

Library Management system

Group 8



January 26, 2020

MIU student

Contents

[Introduction 2](#_Toc30955769)

[Purpose 2](#_Toc30955770)

[End User 2](#_Toc30955771)

[System Environment 3](#_Toc30955772)

[Design Approach 3](#_Toc30955773)

[UI Design 6](#_Toc30955774)

[Design Patterns 14](#_Toc30955775)

[Extra Features 15](#_Toc30955776)

1. **Introduction**

This introduction provides an overview of the Library Management System. It is managing the library and library-oriented tasks. It helps to provide information on any book present in library to the user as well as staff member. It keeps a track of book issued, returned and added to library.

1. **Purpose**

The purpose of this application are as follows:

* To enter details related to a particular book
* To provide membership to members
* To provide employee account to employee
* To check out the books when issued to members
* To add a copy of book
* To display all library record

1. **End User**

Our end users are:

* **Administrator**

The responsive pages for administrator are:

* Create Employee Account
* Add New Member
* Add New Book
* Add A Copy of Existing Book
* **Librarian**

The responsive pages for administrator are:

* Check Out A Book
* Check in A Book
* **User owns both account types**

It contains administrator and librarian responsive pages

1. **System Environment**

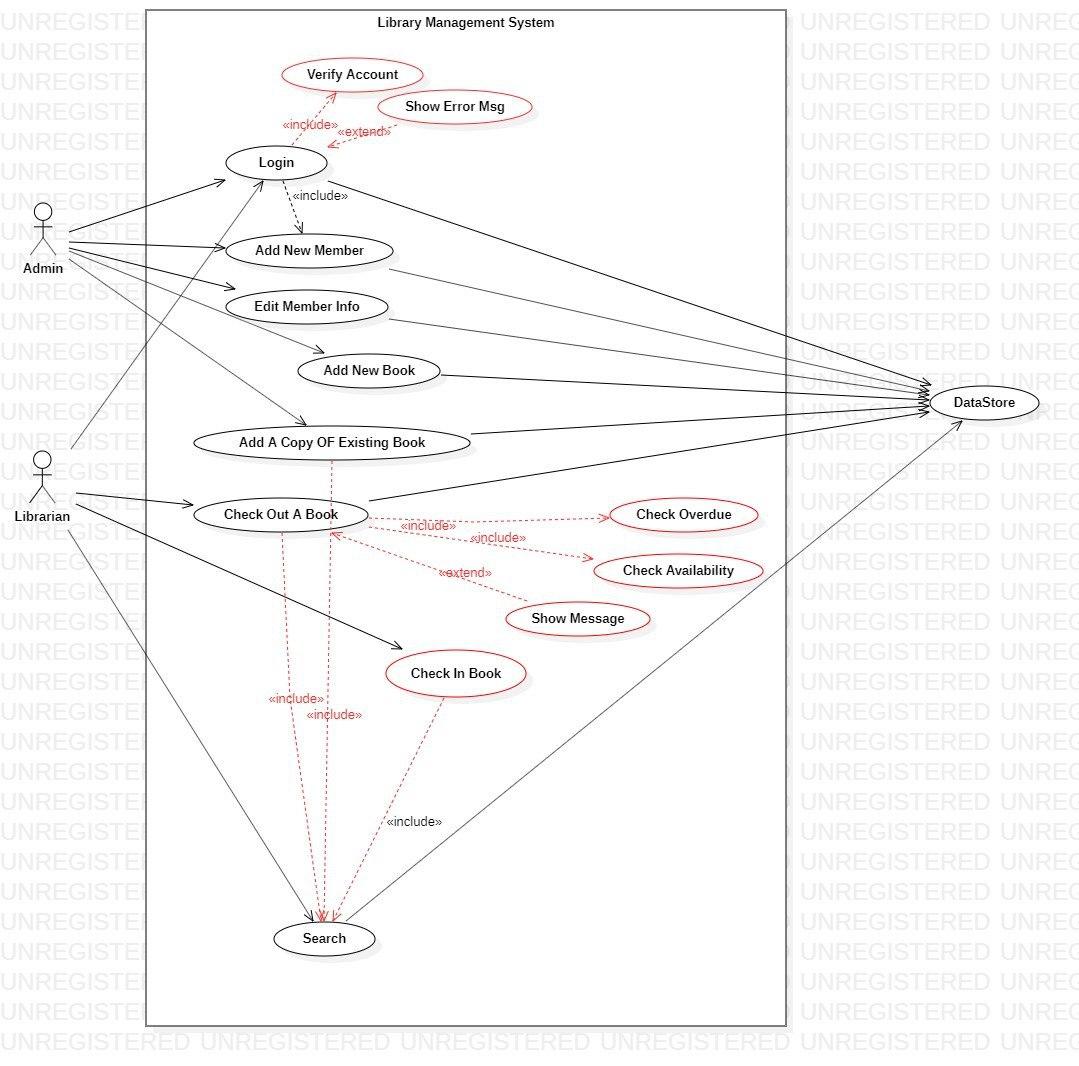
* Development: Eclipse
* Diagrams: StartUmL
* Database Management: Class of Object Serialization
* OS: window 10

