TECHNICAL DOCUMENTATION ON CAMPUS LIBRARY SYSTEM

This paper is made to complete the final assignment of the Object Oriented Programming course

supervised by: Ir. Galih Wasis Wicaksono, S.Kom., M.Cs

Compiled by:

Amalia Sanyoto	202410370110038
Ani Seila Nanda Putri	202410370110191
Farel Faiza	202410370110446

UNIVERSITAS MUHAMMADIYAH MALANG

FAKULTAS TEKNIK

PRODI INFORMATIKA

2025

LIST OF CONTENT

CHAPTER I

INTRODUCTION

1.1 Background

The development of software applications using Object-Oriented Programming (OOP) principles has become essential in creating modular, scalable, and maintainable systems. This final project report presents the design and implementation of a *Campus Library System*, a desktop-based application developed as part of the Object-Oriented Programming course. The system is designed to assist campus libraries in managing their daily operations, including book inventory management, borrowing and returning processes, and user management.

The project emphasizes key OOP concepts such as encapsulation, inheritance, polymorphism, and abstraction. By applying these principles, the system was structured into reusable and organized components, enabling easier development and future maintenance. The application also aims to provide a user-friendly interface for both librarians and students, improving the efficiency of library services within the campus environment.

This report outlines the system requirements, design methodology, class structures, key functionalities, and testing processes. It also reflects on the challenges encountered during development and the lessons learned in applying OOP in a real-world project context.

1.2 Objective.

1. To design and develop a Campus Library System using Object-Oriented Programming principles that can efficiently manage library operations such as book borrowing, returning, and catalog management.
2. To implement core OOP concepts such as encapsulation, inheritance, polymorphism, and abstraction in a real-world application.
3. To provide an intuitive and user-friendly interface for both librarians and students, allowing for smooth interaction with the system.
4. To improve the efficiency and accuracy of library data management by replacing or complementing manual processes with a computerized system.

5. To gain practical experience in software development, including system design, coding, debugging, and testing, as part of the learning outcomes of the Object-Oriented Programming course.
6. To integrate JavaFX for graphical user interface development and MySQL for data storage and management

1.3 Scope

- **Member registration.**
  Registration interface for new members or new students.
- **Member and admin login.**
  Log in menu for member and admin.
- **Book borrowing and returning**
  Provides a borrowing and returning menu for student to manage their book transaction.
- **Fine payment**
  Provides a payment method for overdue return penalties.
- **Manage member**
  Provides a menu for admin to manage students, adding students, deleting student data and configuring student book statistics.
- **Manage book (CRUD)**
  Provides a CRUD (Create, Read, Update, Delete) menu for managing book records

1.4 Target Audience

1. **Student**
   Students will use the system to borrow and return books, view borrowing records, and process fine payments in case of overdue returns.
2. **Librarian**
   Librarians will utilize the system to perform CRUD operations, create, read, update, and delete on both book and member data.
3. **System Administrator**
   System Administrators will oversee user accounts, maintain system integrity, and ensure the overall smooth operation of the system.

1.5 Literatur Review

1.5.1 Object Oriented Programming (OOP)

Object-Oriented Programming (OOP) is a programming paradigm that focuses on the use of objects to represent data and methods. It allows developers to structure code in a more modular and reusable way. The four main principles of OOP are:

1. Encapsulation

   The bundling of data and methods that operate on the data, restricting direct access to some of the object's components to protect data integrity.

2. Inheritance

   The mechanism by which one class can inherit the attributes and methods of another, promoting code reuse and hierarchy.

3. Polymorphism

   The ability of different classes to respond to the same method call in a way appropriate to their type, improving flexibility in code design.

4. Abstraction

   The process of hiding complex implementation details and showing only the essential features of the object

These principles enable the creation of robust and scalable applications. In this project, OOP principles are applied to structure the system into reusable classes and components.

