BINUS UNIVERSITY
BINUS INTERNATIONAL

Final Project Report
(Group Work)

Student Information:

Surname:	**Given Name:**	**Student ID Number:**
Teguh	Ari Jaya Teguh	2702403996

Course Code : COMP6699001 **Course Name** : Object Oriented Program
Class : L2AC **Lecturer** : Jude Joseph Lamug Martinez MCS

Type of Assignments : Final Project Report
Due Date : 20 June 2024 **Submission Date** : 02 June 2024

Table of Contents

1. Project Specification

The primary objective of this project is to develop a comprehensive Library Management System (LMS) to address and resolve the various issues associated with poor management within a library environment. The current system exhibits significant inefficiencies and needs more organization and proper data handling mechanisms. These shortcomings lead to operational inefficiencies, making it difficult to manage the catalog of books, register and monitor member activities, and handle borrow and return operations effectively.

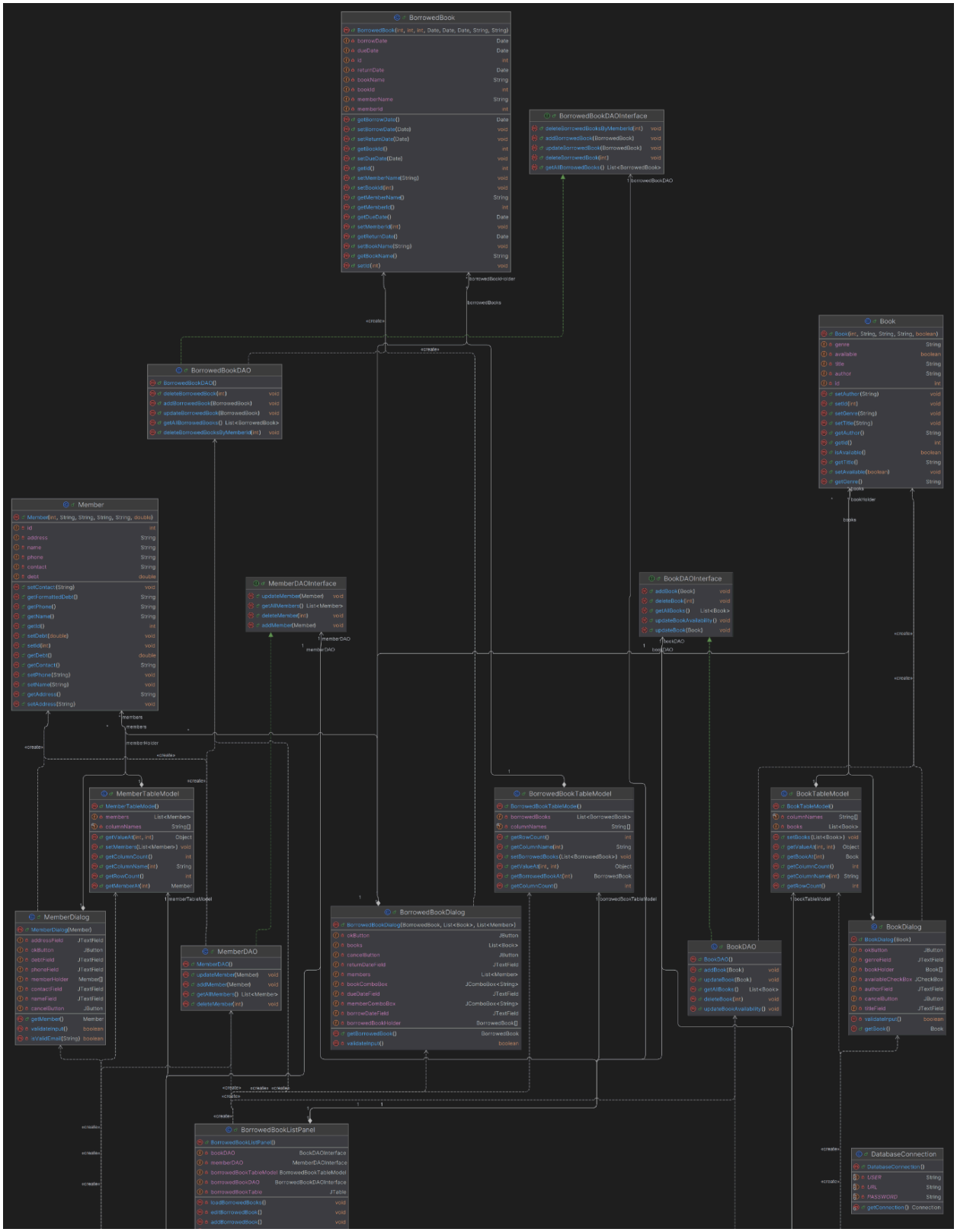
The proposed solution is designed to streamline these processes by developing a user-friendly application that integrates robust database management features. By leveraging modern database technologies, the system will ensure seamless data management and enhanced operational efficiency. This project seeks to provide a holistic approach to library management, encompassing various critical functions such as book inventory management, member registration and tracking, transaction recording, and reporting capabilities.