1. **Design Approach**

The system design based on the following steps:

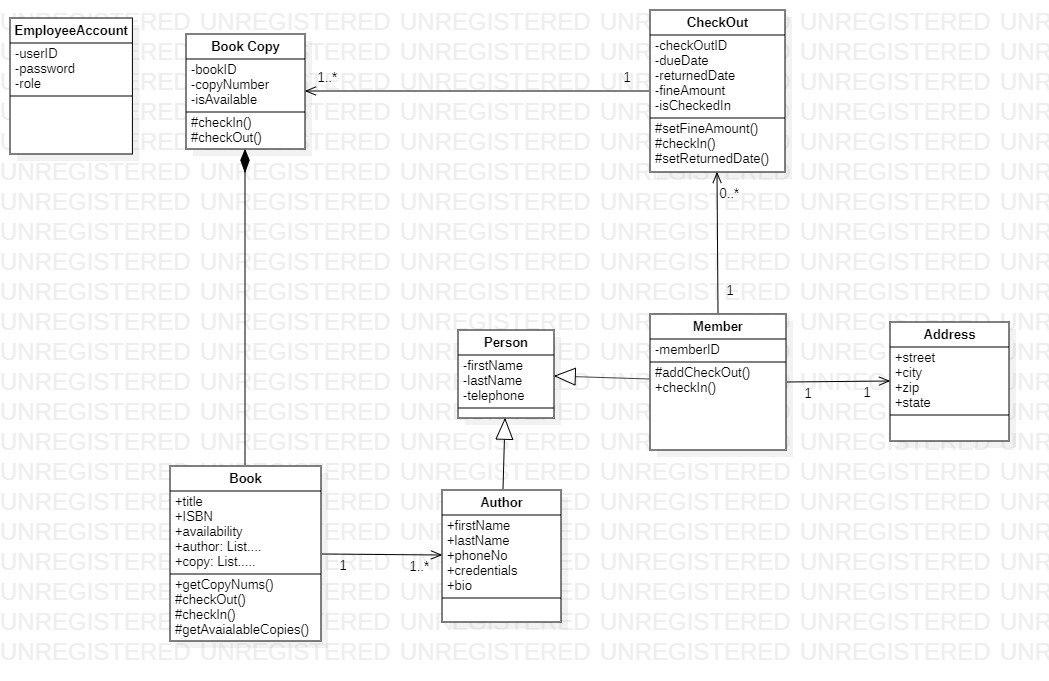
* 1. **Use Case**

The use case diagram of library management system will be describing the interaction between actor and system action. The actor in each case is either a Librarian or an administrator who will be using the system. The actor will be interacting with the UI. There is picture below to describe the use case diagram.



* 1. **Class Diagram**

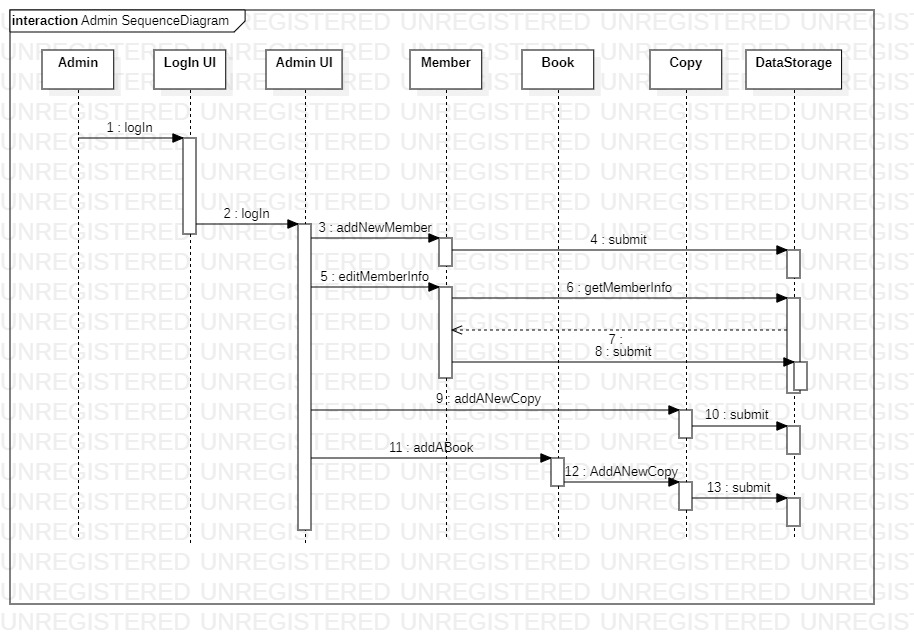
The class diagram contains a class of system. Each class has attributes and operations for implementing purpose and identify inheritance relationship. Create a complete class diagram that includes the associations between your class.



