book issue, return, renew, loss and fine calculation. Module testing was performed in order to know whether the modules do the task for which they were developed. After building the operational modules, they were tested to identify if there were any errors in the application. Various modules like book issue, book return, book renew, and book loss have been combinedly tested.

- **System Testing :** System testing was done to test the functioning of the system as a whole. It determines if the discrete modules function together as planned and whether discrepancies exist between the ways the system actually works and the way it was conceived. As this system is web-based, it is tested in various environments like in windows operating system and Mac computers.

- **User Acceptance Testing :** An acceptance test has the goal of selling the user on the validity and reliability of the system. It verifies that the system's procedures operate as system specifications indicate, and that the integrity of important data is preserved. User acceptance test is then done by the users. A limited number of users are given access to use the application before it is officially released. User motivation is vital for the successful functioning of the system. After that, a comprehensive test report is prepared. The report after this testing shows the system's tolerance, performance range, error rate, and accuracy

CHAPTER 6: SYSTEM IMPLEMENTATION

6.1 OBJECTIVE

The implementation phase marks the transition from the conceptual design of the Library Management System (LMS) to its actual deployment and operation. This phase is critical for ensuring that the system is effectively integrated into the library's existing workflows and meets the needs of its users. The key stages of implementation include:

- **Planning:** The initial step involves creating a comprehensive plan that outlines the objectives, timeline, and resources required for the successful deployment of the LMS. Planning involves assessing the current library operations, defining clear goals for the system, and preparing a structured approach for its implementation.
- **Training:** A crucial aspect of implementation is equipping library staff with the necessary skills to effectively use the new LMS. Training sessions are organized to familiarize users with the system's features, functionalities, and best practices. This ensures that staff can efficiently operate the system and leverage its capabilities to enhance library management.
- **System Testing:** Before the LMS goes live, extensive testing is conducted to ensure its reliability and performance. This stage includes verifying that the system meets all technical specifications, performs as expected under various conditions, and is free of critical issues. Testing helps identify any problems that need to be addressed before the system is fully operational.
- **Changeover Planning:** Transitioning from the old system to the new LMS requires careful planning to minimize disruptions. This involves developing a detailed strategy for data migration, managing user transitions, and ensuring that all operational aspects are smoothly shifted to the new system. Effective changeover planning is essential for a seamless integration process.

The implementation of the LMS involves coordination among various departments and stakeholders to ensure a successful rollout. An implementation coordination team is established to oversee the process, address any challenges, and ensure that all tasks are completed as planned. The team focuses on:

- **System Environment Assessment:** Evaluating the technical environment in which the LMS will operate, including hardware, software, and network infrastructure. This assessment ensures that the system is compatible with existing technologies and operates efficiently within the library’s setup.
- **Task Management and Team Coordination:** Assigning specific responsibilities to team members and ensuring that all tasks are carried out effectively. This involves organizing roles related to training, testing, support, and implementation to ensure that each aspect of the process is handled appropriately.
- **Stakeholder Engagement:** Consulting with library staff, administrators, and other relevant stakeholders to gather input and address concerns. Engaging stakeholders helps tailor the system to meet the specific needs of the library and ensures that their feedback is incorporated into the implementation process.
- **Resource Allocation:** Managing the necessary resources, including technical support, infrastructure, and backup systems. This involves ensuring that all resources are available and effectively utilized to support the implementation and ongoing operation of the LMS.
- **Support and Communication:** Establishing effective channels for ongoing support and communication to address any issues that arise during and after implementation. Providing clear lines of communication ensures that users have access to assistance and that any problems are promptly resolved.

By focusing on these aspects, the implementation phase aims to deliver a robust and efficient Library Management System that enhances library operations and meets the needs of its users.

6.1.1 SYSTEM MAINTENANCE

All systems need maintenance from time to time. It is important that software be maintained since there are often some residual errors or bugs in the system that must be removed as they are discovered. Many of these issues surface only after the system has been in operation for a longer period of time. When these errors are discovered, they need to be removed, leading to having the software changed. Maintenance involves understanding the effects of change, making the changes to both the code and documents, testing the new parts and retesting the old parts. Support and maintenance on this application is ongoing.

6.1.2 Future Scope : However efficient the system may be, there is always room for improvement in any software. In order to make improvements, the system should be flexible enough for future modifications. New technologies and features will be adapted to make it the best in the future so that this system can act as a complete library management system. The system is going to be a base for creating an e-learning environment.

6.1.3 Backup and Recovery : Administrators have the privilege of backing-up and recovery. An admin is able to make a backup of the database and restore it in case of a system crash or any other need for a reinstall.

5.3 Security : This system with its login feature along with its password encryption provides a secure system for educational use. Two different users can log into the system. One is the admins (librarians) and the other is a normal users (students). Admins have full access to all processes whereas normal users have restricted access only. Here the main objective is to secure is the data of the books and students. As the students have access to the page and cannot view other information apart from the information on books, the student information is safe and is stored in a database which has its own login system which has to be accessed in order to view or modify any information.

