PROJECT NAME : LIBRARY MANAGEMENT SYSTEM

GROUP MEMBERS : M.KHALIS, SHAYAN AHMAD

SUBBMITTED TO : SIR USMAN

CLASS : SE 5-1

Library Management System (LMS) Report

M khalis & Shayan Ahmad

1. Introduction

The Library Management System (LMS) is a software application designed to streamline library operations, allowing both librarians and students to interact with the library system effectively. With the growing demands of managing books, user data, and ensuring security, an automated system becomes a necessary solution. This system facilitates book management, student records, and book issuing in an easy-to-use platform. The project was developed in Java to leverage object-oriented programming and efficient data handling, enabling secure user login and role-based access.

2. Objectives

The main objectives of this project are:

- To create an easy-to-use system for librarians to manage books and students to access them.
- To ensure secure login functionality with role-based user access.
- To maintain data persistence for users and book records across sessions.
- To introduce a user-friendly interface that ensures smooth interaction.
- To implement efficient file handling mechanisms to store user data securely.

3. Features

The key features of the LMS are:

User Roles:

The LMS has two main types of users, each with distinct functionalities: Librarians and Students. Each user type has specific permissions and menus, ensuring role-based access control.

- Librarian:
- Can register students into the system.
- Can add, update, or delete books from the library.
- Can manage the issuing and return of books to students.

Student:

- Can register into the system with unique credentials.
- Can view a list of available books in the library.
- Can issue books for a specified duration.
- Can track books they have issued.

Secure Login System

The LMS employs a secure login system to verify user credentials (username and password). Each user is assigned a role, and depending on the role, they have access to different functionalities. The login is managed by a simple comparison of the provided credentials with those stored in the system.

Data Storage

User data (including usernames, passwords, and roles) and book records are stored persistently in a text file (`users.txt`). This ensures that users can log in at any time, and their data will not be lost after closing the application. File handling methods are used to read and write user data, ensuring a simple yet effective solution for persistence.

4. Project Structure

Classes Created

The system follows Object-Oriented Programming (OOP) principles. Here is a description of the classes used in the LMS:

User (Abstract Class):

The base class that holds common properties such as username and password. It also provides methods like login, getUsername, and getRole that are used by the inherited classes.

Librarian:

A class that extends the User class, with additional functionalities related to book management. Librarians can add new books, view current books, and manage the issuing process.

Student:

This class represents the student user, extending the User class. Students can issue books, view the available books, and track their issued books within the system.

Book:

The Book class represents a book in the library. It contains properties such as title, issue date, and return date. The Book class helps store and manage book details in the system.

5. Development Process

Planning

In the planning phase, we outlined the project's key objectives and the main functionalities that needed to be implemented. The decision to store user data in a file format was made to ensure simplicity and persistence without the complexity of a database. The user interface was designed to be simple, with menu options that would provide different functionalities based on user roles.

Implementation

During the implementation phase, the system was divided into components (classes) based on user requirements. Each class was designed to encapsulate specific behaviors: the User class for managing login and roles, the Librarian class for book management, and the Student class for book issuing. Additionally, proper validation was added to ensure secure login and correct date formats for issuing books.

Testing

After development, the system underwent thorough testing, including: testing user registration, login functionality, adding books, issuing books, and viewing available books. File handling was also tested to ensure that data is saved and loaded correctly, allowing users to log in and use the system after restarting the application.

6. Challenges and Solutions

Challenges Faced

During development, several challenges arose: - Managing input validation for dates and ensuring they are correctly formatted.

- Handling file I/O operations to ensure that the system does not lose data after restarting.
- Implementing a secure login system to avoid unauthorized access.

Solutions Implemented

To address these challenges, the following solutions were implemented:

- Java's built-in libraries like `LocalDate` and `DateTimeFormatter` were used for date validation.
- File reading and writing operations were handled with proper exception handling to ensure data integrity.
- A simple login system was created to match user credentials securely.

7. Future Enhancements

Future enhancements for the LMS could include the following: - Implementing a book return feature, allowing students to return books and track overdue items.

- Adding a fine system for late book returns.
- Introducing a Graphical User Interface (GUI) for easier interaction.
- Replacing file-based storage with a relational database (e.g., MySQL) for more scalable data management.
- Incorporating email notifications for overdue books.

8. Conclusion

The Library Management System is a robust and efficient tool for managing a library's operations. Through the implementation of role-based access and secure login, as well as persistent data storage, the system meets the needs of both librarians and students. With future enhancements planned, such as a more sophisticated book return system and database integration, the LMS is well-positioned for growth and improvement.