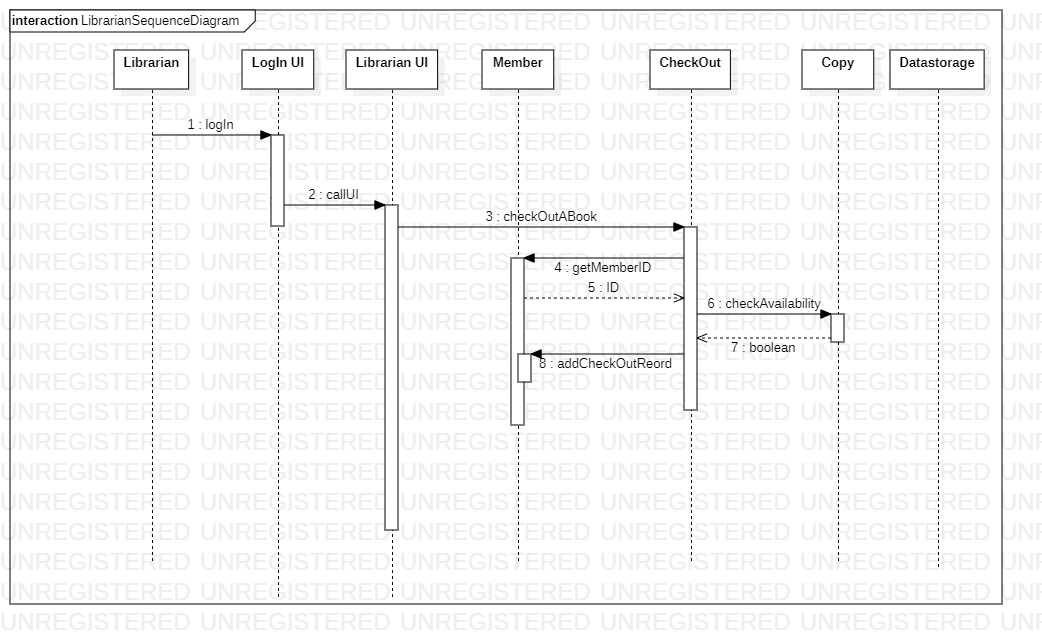
* 1. **Sequence Diagram**

The Sequence Diagram will be describing the flow for each of the use case diagram. It helps to identify operations in class diagram.

**Administrator Sequence Diagram**



**Librarian Sequence Diagram**



1. **UI Design**

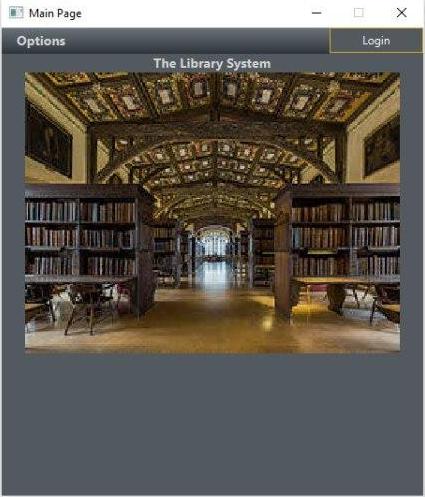
The system User Interface is done with Javafx. It is an effective for design desktop application for collecting and presenting functionality.

The System has different pages with different functionality

* Main Page
* Login Page
* Add New Member Page
* Add New Book Page
* Add a Copy of Book Page
* Books Check Out Page
* Library Records Page
* Books Check In Page.

**Main Page:**

* It is the first page the user will look at.
* There is a change theme option on this page.