CHAPTER 7- CONCLUSION

7.1 CONCLUSION

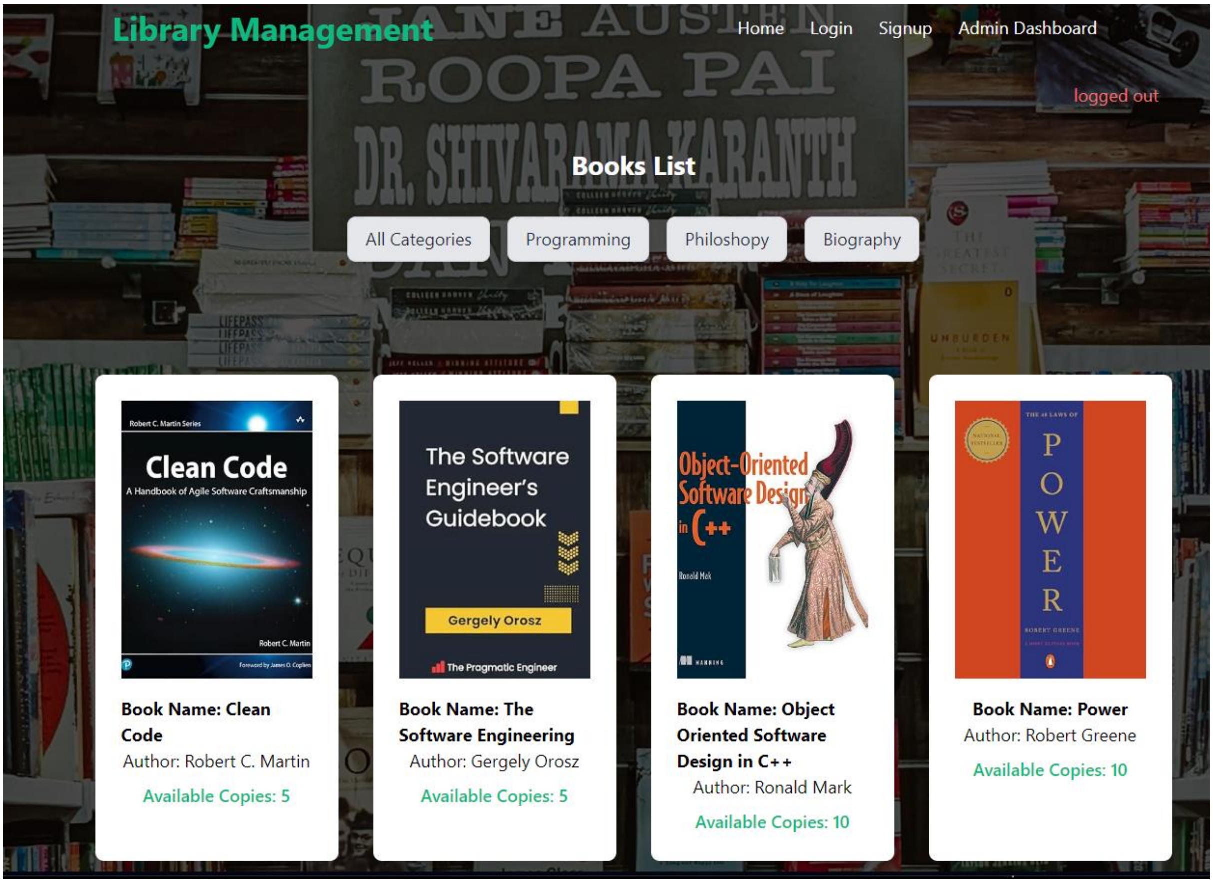
The main goal of this thesis was to explore the process of building a Library Management System with an intent to helping schools. The purpose of the study was to design a Library Management System with a user-friendly interface. The Library Management System was built on an opensource platform; this significantly reduced the development cost. Further operation and maintenance costs are very low. As the project grows, more and more libraries will be integrated and the system with be updated regularly. The system uses a fully graphical interface, which works very efficiently in the management system. The main objectives of the application are to digitise and automate the existing system of manually maintaining the records of the book issue, book return from the student, catalogue and book search. Therefore, the book processing such as issuing, returning, searching will be faster. Records of all the details of the users and library information are stored in database. The users need not visit the library every time to find an item; instead, they can search for items from their own smart device. In the later part of the development process, the programmer would focus on creating an e-learning environment using this management system as the base. The maintenance of the system is also very important, as well as improving the efficiency of procedures by optimizing the database and simplifying the structure of the script.

REFERENCES :

- https://www.geeksforgeeks.com
- ” Entity Relationship Diagram ”, published on visual-paradigm.com, (https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/)
- “Use Case Diagram ”, published on visual-paradigm.com, (https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/)
- “Software Testing ”, published in Wikipedia, (https://en.wikipedia.org/wiki/Software_testing),
- “White-box testing”, published on Wikipedia, (https://en.wikipedia.org/wiki/White-box_testing),
- “ BLACK Box Testing”, published on guru99, (https://www.guru99.com/black-box-testing.html),
- “Unit Testing ”, published on Wikipedia, (https://en.wikipedia.org/wiki/Unit_testing),
- “Integration Testing ”, published on guru99.com, (https://www.guru99.com/integration-testing.html),
- “System Testing”, published on guru99.com, (https://www.guru99.com/system-testing.html)

APPENDIX

Project Screenshot



Books List

Book Name: Clean Code
Author: Robert C. Martin
Available Copies: 5

Book Name: The Software Engineering
Author: Gergely Orosz
Available Copies: 5

Book Name: Object Oriented Software Design in C++
Author: Ronald Mark
Available Copies: 10

Book Name: Power
Author: Robert Greene
Available Copies: 10