1.5.2 Librarian Management System

A Library Management System is an application designed to automate and manage the daily operations of a library. These operations include book cataloging, borrowing and returning transactions, user registrations, and report generation. A digital library system improves efficiency by reducing manual work and minimizing errors in record-keeping. Features such as book availability tracking, due date management, and user access control are commonly found in modern systems.

1.5.3 User Interface Design

A well-designed user interface (UI) plays a crucial role in enhancing user experience. The project utilizes JavaFX, a GUI toolkit for Java, to create an interactive and user-friendly interface. Important UI design principles include

consistency, clarity, responsiveness, and accessibility. In line with Human-Computer Interaction (HCI) practices, the interface is designed to be intuitive for both librarians and students, ensuring ease of navigation and clear visual feedback.

1.5.4 Software Development Life Cycle (SDLC)

The Software Development Life Cycle (SDLC) refers to the process followed during the development of a software product. This project adopts the Waterfall model, which consists of sequential stages:

1. **Requirement Analysis -** Identifying and documenting user need
2. **System Design -** Creating the architecture and class structure
3. **Implementation -** Writing and compiling code
4. **Testing -** Verifying system functionality and fixing error
5. **Deployment and Documentation -** Delivering the final system with appropriate documentation

Following this structured approach helped the team ensure that each stage of development was completed before moving to the next.

1.5.6 Technologies Used

This project uses a combination of programing tools and technologies:

1. **Java**

   The main programming language, chosen for its strong support for OOP and cross-platform capabilities.

2. **JavaFX**

   Used to build the graphical user interface, offering components like buttons, tables, input fields, and layout managers.

3. **MySQL**

   For persistent data storage, allowing the system to manage user and book records reliably.

4. **IDE Tools**

   Such as IntelliJ IDEA or Eclipse, to streamline development through features like syntax highlighting, debugging, and version control integration

CHAPTER II

SYSTEM OVERVIEW

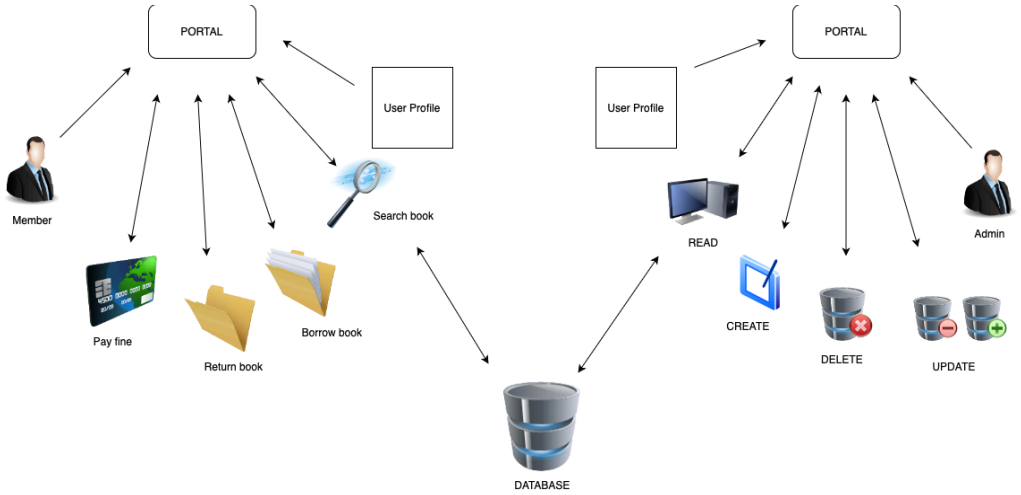
2.1 System Design Overview

The Campus Library System is developed to facilitate and optimize key library operations, including book borrowing, returns, and other administrative tasks, ensuring greater efficiency and accessibility for both students and staff.