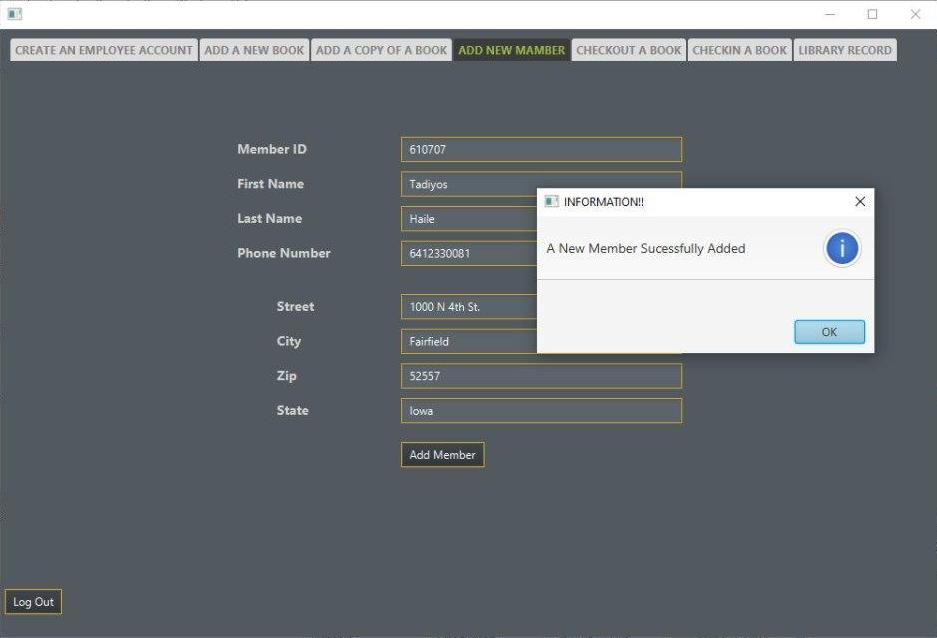
**Login Page**

* This is the page at which the user can login into the system
* The user required to put Username and Password
* Authorized User will be allowed to navigate further within the application.



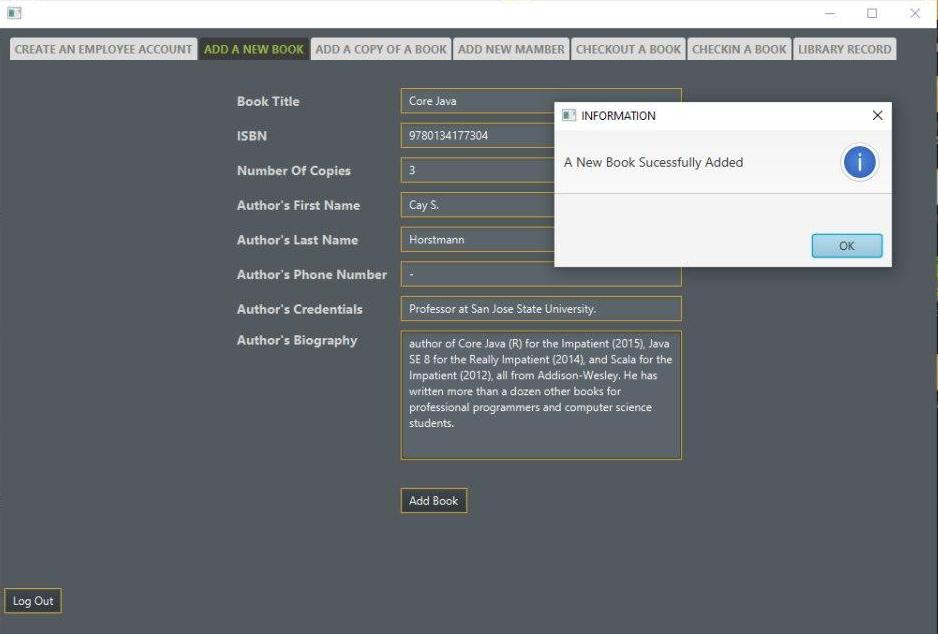
**Add New Member Page**

* This page can be accessed by administrator
* On this page the admin can create a new member.



**Add New Book Page**

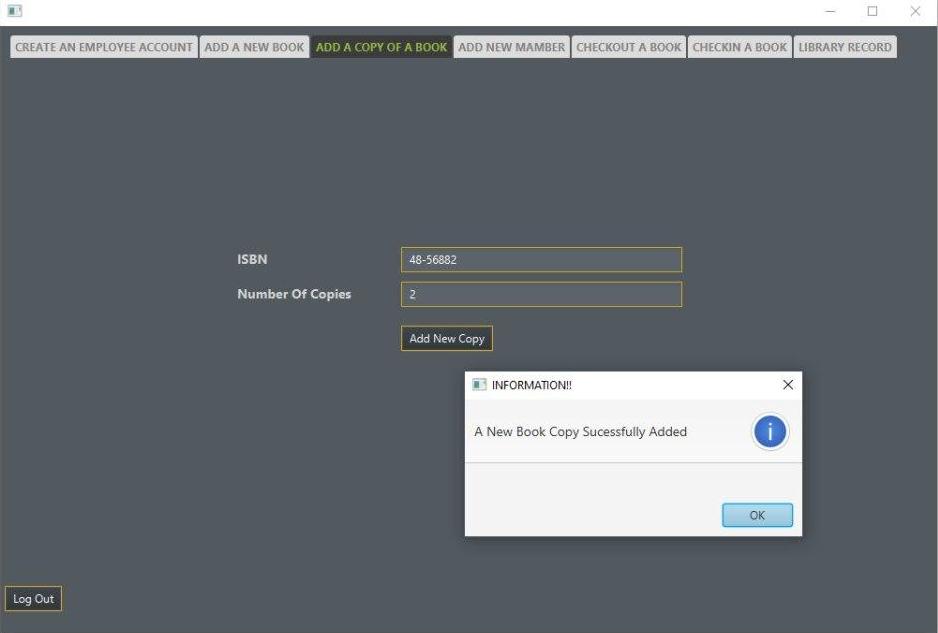
* This page can be accessed by administrator
* On this page the admin can register new books that are introduced to the library.



