Project Report

(Group – 11)

Ceylin Oner

Farid Aljanov

Mehmet Alperen Calıs

Md Mashroor Zilan Snigdho

Department of Statistics, Middle East Technical University

STAT295: Object Oriented Programming

Dr. Aysegul Ozkaya Eren

May 20, 2025

Library Management System - Final Project Report

1. Project Title:

Library Management System

2. Overview:

The Library Management System (LMS) is a comprehensive Java-based desktop application developed to efficiently manage library operations. It streamlines activities such as book tracking, member registration, borrowing and returning books, fine calculation, digital resource access, and event management. By applying core object-oriented programming (OOP) principles, this project demonstrates how software can mimic and improve real-world systems.

3. Problem Statement:

Traditional manual methods of managing a library are inefficient, error-prone, and unscalable. Tasks such as tracking book availability, calculating fines, and managing reservations can be time-consuming. This system addresses these issues by providing a robust and modular Java-based solution that automates and simplifies library operations.

4. Project Objectives:

- Implement a Java application based on core OOP principles.
- Support full lifecycle of library resource management (books, digital materials, members).
- Provide functionalities for borrowing, returning, reserving books, and handling overdue fines.
- Design a user-friendly GUI for interaction.
- Ensure code quality through proper encapsulation, inheritance, abstraction, and polymorphism.
- Deliver a UML class diagram representing system architecture.
- Conduct and document comprehensive test cases.

5. Scope of the Project:

The system includes the following modules:

- **Book Management:** Add, update, delete, search, and review books.
- **Member Management:** Register, update, remove members, and track their activities.
- **Borrowing/Returning:** Record loans and returns, manage due dates.
- **Fine Calculation:** Automatically compute fines for overdue items.
- **Reservation System:** Allow members to reserve unavailable books.
- **Digital Resources:** Include e-books and PDFs as non-physical assets.
- **Library Events:** Schedule and manage events.
- **GUI Interface:** Implemented using console-based terminal interactions.

6. Key Classes and Their Responsibilities:

- **Library:** Central coordinator for managing books, members, and events.
- **Book:** Represents a physical book, tracks stock, and enables borrowing.
- **Category:** Categorizes books and digital materials.
- **LibraryResource (Abstract):** Base class for all resources.
- **DigitalResource:** Represents digital content.
- **Person (Abstract):** Base class for people in the system.
- **Member:** Library user who borrows/reserves books and pays fines.
- **Librarian:** Admin responsible for inventory and member oversight.
- **Loan:** Records book borrowing instances.
- **Reservation:** Holds reservation details for future checkouts.
- **Fine:** Calculates penalties for late returns.
- **Review:** Enables members to leave feedback.
- **LibraryEvent:** Manages scheduled events like book fairs.

7. Design Principles Applied:

- **Encapsulation:** Data is protected through private fields and public accessors.
- **Abstraction:** Abstract classes hide complexity and define common behaviors.
- **Inheritance:** Shared behaviors and attributes are inherited (e.g., Member and Librarian from Person).
- **Polymorphism:** Method overriding enables diverse behaviors across subclasses.

- **Exception Handling:** Try-catch-finally blocks and custom exceptions manage invalid operations.
- **Extensibility:** Easily accommodates future features or entities.

8. Tools & Technologies Used:

- **Programming Language:** Java
- **Development IDE:** VSCode
- **GUI Framework:** None. Terminal/Console-based interface used.
- **Diagram Tools:** LucidChart, Mermaid

9. Challenges and Solutions:

- **Efficient Data Handling:**
  - *Solution:* Used ArrayLists and HashMaps for in-memory data storage.
- **Handling Invalid Input or Operations:**
  - *Solution:* Implemented try-catch blocks and user-friendly error messages.
- **Designing the GUI:**
  - *Solution:* Developed simple interfaces using console or terminal-based interactions.

10. UML Class Diagram:

A comprehensive UML diagram was created to visualize class relationships, inheritance hierarchies, and method responsibilities. It includes all major entities and their interactions. The diagram can be accessed at: UML Diagram Link

11. Testing Scenarios:

1. Add a new book and verify its presence in the system.
2. Register a new member and ensure unique identification.
3. Borrow a book and verify updated stock.
4. Return a book and test fine calculation for overdue.
5. Attempt to borrow an unavailable book.
6. Reserve a book and check the reservation list.
7. Add a digital resource and validate access.

12. Project Deliverables:

- Fully functional Java-based Library Management System.
- UML class diagram of the system architecture.
- README with installation and usage instructions.
- A ChatGPT/DeepSeek (or any other AI tool) usage report
- Final project report documenting all phases.

13. Team Members:

Name	**Department**
Ceylin Oner	Statistics
Farid Aljanov	Statistics
Mehmet Alperen Calıs	Statistics
Md Mashroor Zilan Snigdho	Mechanical Engineering

14. Conclusion:

The Library Management System successfully simulates a real-world library environment using object-oriented design. It showcases all four pillars of OOP. Through this project, the team gained valuable experience in collaborative development, modular design, and practical Java programming.