1. **Key Design Principle**
    - **Modularity -** Separates frontend, backend, and database for maintainability
    - **User Centric UI -** Well-designed interfaces tailored to meet the needs of students, librarians, and system administrators, ensuring both functionality and ease of use
2. **Core Modules**

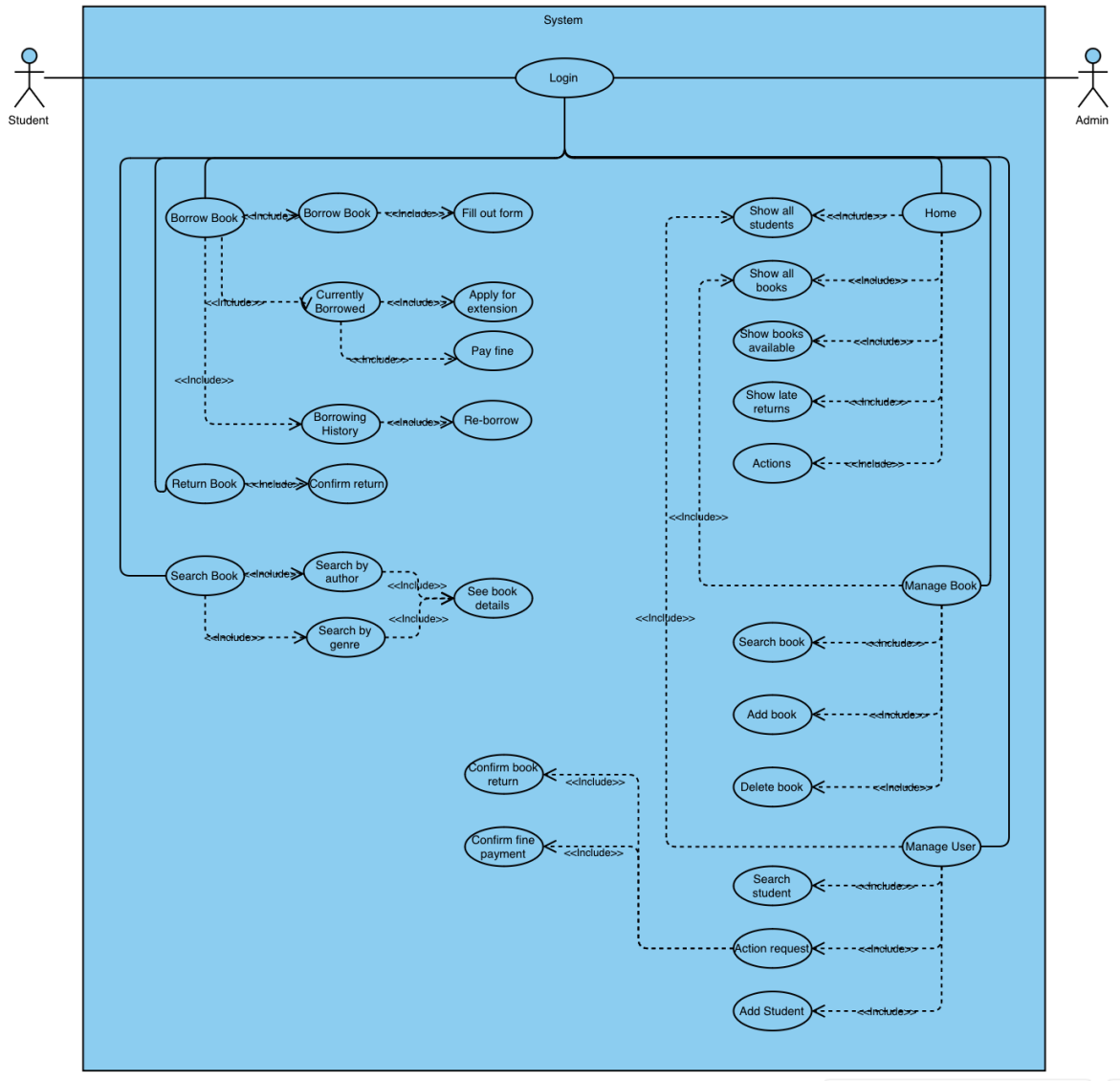
Management User	Registration, login, logout, delete user data, add user data, display user
Management Book	Delete book, add book, manage book status, update book, display book
Borrowing Book	Check out
Returning Book	Due date
Searching Book	Searching and filtering book
Fine management	Fine payment, fine calculation

