	Name	Index No
1	Saduka Athukorala	KIC-DCSAI-241-F-039
2	D.M.S.S.Dissanayake	KIC-DCSAI-241-F-016
3	S.M.Ahamed	KIC-DCSAI-241-F-005
4	R.M.Pasandul	KIC-DCSAI-241-F-033
5	N.S.S.Marasinghe	KIC-DCSAI-241-F-030
6	P.B.Ruwanpitiya	KIC-DCSAI-241-F-025

Library Management System

The Library Management System (LMS) is designed to streamline the operations of a library by providing an efficient way to manage books and members. In a typical library environment, staff members often face challenges in tracking book availability, managing member registrations, and processing book loans and returns. The LMS addresses these challenges by offering a digital solution that automates these processes, thereby reducing manual errors and improving overall efficiency.

The system is intended for use by library staff who need to manage the library's inventory of books and the membership of users. It allows for easy access to information, quick updates to the database, and a user-friendly interface that enhances the experience for both staff and library members.

Application Features

The Library Management System includes the following key features:

1. **User Authentication**:
    - A secure login screen that requires users to enter their credentials (username and password) to access the system. This ensures that only authorized personnel can manage library operations.
2. **Book Management**:
    - **Add Book**: Users can add new books to the library by entering the title, author, and ISBN. The system validates the input to ensure all fields are filled.
    - **View Books**: Users can view a comprehensive list of all available and borrowed books in a dialog window, allowing for easy tracking of inventory.
    - **Borrow Book**: Users can borrow books by entering the ISBN. The system checks if the book is available and updates its status accordingly, moving it from the available list to the borrowed list.
    - **Return Book**: Users can return borrowed books by entering the ISBN, which updates the book's status back to available and removes it from the borrowed list.
3. **Member Management**:
    - **Add Member**: Users can add new members by entering their name and member ID. The system ensures that the member ID is unique and valid.
    - **View Members**: Users can view a list of all registered members in a dialog window, providing an overview of the library's membership.
4. **User Interface**:

- The application features a clean and intuitive graphical user interface (GUI) built using Java Swing. The layout is designed for easy navigation, with buttons clearly labeled for each action.

5. **Error Handling and Feedback**:
    - The system provides immediate feedback through dialog messages for successful actions or errors (e.g., missing fields, invalid input). This enhances user experience by guiding users through the process.

Navigation Flow

1. **Login Screen**:
    - Users start at the login screen where they enter their username and password. Successful authentication leads to the main menu.
2. **Main Menu**:
    - The main menu presents buttons for various actions: Add Book, View Books, Borrow Book, Return Book, Add Member, and View Members. Clicking a button opens the corresponding dialog or screen.
3. **Dialogs**:
    - Each action (e.g., adding a book or member) opens a dialog where users can input necessary information. Upon submission, the system validates the input and updates the relevant data.
4. **Feedback**:
    - The system provides feedback through dialog messages for successful actions or errors (e.g., missing fields, invalid input). This ensures users are informed of the status of their actions.

5. **Exit**:
    - Users can exit the application by closing the main window, which triggers the default close operation to terminate the program.

Challenges Faced During Development

1. **User Input Validation**:
    - Ensuring that user inputs were valid (e.g., non-empty fields, correct formats) was a challenge. This was addressed by implementing validation methods that check inputs before processing them, providing error messages when necessary. For example, the member ID must be alphanumeric, and the system throws an error if the input does not meet this criterion.
2. **UI Responsiveness**:
    - Maintaining a responsive UI while performing operations was crucial. The use of **SwingWorker** for background tasks (like showing the splash screen) helped keep the UI responsive during long-running operations. This approach prevents the application from freezing while waiting for user input or processing data.
3. **Data Management**:
    - Managing the state of books and members (e.g., available vs. borrowed) required careful handling of data structures. This was addressed by using lists to track available and borrowed books, ensuring that updates were reflected in the UI.
      The **BookManager** and **MemberManager** classes encapsulate the logic for managing these lists effectively.
4. **Error Handling**:

- Proper error handling was essential to prevent crashes and provide a good user experience. This was achieved by using try-catch blocks and displaying informative error messages to users. For instance, if a user attempts to borrow a book that is not available, the system provides a clear message indicating the issue.

5. **User Experience Design**:
    - Designing an intuitive user interface that caters to users with varying levels of technical expertise was a challenge. This was addressed by conducting user testing sessions to gather feedback on the interface design and navigation flow. Adjustments were made based on user input to ensure that the application is accessible and easy to use for all staff members.

Conclusion

The Library Management System provides a comprehensive solution for managing library operations, enhancing user experience through its intuitive interface and robust features. The challenges faced during development were effectively addressed, resulting in a functional and user-friendly application. The system not only improves the efficiency of library operations but also ensures that users can easily manage their interactions with the library, making it a valuable tool for any library environment.