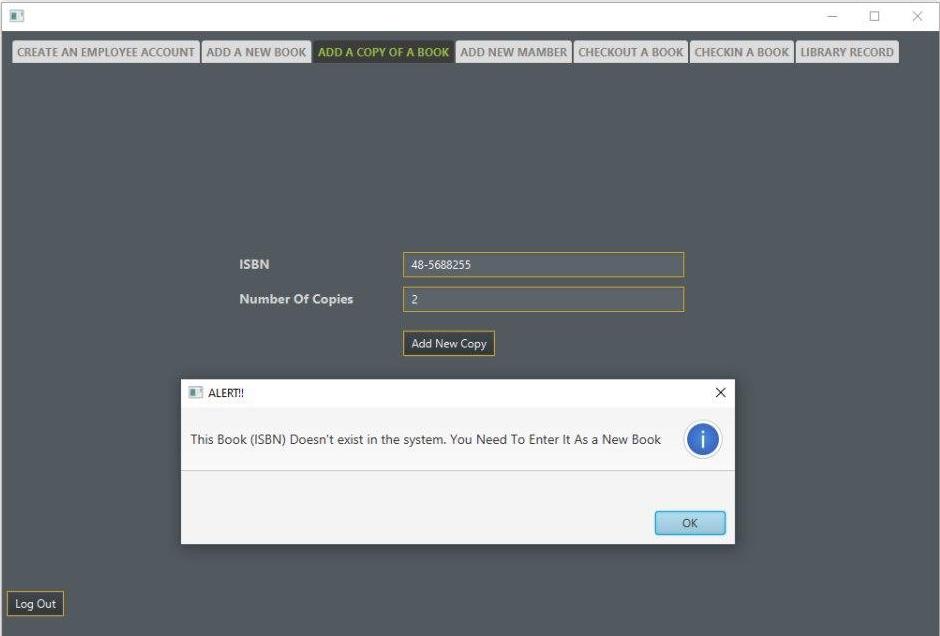
**Add A Copy of Book Page**

* This page can be accessed by administrator
* On this page the administrator can register a new copy of already existing book.
* The admin doesn’t required to put every data related to books since the book data is previously registered on the system.