2.2 System Architecture

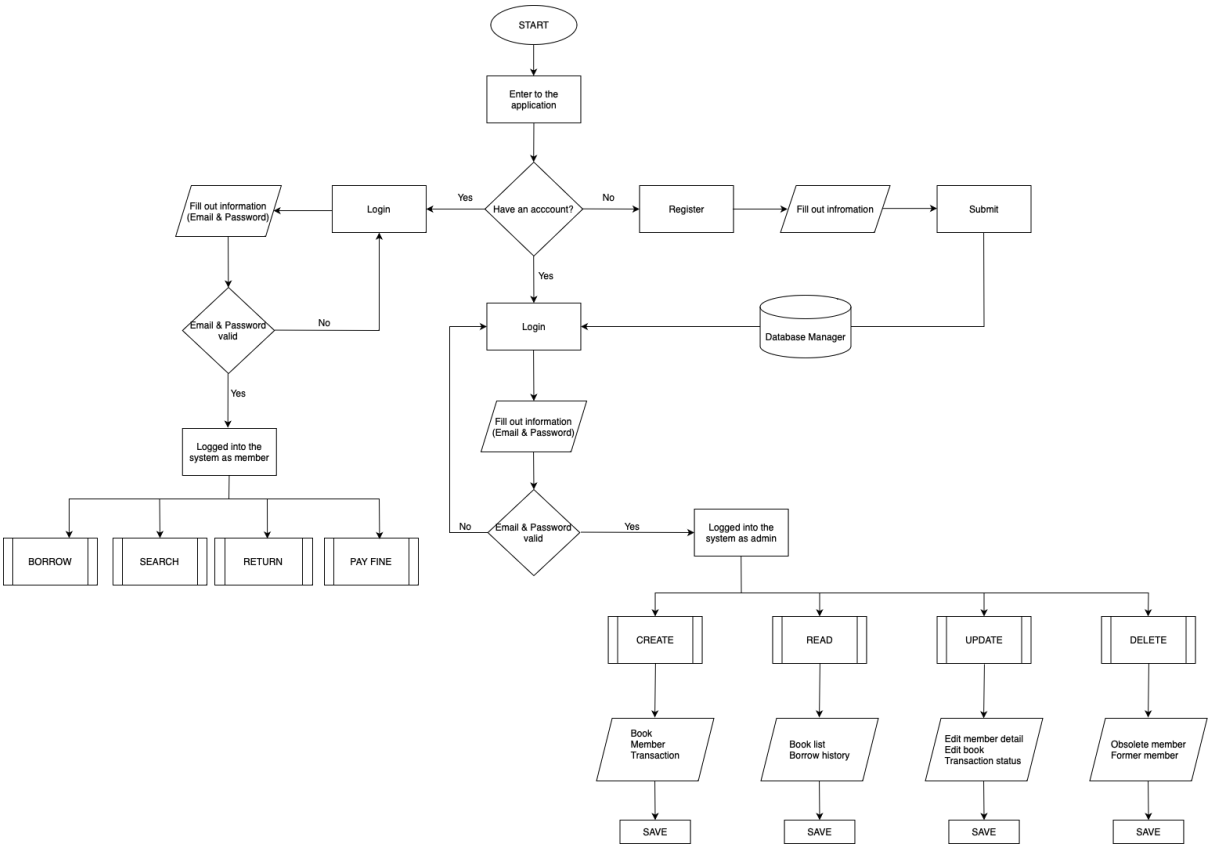


This diagram illustrates the main interaction flow between system users (Members and Admins) with the campus library system via the Portal.