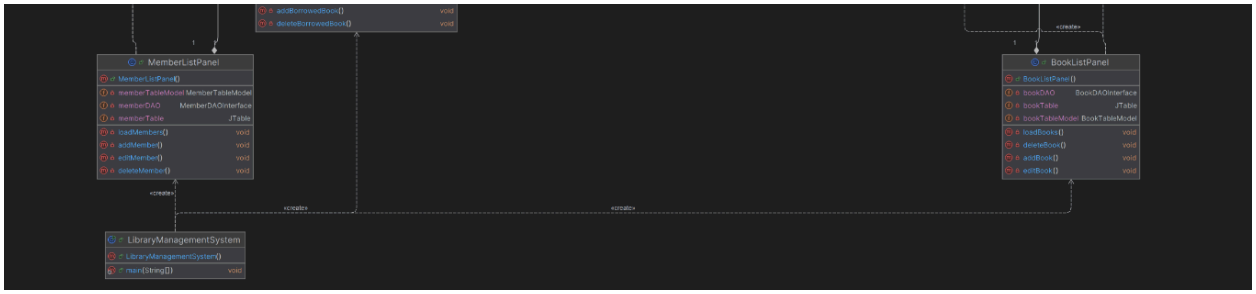
The new LMS will incorporate an intuitive graphical user interface (GUI) to facilitate ease of use for library staff and patrons. The system will allow for efficient cataloging of books, automated member management, and simplified borrow/return processes, thereby reducing manual workload and potential errors. Additionally, the LMS will offer advanced search and reporting functions, enabling quick retrieval of information and comprehensive data analysis.

Overall, this project aims to transform the library management experience by introducing a well-organized, efficient, and user-friendly system that addresses the limitations of the existing setup. By implementing this LMS, libraries can expect improved operational efficiency, better resource management, and enhanced user satisfaction.

2. Solution Design

2.1. Class Diagram





The class diagram for the Library Management System includes the following main components:

1. DatabaseConnection: Manages the connection to the database.
2. Book: Represents a book in the library with attributes like id, title, author, genre, and availability.
3. Member: Represents a library member with attributes like id, name, address, phone, contact, and debt.
4. BorrowedBook: Represents a borrowed book with details about the borrowing and returning dates, and the associated member and book.
5. BookDAO: Data Access Object for managing CRUD operations related to books.
6. MemberDAO: Data Access Object for managing CRUD operations related to members.
7. BorrowedBookDAO: Data Access Object for managing CRUD operations related to borrowed books.
8. BookDialog: GUI component for adding or editing book details.
9. MemberDialog: GUI component for adding or editing member details.
10. BorrowedBookDialog: GUI component for adding or editing borrowed book details.
11. BookListPanel: GUI component displaying a list of books with options to add, edit, delete, and refresh.
12. MemberListPanel: GUI component displaying a list of members with options to add, edit, and delete.
13. BorrowedBookListPanel: GUI component displaying a list of borrowed books with options to add, edit, and delete.

2.2. Discussion

- Database Connection

  The `DatabaseConnection` class is crucial in managing the connection between the application and the MySQL database. It employs a singleton pattern to ensure that only one instance of the database connection exists throughout the application lifecycle. This approach not only minimizes the overhead associated with establishing multiple connections but also guarantees that all database operations are executed using a consistent and reliable connection. By centralizing the connection logic, we also facilitate easier maintenance and troubleshooting of database connectivity issues.

- Data Access Objects (DAO)

  The DAO pattern is a core design principle used in this application. DAOs such as `BookDAO`, `MemberDAO`, and `BorrowedBookDAO` serve as intermediaries between the application and the database. They encapsulate all the CRUD (Create, Read, Update, Delete) operations, ensuring that the business logic is decoupled from direct database access. This abstraction layer allows for flexibility in modifying the database interaction code without impacting the business logic. For instance, changes to SQL queries or database schema can be managed within the DAO classes without requiring changes to the rest of the application. Additionally, DAOs improve code readability and maintainability by providing a clear structure for database operations.

- Graphical User Interface (GUI):

  The application features a robust GUI designed using Java Swing. Key components include:

  - ➔ BookDialog: A form for adding and editing book details. It ensures that all necessary fields (title, author, genre, availability) are provided and valid.
  - ➔ MemberDialog: A form for managing member details, ensuring that the phone number is numeric and the email is correctly formatted.
  - ➔ BorrowedBookDialog: A form for managing the borrowing and returning of books, including selecting available books and members.
  - ➔ BookListPanel: Displays a list of books with options to add, edit, delete, and refresh. It dynamically updates to reflect changes in book availability.
  - ➔ MemberListPanel: Displays a list of members with similar CRUD functionalities.
  - ➔ BorrowedBookListPanel: Manages the borrowing records, allowing the user to keep track of borrowed books and their return statuses.