**Successful Book Copy Add Page**



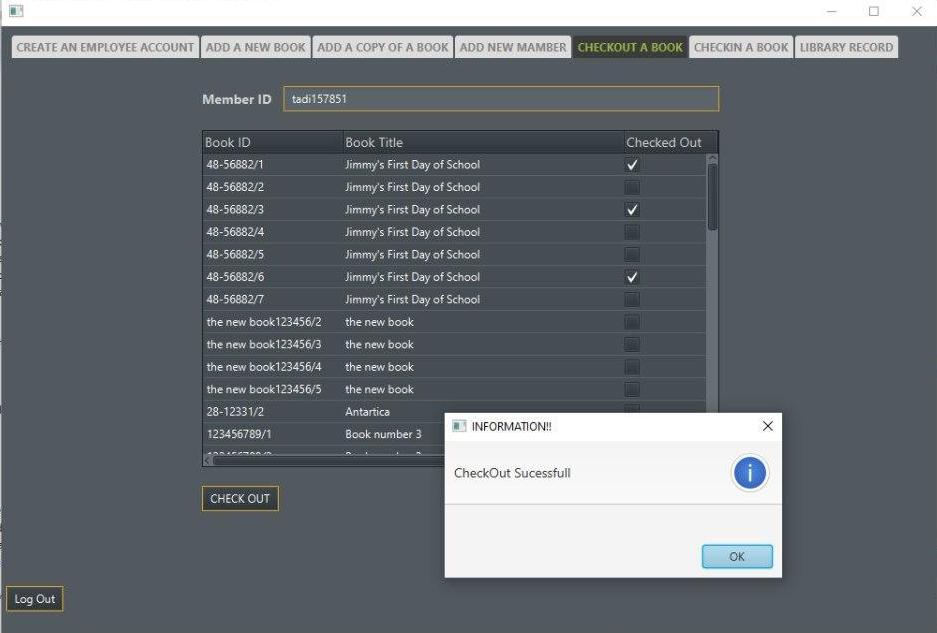
**Un Successful Book Copy Add Page**



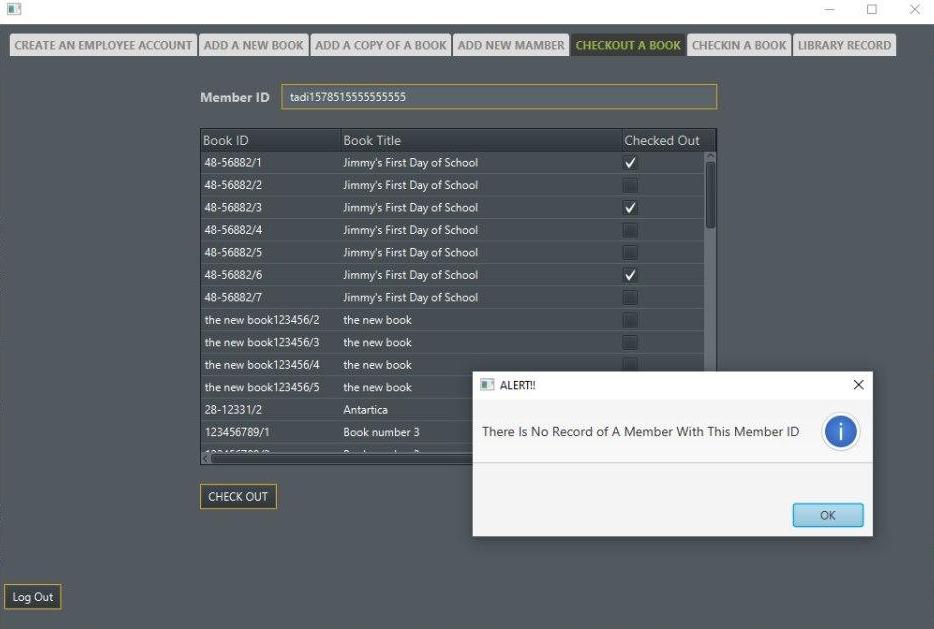
**Check Out Book Page**

* This page can be accessed by Librarian.
* On this page a librarian can check out a book for library members only.