2.3 Use Case Diagram



2.4 Flowchart Diagram



2.5 Technology Stack

- **Frontend**

  Using JavaFX, building interactive and responsive user interfaces
- **Backend**

  Using Java to handle business logic, authorization, data processing, and communication with the database.
- **Database**

  Using SQL, to store, manage, and retrieve application data

CHAPTER III

FUNCTIONAL AND NONFUNCTIONAL REQUIREMENT

3.1 Functional Requirement

3.1.1 User Role and Permission

Member	- Borrow book - Return book - Search book - Pay fine - Login - Register
Admin	- Add memberdata - Delete member data - Add book - Delete book - Display book - Display member

3.1.2 Core Features

- **Management book**

  The admin can add books, delete them, view book displays, and manage their status.

- **Management user**

  The admin can manage users by adding new users, removing them, and viewing user information displays

- **Fine payment**

  Users can make fine payments for overdue book returns.

- **Searching**

  Users can perform book searches and apply specific filters or criteria to refine the results

- **Borrowing and returning**

  Users can borrow books, check them out, and return them through the system.

3.2 Non Functional Requirement

- **Performance**

  The system respond within 2 second
- **Reliability**

  System maintenance accurate transaction logs and data integrity
- **Usability**

  System interface is user friendly and intuitive for both roles
- **Scalability**

  System allow additional book and users as needed
- **Security**

  Login credentials are securely handled.

CHAPTER IV

TESTING

4.1 White Box Testing

No	Test case	Code Testing	Input	Expected Output	Result
1.	Login Logic	-	Input choice button	Login system to admin or login system to user	Success
2.	Admin Login Logic	-	Valid Username : “admin” Password : “admin”	Successfully login to dashboard admin	Success
3.	User Login Logic	-	NIM : “202410370 110494” Password : “1234”	Successfully login to dashboard student or dashboard user	Success
4.	Search Student Logic	-	Input student NIM or name	Successfully display student data	Success
5.	Search Book Login	-	Book name, or author, or genre	Successfully display book data	Success
6.	Borrow Book Logic	-	Input data, name, NIM, book name, book code, borrowing date, returning date	Successfully display message borrowing book	Success
7.	Add Book Logic	-	Input data book name, author, ISBN, book code, release date, total	Successfully display message, restore data to database	Success

			book, genre		
8.	Add User / Student Logic	-	Input data name, email, phone number, address, faculty, study program	Display message successfully added	Success
9.	Delete User Logic	-	Click delete and confirm button	Display message delete successfully	Success

4.2 Black Box testing

No	Test case	Test Step	Input	Expected Output	Result
1.	Admin Login Authentication	Enter valid admin credential	Username : “admin” Password : “admin”	System successfully logs in and redirects to the admin dashboard	Success
2.	Admin Login Authentication	Enter invalid admin credential	Username : “Salah” Password : “Salah”	System denies access and displays an error message (e.g., "Invalid username or password")	Success
3.	User Login Authentication	Enter valid NIM/email and password	NIM : “202410370 110494” Password : “1234”	System successfully logs in and redirects to the student dashboard	Success
4.	User Login Authentication	Enter invalid NIM/email and password	NIM : “wrong” Password : “apalah”	System denies access and displays an error message (e.g., "Invalid NIM/email or password")	Success

5.	User Registration	Enter valid data registration	Valid Data : Name, email, phone number,, address, faculty, program study	System successfully registered and redirects to the login page	Success
6.	User Registration	Enter invalid data registration	Invalid Data : Name, email, phone number,, address, faculty, program study	System denies	Success
7.	Borrowing Book	Entering valid borrowing form data	Valid Data : Name, NIM, book title, book code, return date, borrowing date	System successfully and and displaying next step message	Success
8.	Borrowing Book	Entering invalid borrowing form data	Invalid Data : Name, NIM, book title, book code, return date, borrowing date	System denies	Success
9.	Returning Book	Click returning button	-	Successfully and displaying text notification	Success
10	Fine Payment	Click pay fine	-	Successfully and display text messages waiting for admin confirmation.	Success
11	Searching Book	Enter valid data searching book	Valid data : book title, genre, or author	Display available book	Success