  These GUI components are designed to be user-friendly and intuitive, ensuring that library staff can manage the system with minimal training.

- Validation

  Validation is a critical aspect of the application, ensuring data integrity and preventing erroneous data entry. The `MemberDialog` class, for instance, includes validation logic to ensure that phone numbers are numeric and contacts are valid email addresses. This reduces the risk of data corruption and ensures that all information stored in the database adheres to expected formats.

- Refresh Mechanism:

  The `BookListPanel` includes a refresh button, a vital feature for ensuring that the displayed data is always up-to-date. When books are borrowed or returned, their availability status changes. The refresh button allows the user to reload the list of books, ensuring that the latest availability status is reflected in the UI. This is particularly important in a dynamic environment like a library where the status of books can change frequently.

- Foreign Key Constraints:

  The application employs foreign key constraints to maintain referential integrity within the database. For example, when a member is deleted, any associated borrowing records are also deleted to ensure that there are no orphaned records in the `BorrowedBooks` table. This cascading delete mechanism is handled within the DAO classes, ensuring that data consistency is maintained without requiring manual intervention. This approach simplifies database management and ensures that all related records are appropriately managed.

- Error Handling:

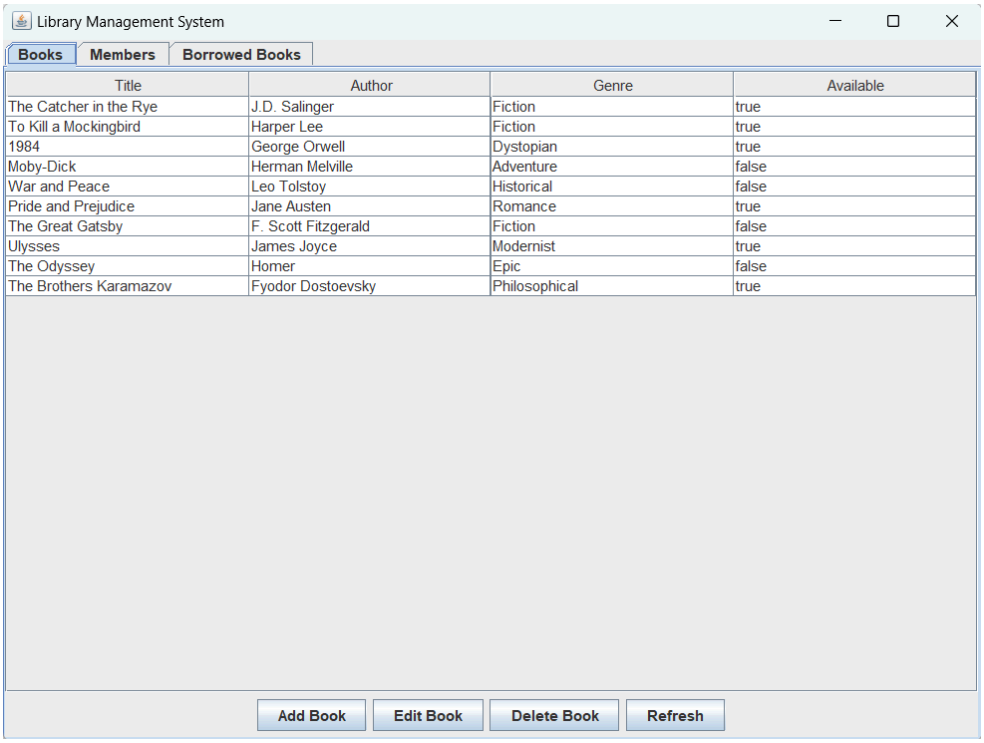
  Comprehensive error-handling mechanisms are implemented across the application to ensure robustness. The DAOs handle SQL exceptions and provide meaningful error messages to the user. This approach helps in diagnosing issues quickly and ensures that the application can recover gracefully from unexpected errors.

- Extensibility:

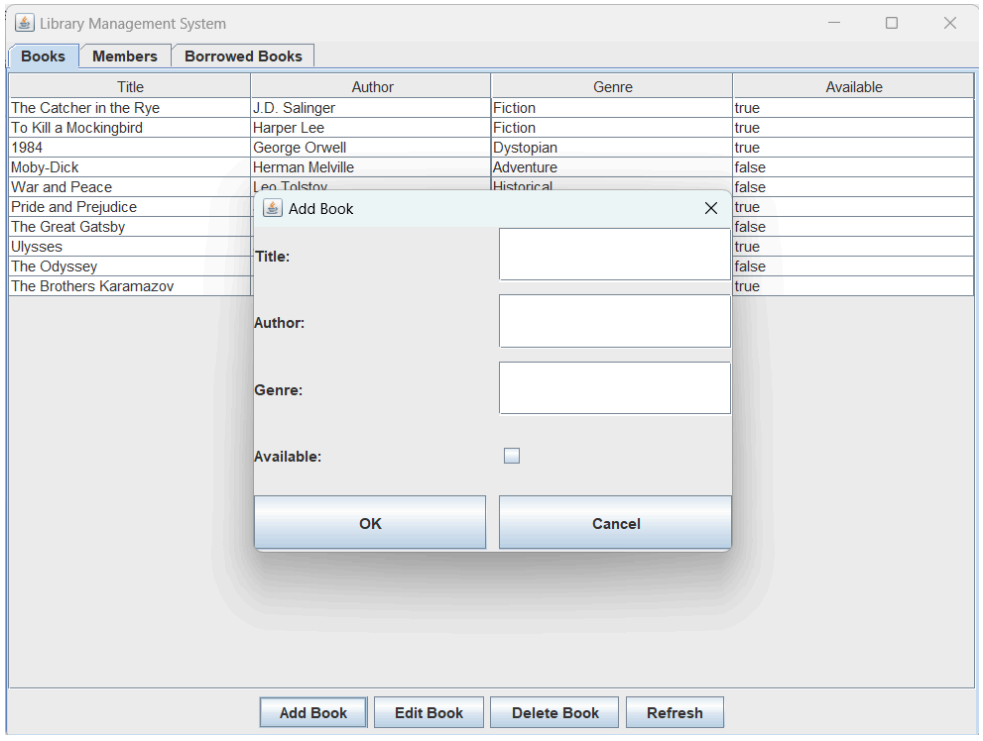
  The design of the application ensures that it is easily extensible. New features or modifications can be added with minimal disruption to existing functionalities. For instance, additional fields can be added to the `Book` or `Member` classes, and the corresponding DAOs and GUIs can be updated to handle these new fields.