**Successful Book Check Out Page**



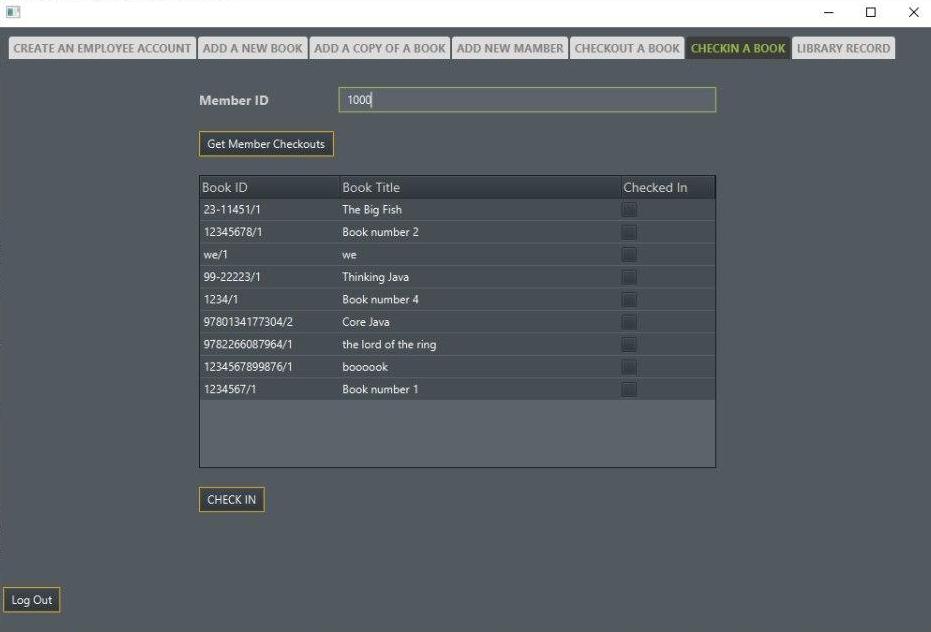
**Unsuccessful Book Check Out**



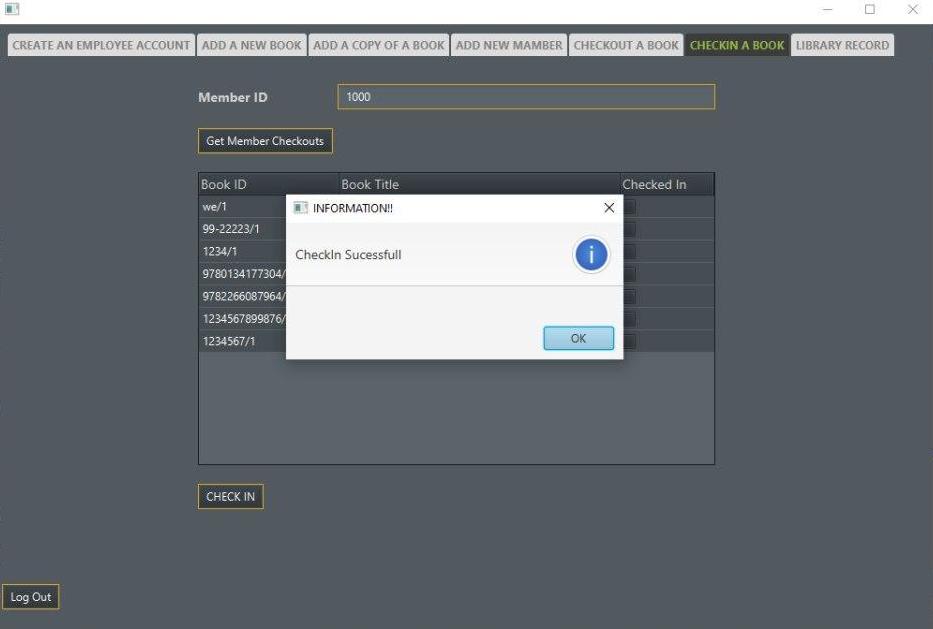
**Check In Book Page**

* This page can be accessed by Librarian
* Check Out histories will be displayed to the user
* On this page a librarian can check in a book using members IDs.
* A system will check the due date and returned date of a book and may calculate a fine if the books returned beyond the due date.