12	Searching Book	Enter invalid data searching book	Invalid data : book title, genre, or author	Display text : “No match data found”	Success
13	Accept Notification	Click confirm button	-	Successfully confirmed, change in book status or update in users fine amount	Success
14	Add Student	Enter valid student data	Valid data : Name, NIM, email, address, phone number, password, faculty, program study	System successfully registered and display success message	Success
15	Delete Student Data	Click delete button	-	Display confirmation message, successfully deleted	Success
16	Add Book	Enter valid book data	Valid data : book title, book kode, ISBN, author, genre, amount, release date, book code or ID	Display successfully message	Success
17	Delete Book	Click delete button	-	Display confirmation message, successfully deleted	Success

CHAPTER V

USER INSTRUCTION

5.1 Student Instruction

5.1.1 Registration

- Open the application
- Click the registration menu
- Enter the required data
- Click submit / login
- Registration completed successfully

5.1.2 Login

- Open the application
- Select the login method for user or student
- Enter the NIM/email used during registration
- Enter the password used during registration
- Click the login button
- Login successful

5.1.3 Edit Profile

- Open the application
- Log in to the application
- Go to the profile menu. Click "Edit Profile"
- Enter the data you want to change
- Click "Save"
- Data successfully updated

5.1.4 Borrow Book

- Open the application
- Log in to the application
- Go to the "Borrow Book" menu
- Click the "Borrow Book" button
- Fill out the book borrowing form
- Click "Submit Borrow Request"
- Book borrowing completed successfully

5.1.5 Return Book

- Open the application
- Log in to the application
- Go to the "Return Book" menu
- Input borrowed book id
- Click "Confirm Return"
- Book return completed successfully

5.1.7 Search Book

- Open the application
- Log in to the application
- Go to the book search menu
- Enter the book title, author, or genre
- Book search completed successfully

5.1.8 Pay Fine

- Open the application
- Log in to the application
- Go to the book borrowing menu
- Click the "Currently Borrowed" menu
- Check the bottom section to see the fine amount
- Click "Confirm Payment"
- Payment confirmation completed successfully

5.2 Admin Instruction

5.2.1 Login

- Open the application
- Select "Login as Admin"
- Enter username or email
- Enter password
- Click the login button
- Login completed successfully

5.2.2 Manage User

- **Display All Students**
    - Open the application

    - Log in to the application
    - Select the "Manage User" menu
    - Select the "Show All Students" option
    - All students displayed successfully
- **Add User**
    - Open the application
    - Log in to the application
    - Select the "Manage User" menu
    - Select “Add Student” menu
    - Fill in the required information
    - The student was successfully added
- **View Student Detail**
    - Open the application
    - Log in to the application
    - Select the "Manage User" menu
    - Select “Show All Students” menu
    - Click “Detail” button
    - Student detail displayed successfully
- **Delete Student**
    - Open the application
    - Log in to the application
    - Select the "Manage User" menu
    - Select “Show All Students” menu
    - Click “Detail” button
    - Click “Delete” button
    - Deleted student data successfully
- **Search Student**
    - Open the application
    - Log in to the application
    - Select the "Manage User" menu
    - Select “Search Student” menu
    - Input NIM or student name
    - Search student process successfully

5.2.3 Manage Book

- **Display All Books**
    - Open application
    - Log in to the application
    - Select “Manage Book” menu
    - Select “Tampilkan Semua Buku” menu
    - All books displayed successfully
- **Add Books**
    - Open application
    - Log in to the application
    - Select “Manage Book” menu
    - Select “Add Book” menu
    - Fill in the required information
    - The book was successfully added
- **View Book Detail**
    - Open application
    - Log in to the application
    - Select “Manage Book” menu
    - Select “Tampilkan Semua Buku” menu
    - Select “Detail” button
    - Book detail displayed successfully
- **Delete Book**
    - Open the application
    - Log in to the application
    - Select the "Manage Book" menu
    - Select “Show All Book” menu
    - Click “Detail” button
    - Click “Delete” button
    - Click “Confirmation” button
    - Deleted book data successfully