3. Screenshot of Application

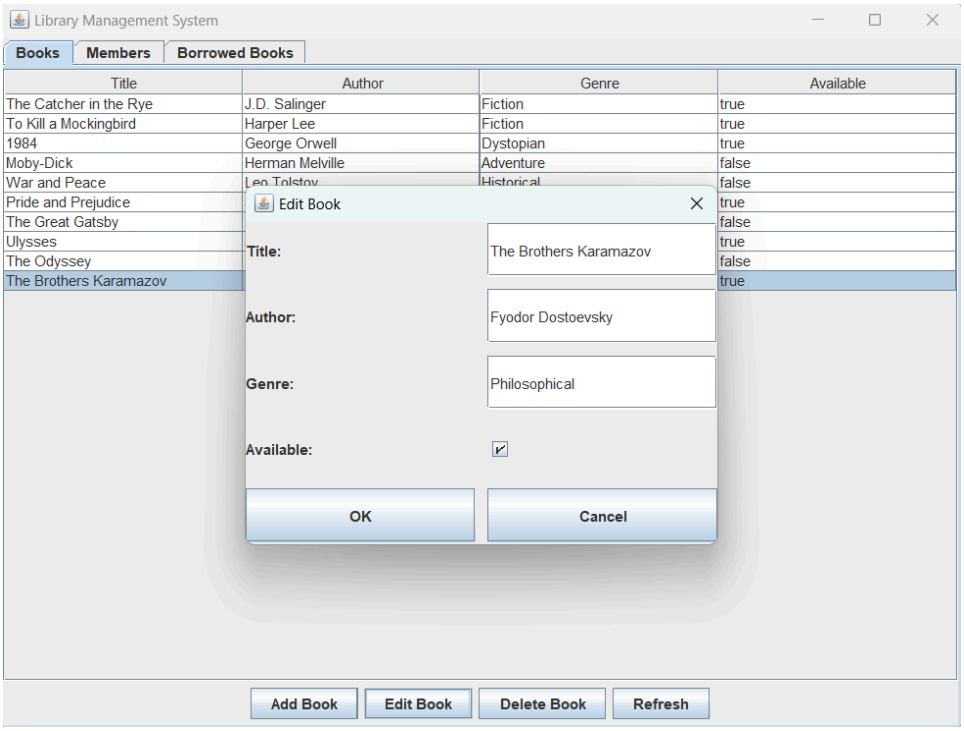
3.1. Book Window



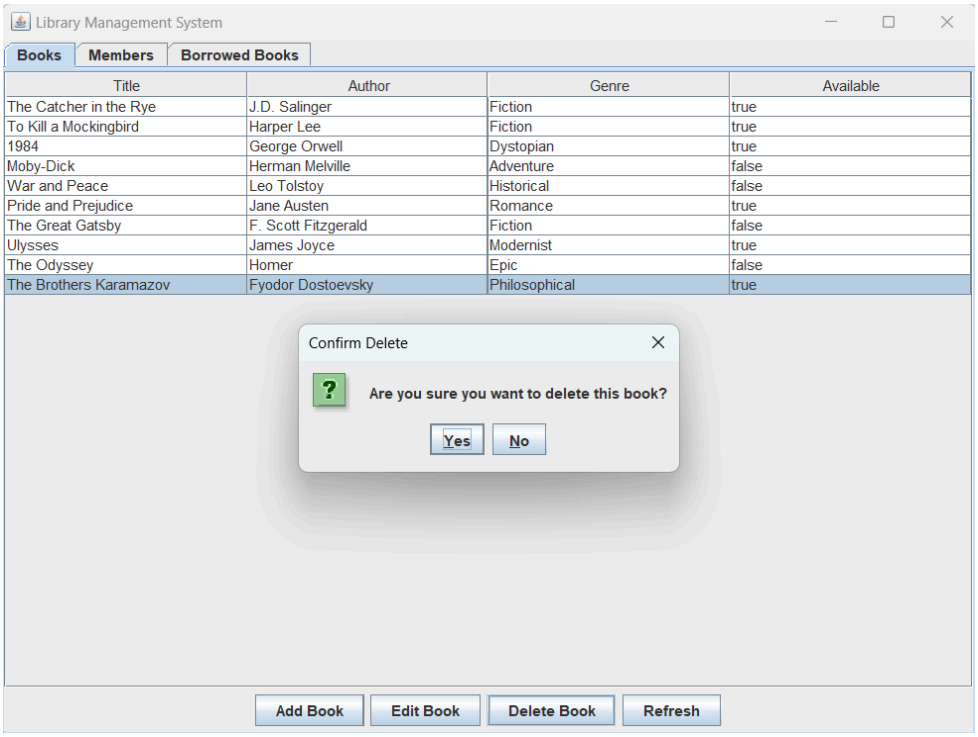
Adding New Book



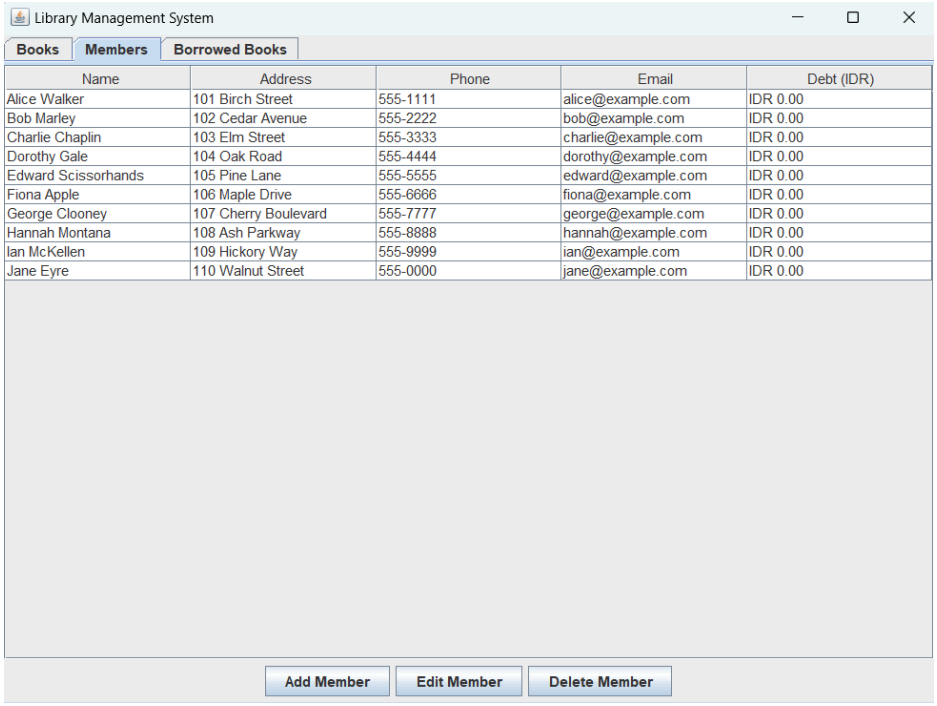
Editing Book



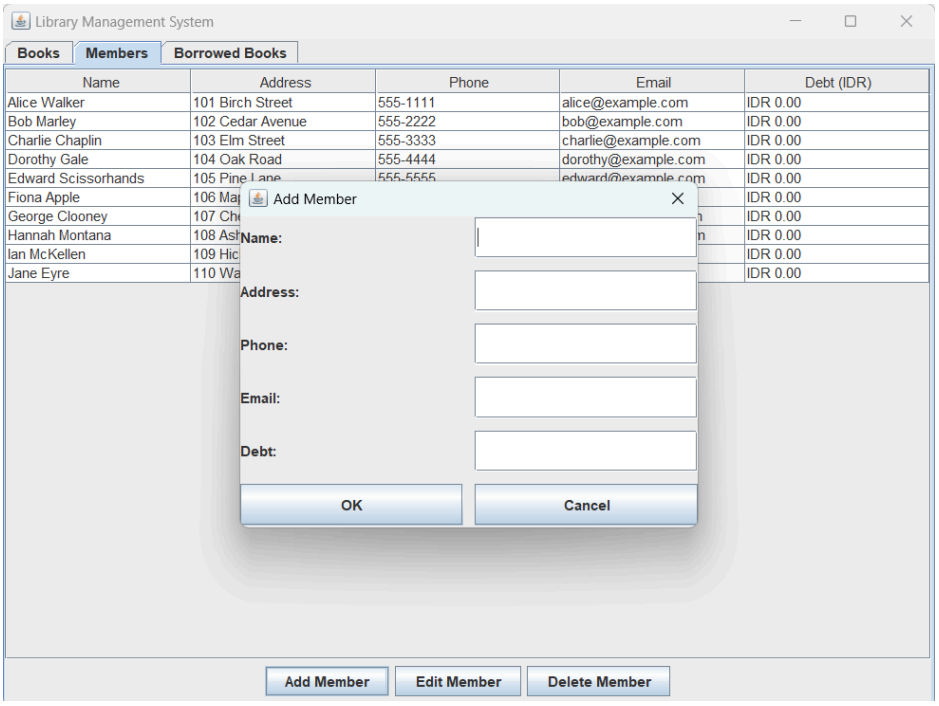
Deleting Book



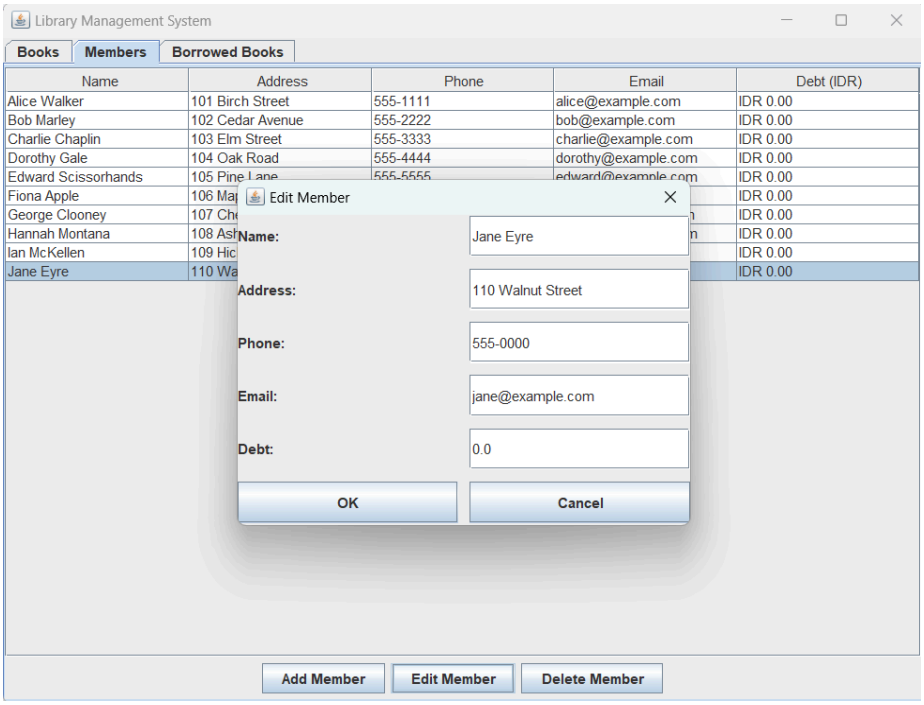