**Check Out history of Books**

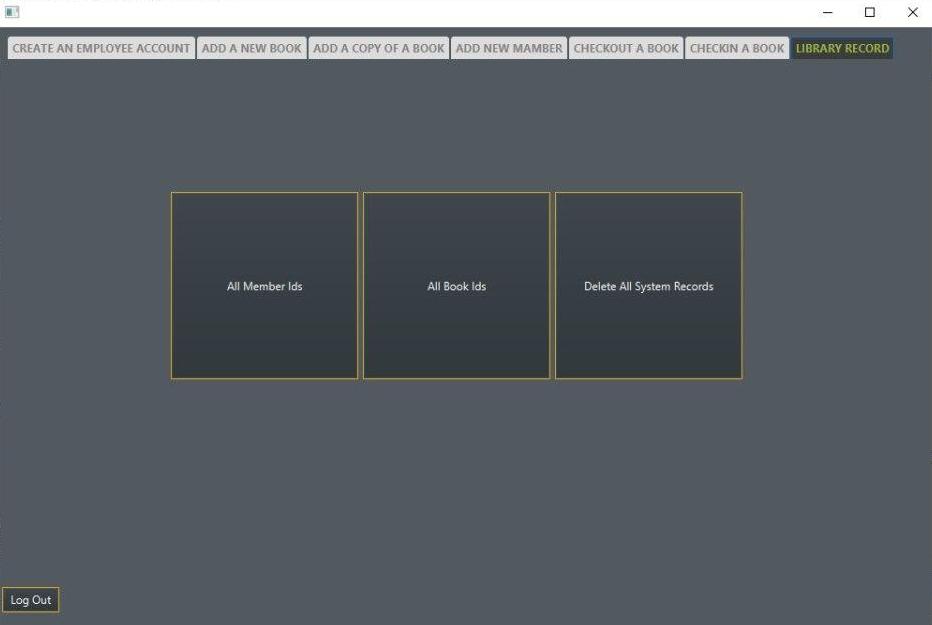


**Successful Check In Page**

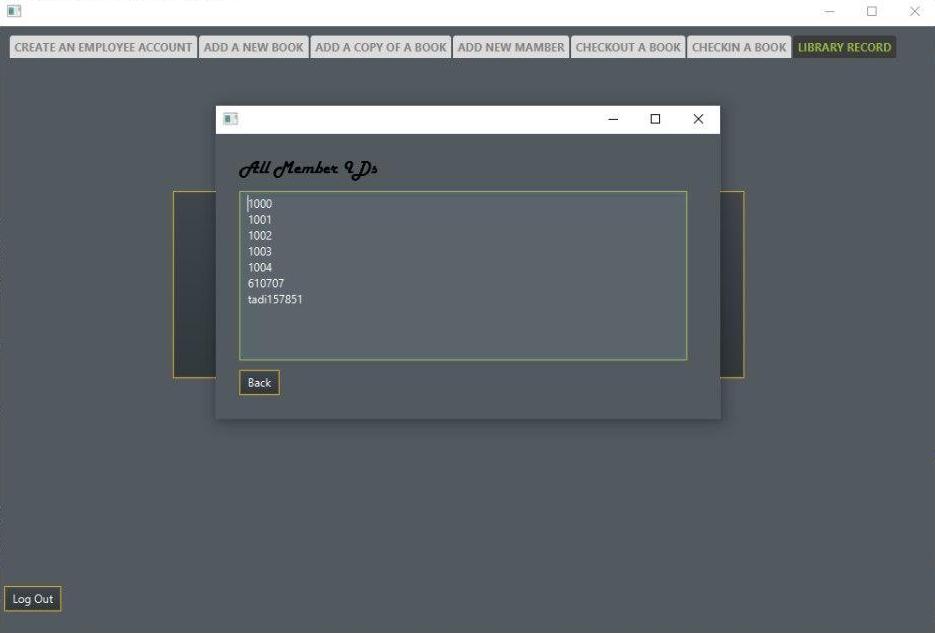


**Library Records Page**

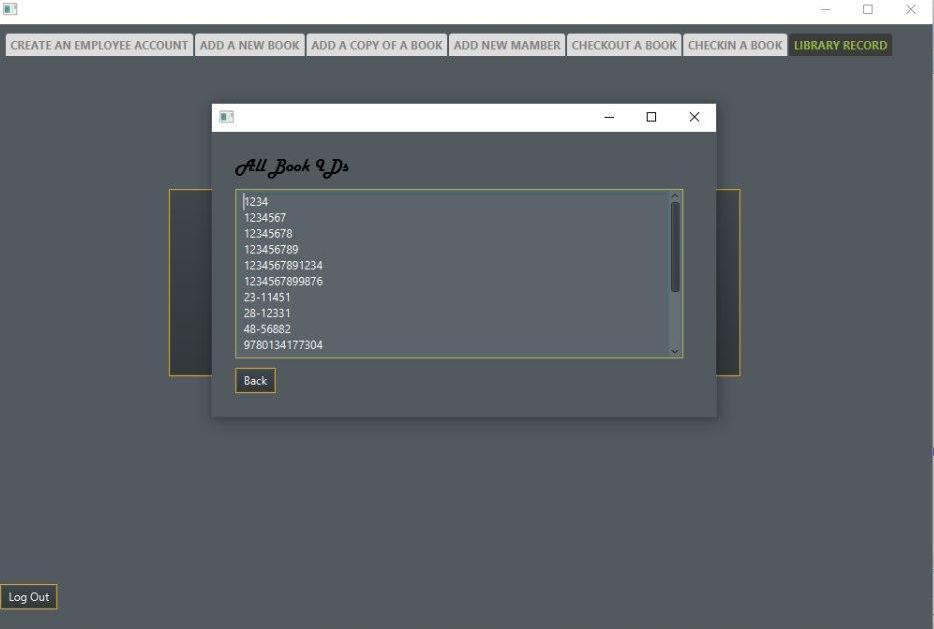
* This page can be accessed by Administrator
* On this page a administrator can see all library record data’s.



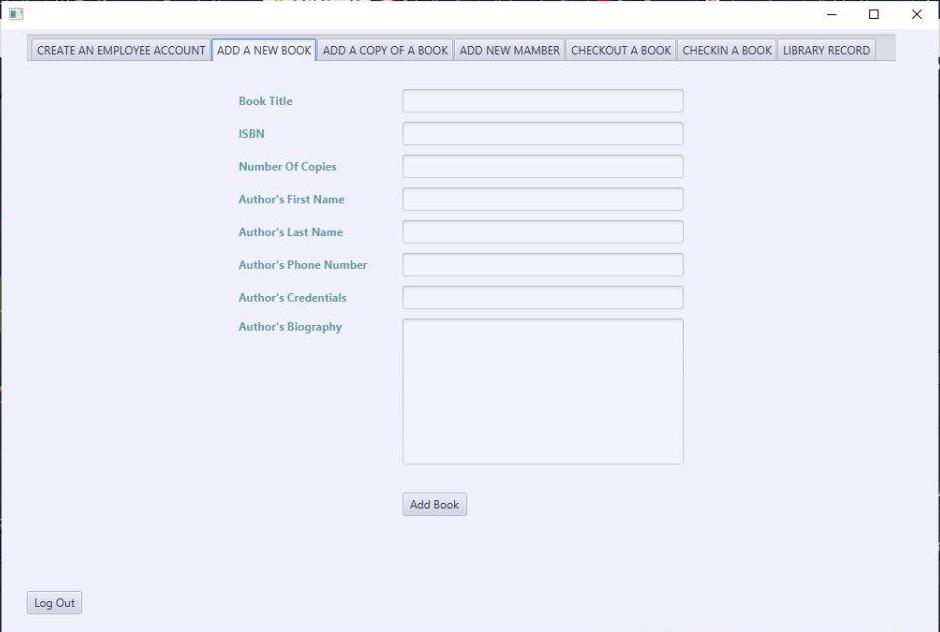
**Display All Members Id**



**Display All Books Id**



**Application With New Theme**



1. **Design Patterns**

This application is designed as an object-oriented system for window-based architecture using three-layer architecture by factoring application classes into the following layers:

1. **Data Access Layer**

* On the application the data needs to be separated from business logic. The data is managed by using persisted class to store using object serialization.

1. **Business Logic Layer**

* This is the layer where the class diagram is implemented using the Object Oriented Programming concept. It is also creating a connection between Data Access and Presentation Layer. It acts as a controller of transfer data. Most objects identified in the OO analysis and design will reside.

1. **Presentation Layer**

* This is the layer where the physical window and widget objects. Any new user interface widgets developed for this application are put in this layer. This layer is completely developed using JavaFx.

1. **Extra Features**

The following new features are implemented on this application. These are

1. Check In multiple books at a time
2. Check out multiple books at a time
3. Delete all system records at a time and only applicable for Admin