3.2. Member Window



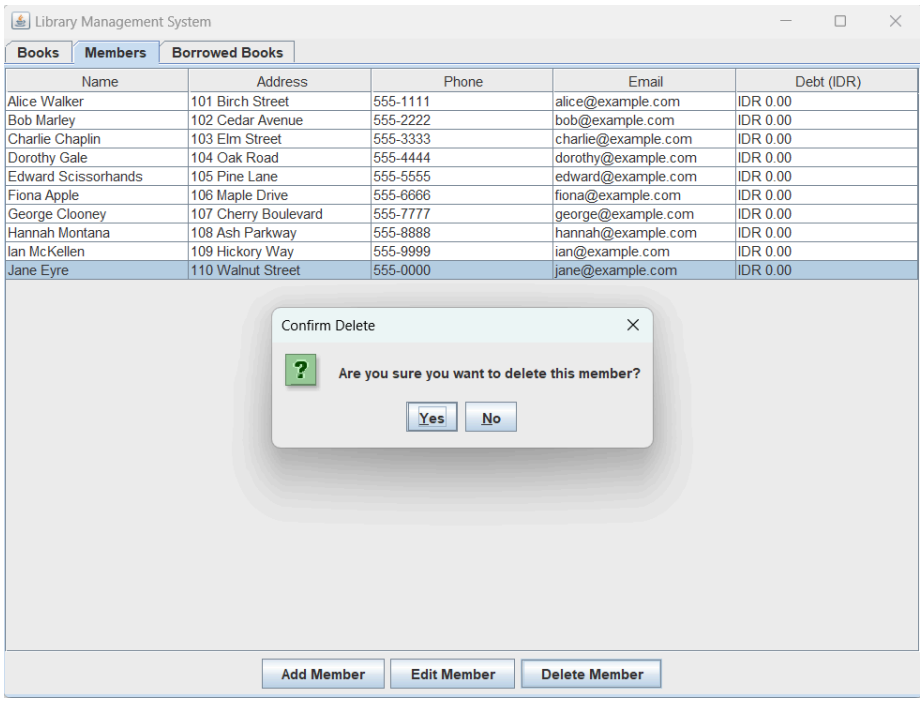
Adding New Member



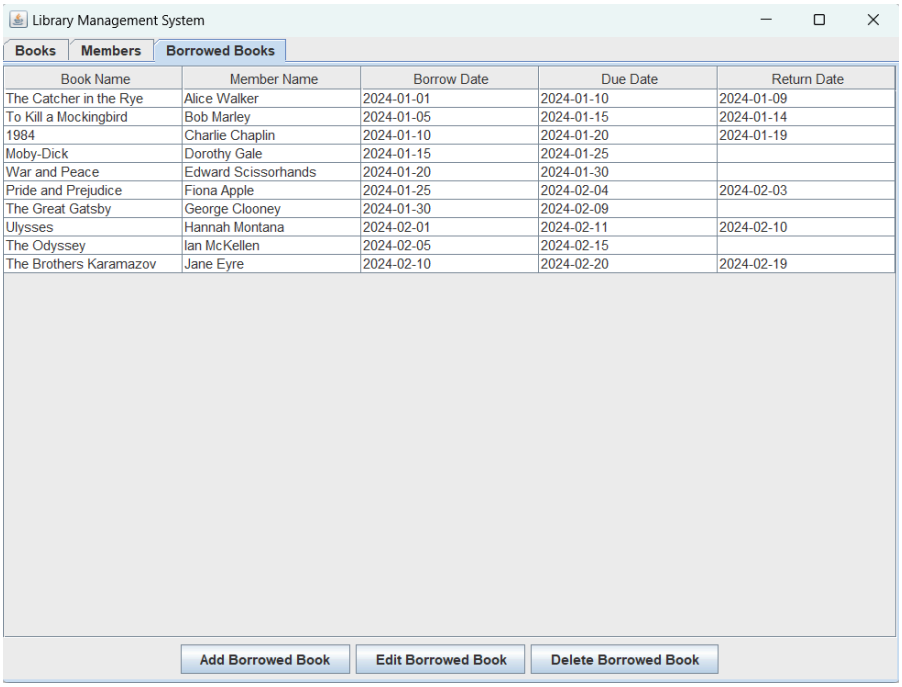
Editing Member



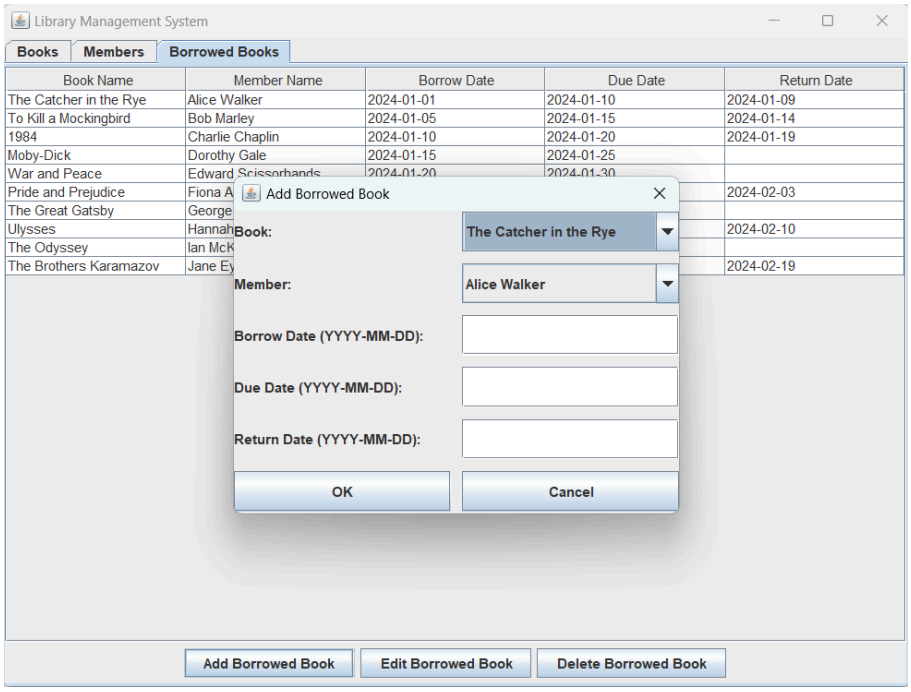
Deleting Member



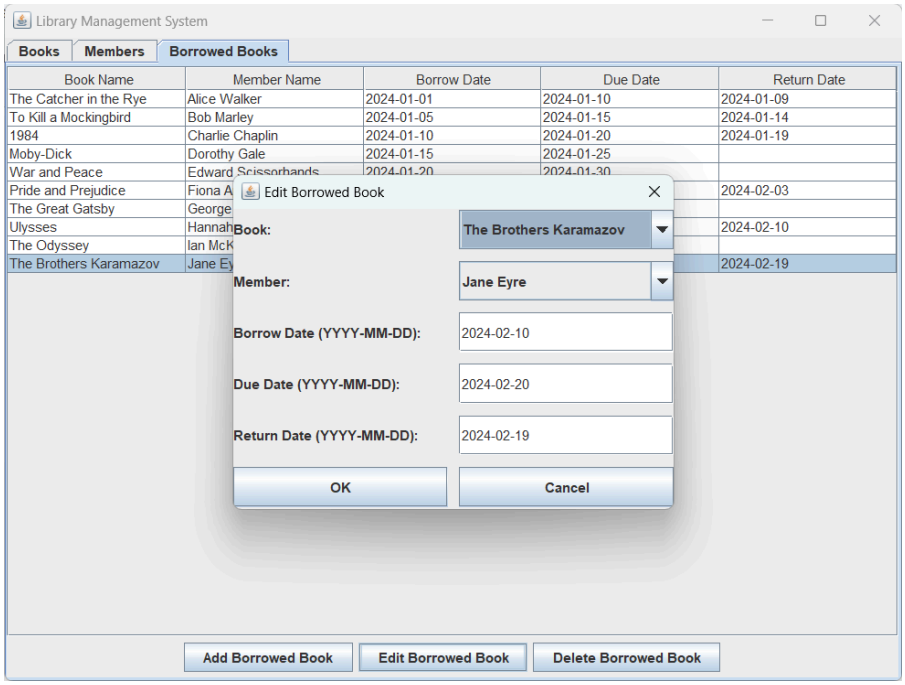
3.3. Borrowed Books Window



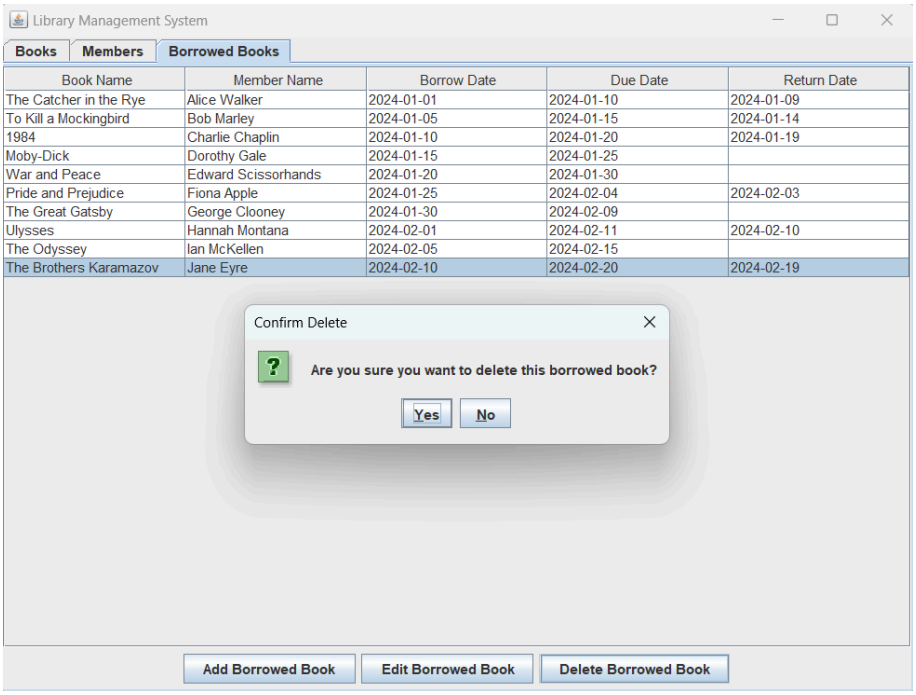
Adding Borrowed Book



Editing Borrowed Book



Deleting Borrowed Book



4. Resources

- ➔ Java: Programming language used to develop the application.
- ➔ Swing: GUI toolkit used for creating the user interface.
- ➔ MySQL: Database management system used for data storage.
- ➔ NetBeans: Integrated Development Environment (IDE) used for coding and designing the application.
- ➔ Poster Link: https://www.canva.com/design/DAGIo34ybw0/jE3oyEYuycHpj3lxLqkpBg/edit?utm_content=DAGIo34ybw0&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton