Book Management

Team members:

* Do Minh Duc
* Le Phuong Vinh

Table of contents

1. Project Over View 04

1. Problem definition 05

2. Customer Request Specification 05

2.1. Search Book 06

2.2. Update book 06

2.3. Delete book 06

2.4. View book 06

2.5. Add new book 06

2.6. Rent book 07

2.7. Return book 07

2.8. Login 07

1. Diagram 08

Class Diagram 09

Sequence Diagram 10

Entity Relations Diagram 21

Table Relationship diagram 23

Use Case Diagram 24

System Architecture Diagram 29

1. User Interface 30

Login form design 31

Admin form design 32

User form design 33

Project Over View

**Problem definition**

1. **The objective of project:**

This project will help librarian who manages all books in library school more easy. Otherwise, this application will help librarian

This application shows status of books which exist or not. This application provides the following facilities:

Search book by name

Search book by author

Update book

Delete book

View book

Add new book

Rent book

Return book

Login

**Customer Request Specification**

**2. Functional Requirements:**

Function requirements of the system are listed as below:

**2.1. Search Book:**

\_ As an admin, I can search books by name.

\_ As a User, I can search books by name.

\_ As a User, I can search books by the author’s name.

**2.2. Update book:**

\_ As an Admin, I can update the information of book base on book id.

**2.3. Delete book:**

\_ As an Admin, I can delete book by base on book id.

**2.4. View book:**

\_ As an Admin, I can view information of all books in the library (information of book like id, name, status, price, quantity, author).

\_ As a User, I can view information of all books in the library (information of book like id, name, status, price, quantity, author).

\_ As a User, I can view information about books that rented.

**2.5. Add a new book:**

\_ As an Admin, I can add a book. All information of the book was added that must have id, name, status, price, quantity, author.

**2.6. Rent book:**

\_ As a User, I can search a book that users want to rent by 2 ways (search by name, search by author).

\_ As a User, I want to choose a book and rent it.

**2.7. Return book:**

\_ As a User, I want to return one book in a list a have borrowed.

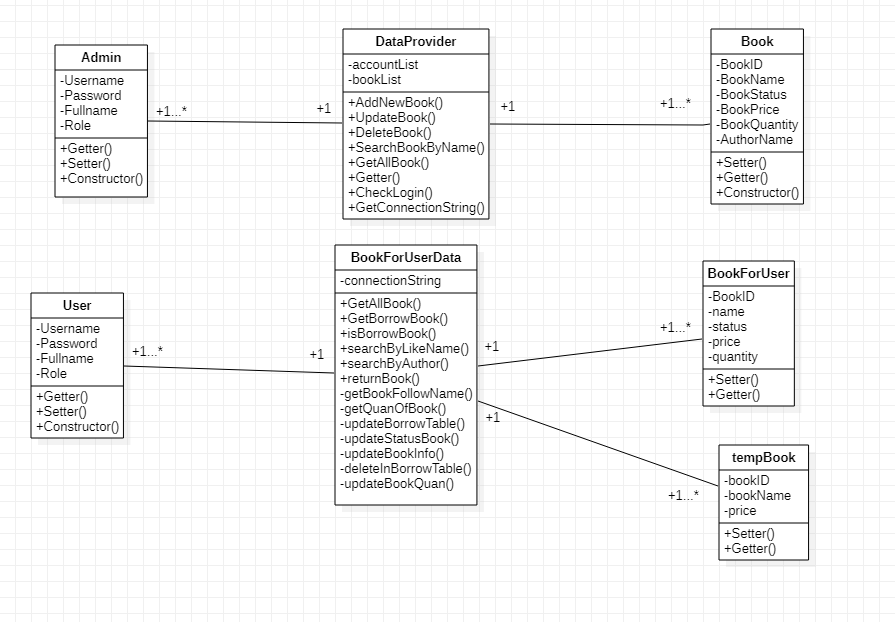
\_ As a User, I want to return all my borrowing books by one click.

**2.8. Login:**

\_ As a software manager, I want you to device the end-user into two types. The first one is a normal user, and the second one is an admin user by authentication when they log in.

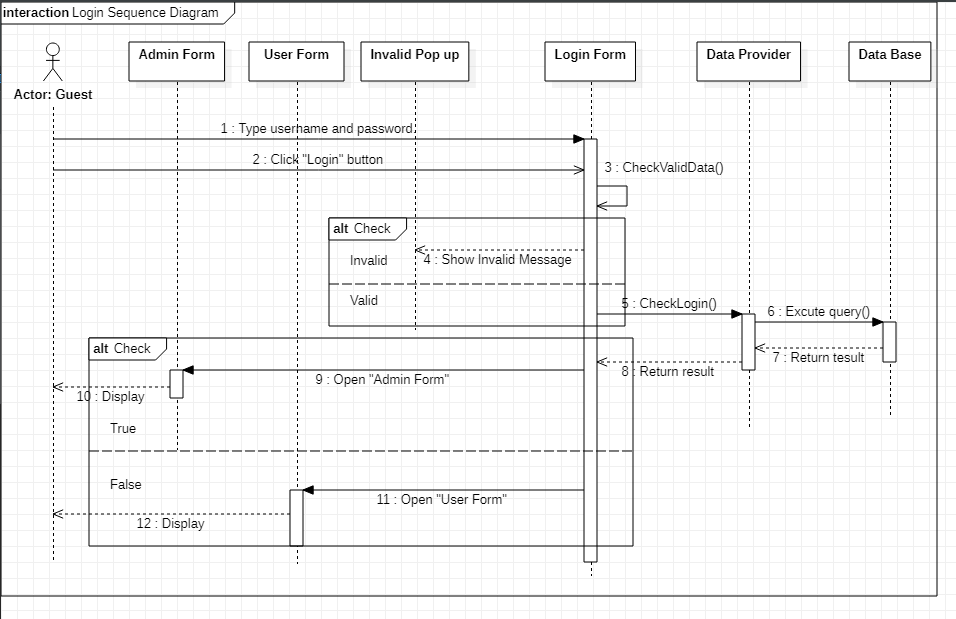
Diagram

**Class Diagram**



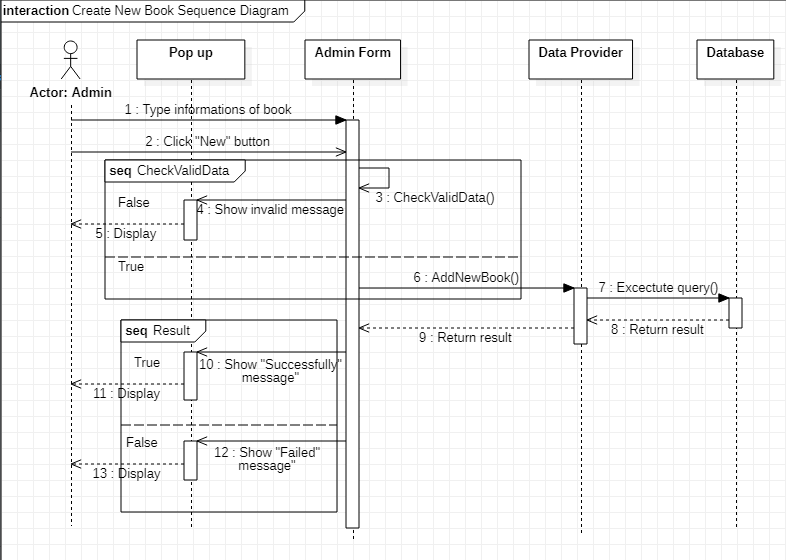
**Sequence diagram**

**Login:**

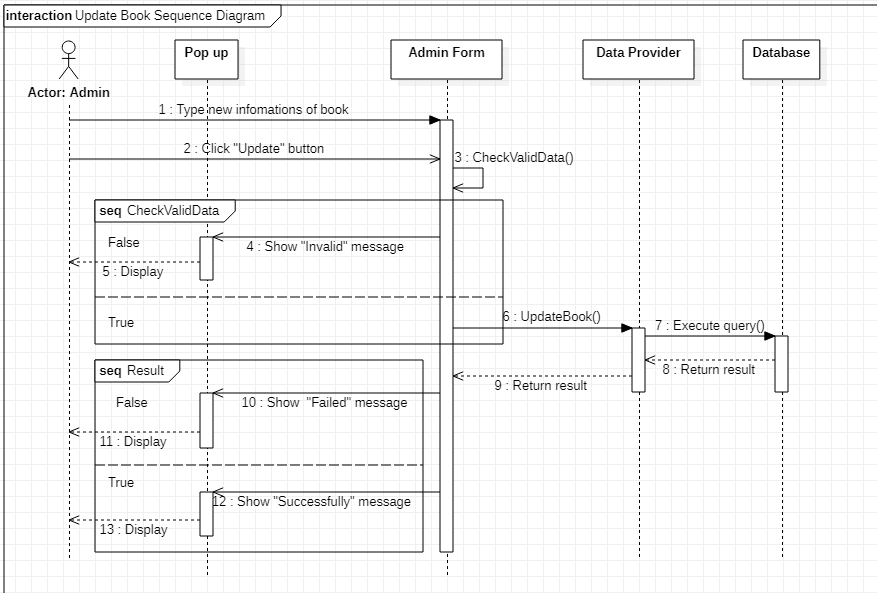


**Admin:**

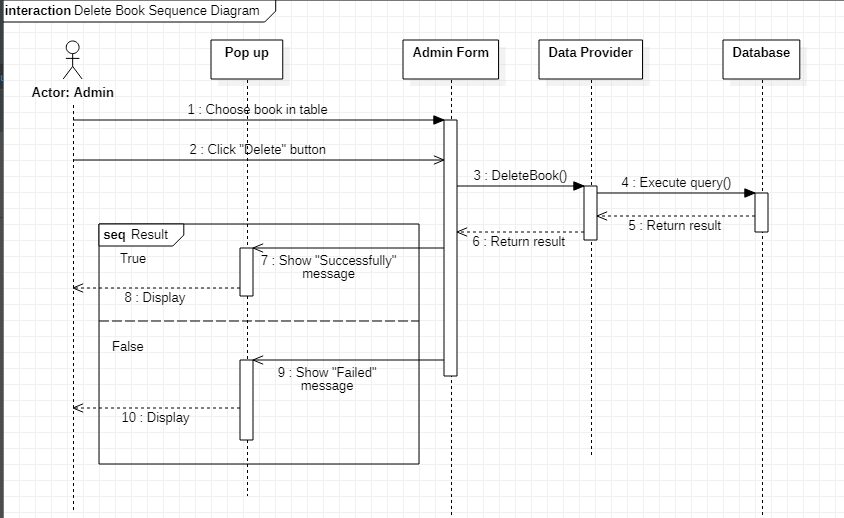
**Add new book:**



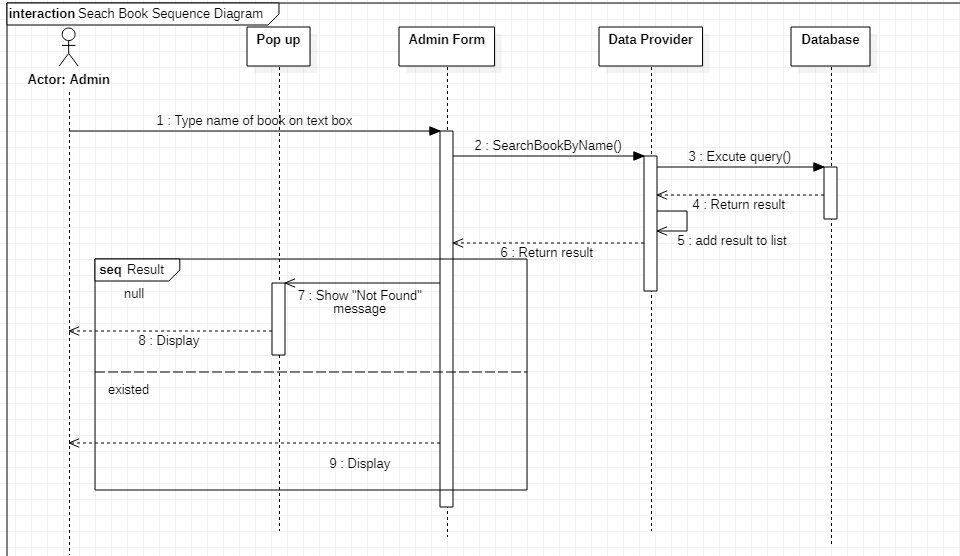
**Update book:**



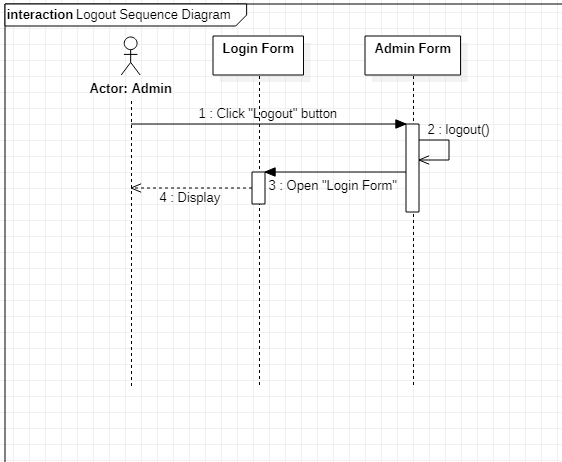
**Delete book:**



**Search book by name:**

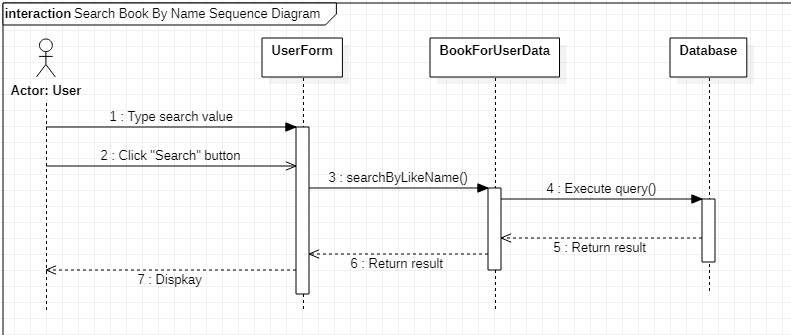


**Logout:**

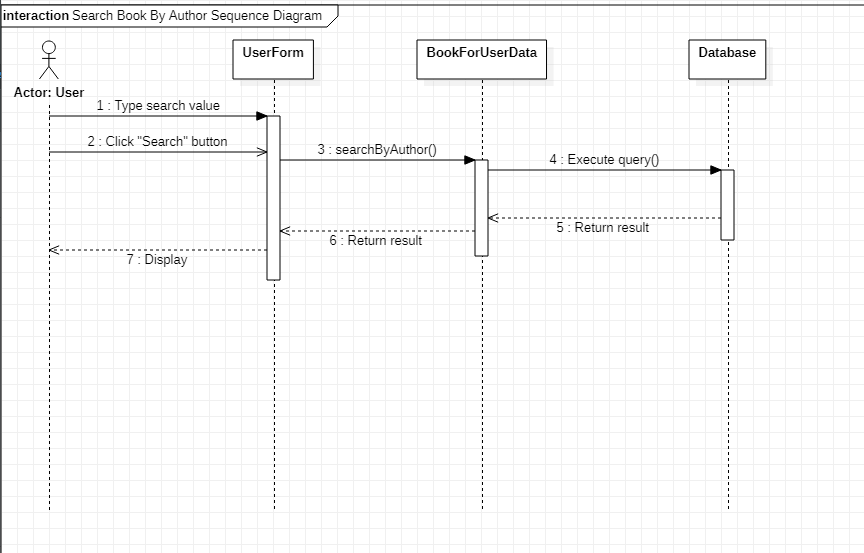


**User:**

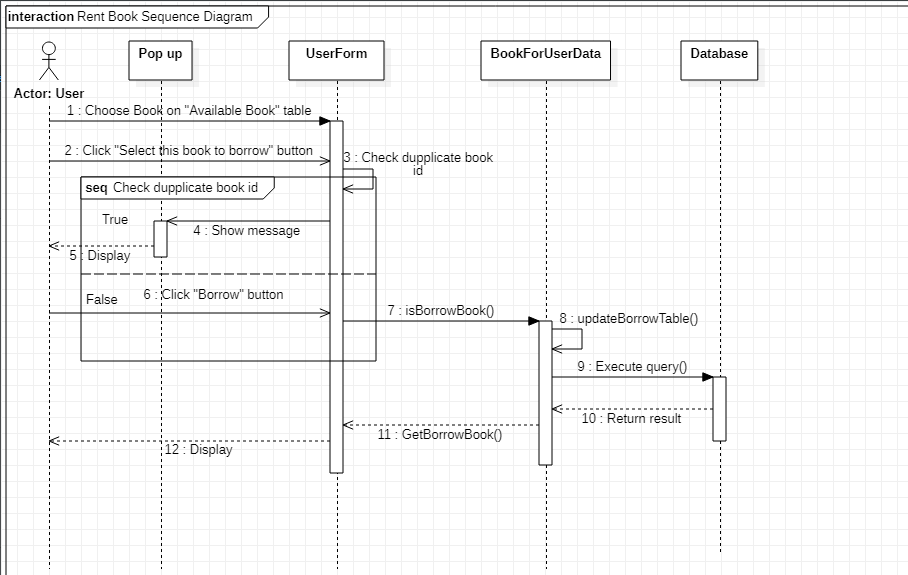
**Search book by name:**



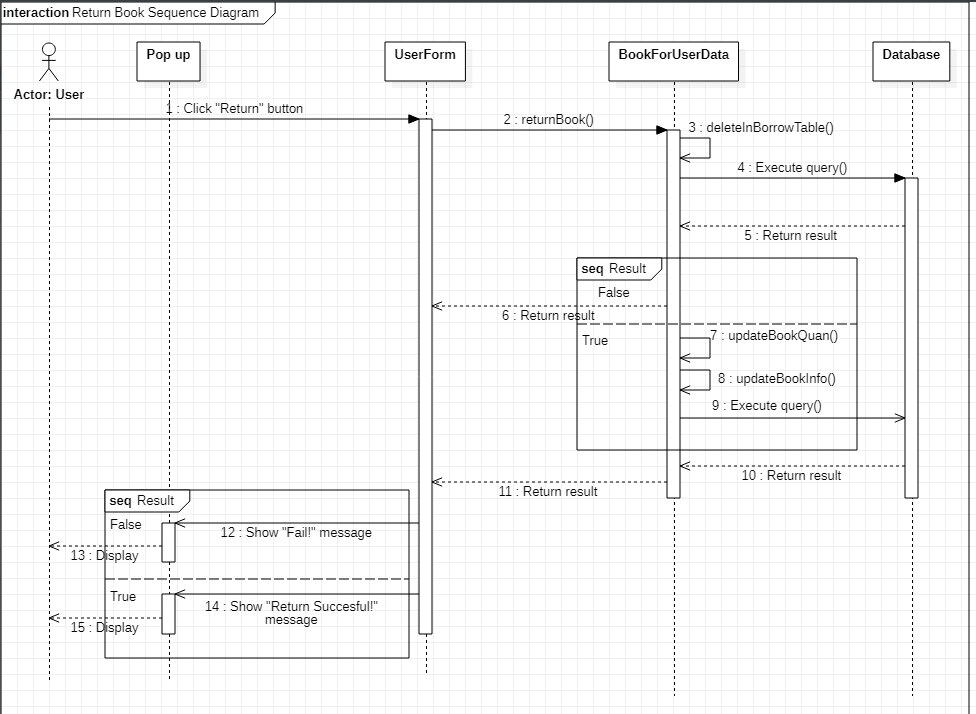
**Search book by author:**

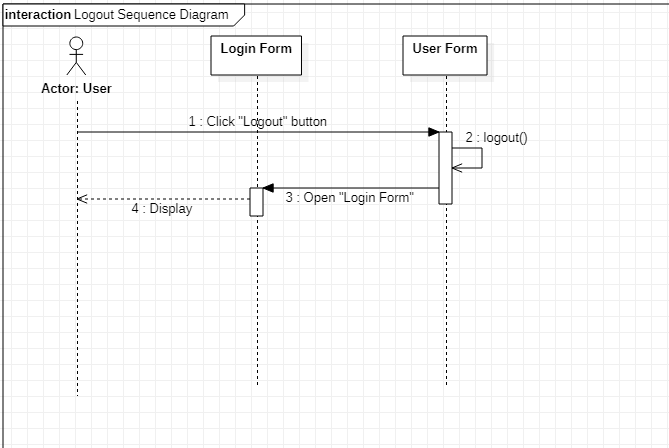


**Rent book:**



**Return book:**

 **Logout:**



**ER diagram:**

Borrow books

User

Book

: Entity : Many to many relationship

: Entity Relationship

**Table design:**

1. Table Account:

Contents user information

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Column Name** | **Data Type** | **Size** | **Check** | **Description** |
| 1 | **🗝**Username | Varchar | 30 | Not Null | User’s ID |
| 2 | Passwords | Varchar | 20 | Not Null | User’s password |
| 3 | Fullname | Nvarchar | 50 | Not Null | Full name of user |
| 4 | Account Status | Bit |  | Not Null | Customer’s user name |

1. Table Book:

Contents book information

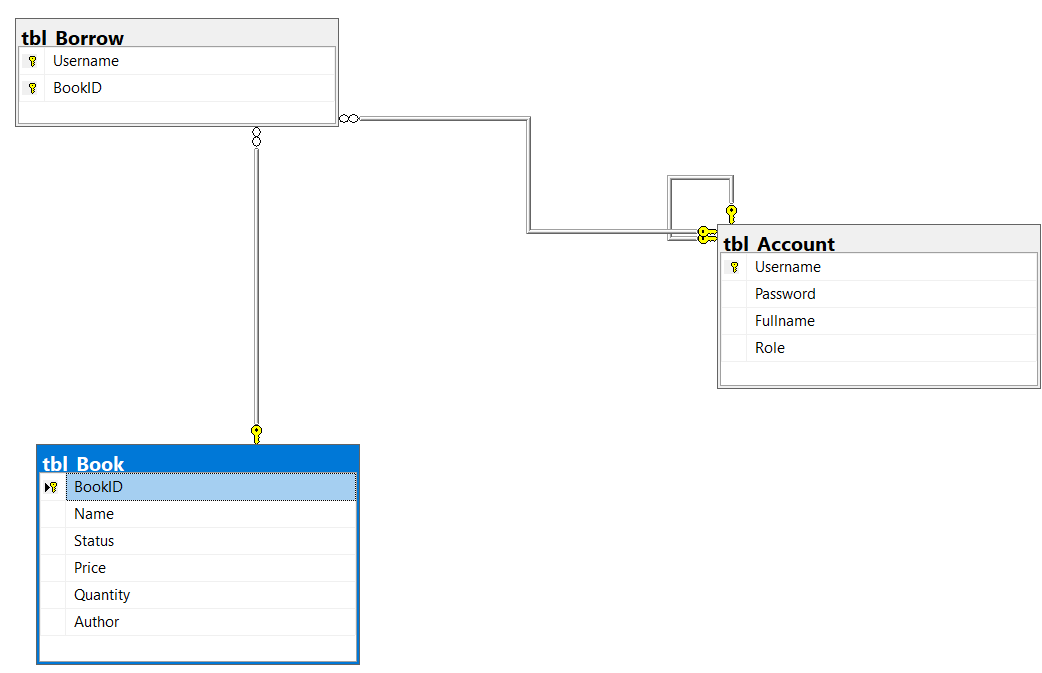
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Column Name** | **Data Type** | **Size** | **Check** | **Description** |
| 1 | **🗝**BookID | Varchar | 30 | Not Null | Book’s ID |
| 2 | Name | Varchar | 30 | Not Null | Book’s name |
| 3 | Status | Bit |  | Not Null | Show that this book is ready for borrow or not |
| 4 | Price | Numeric | 6 | Not Null | Book’s price |
| 5 | Quantity | Int |  | Not Null | Quantity of book in library |
| 6 | Author | Nvarchar | 50 | Not Null | Author of book |

1. Table Borrow:

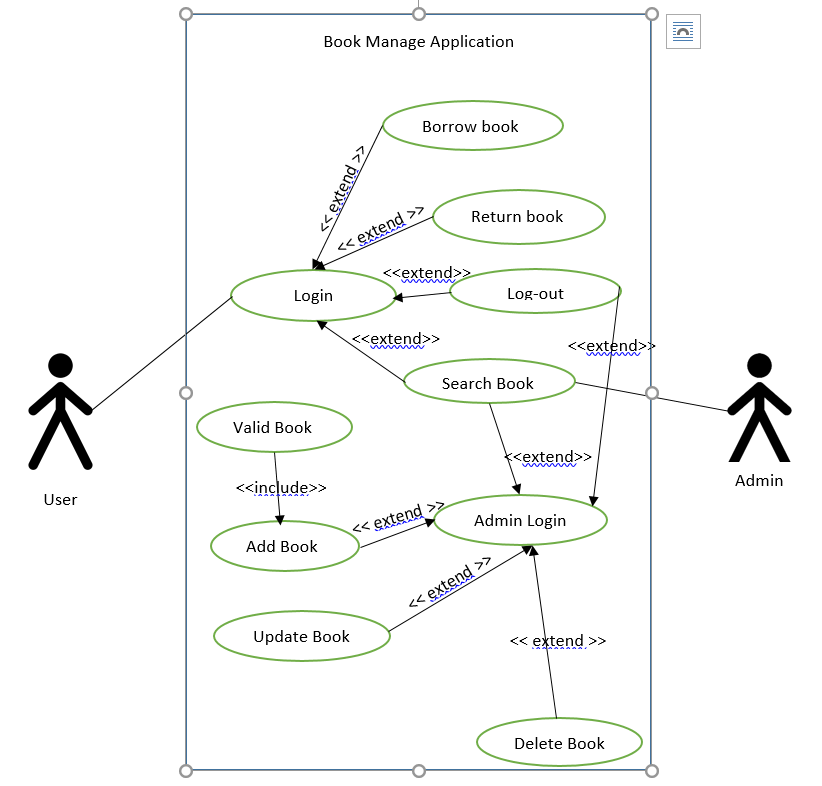
Contents That user is borrowing what book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Column Name** | **Data Type** | **Size** | **Check** | **Description** |
| 1 | **🗝**Username | Varchar | 30 | Not Null | User’s ID |
| 2 | **🗝**BookID | Varchar | 30 | Not Null | ID of book |

**Table Relationship diagram:**



**Use case Diagram:**



Use case description:

1. Actor:
   1. User: is a student who will come to library and borrow book, or view, or return their borrowing books.
   2. Admin: is a library manager, who will manage book by update information of books, delete one book from library, add new book to library.
2. Use case:

|  |  |
| --- | --- |
| **Usecase** | Login |
| **Short Description** | Login in to the book management system |
| **Pre-Conditions** |  |
| **Post-Conditions** |  |
| **Main flow** | * User input information: * Username * Password * User click on the button login * System check the username and password * If it’s valid, the system will move them to the corresponding UI follow the role * If the password is wrong shows the message invalid to them. |
| **Alternative flow(s)** | * If the user clicks on the “Reset” button, the Login ID field and Password field are blank. * If the username textbox or Password textbox is blank when the guest clicks on the “Login” button, the system will show invalid. |

|  |  |
| --- | --- |
| **Usecase** | Logout |
| **Short Description** | Log out of the book management system |
| **Pre-Conditions** |  |
| **Post-Conditions** |  |
| **Main flow** | * User click the “Logout” button. * The system will forward user to the login page. |
| **Alternative flow(s)** |  |

|  |  |
| --- | --- |
| **Usecase** | Search Book |
| **Short Description** | The user after login can search book in library |
| **Pre-Conditions** |  |
| **Post-Conditions** |  |
| **Main flow** | * User input the information of book into the search textbox * User click the button search * The system will query to the database * If it has corresponding information of book will show all result in the table * If it doesn’t have shown a null table |
| **Alternative flow(s)** | * If the user doesn’t type anything into the textbox and click search system will do nothing. |

|  |  |
| --- | --- |
| **Usecase** | Borrow Book |
| **Short Description** | The student come to library and borrow book |
| **Pre-Conditions** |  |
| **Post-Conditions** |  |
| **Main flow** | * The user will select a book on the available book table. * Click the button select to select that book they are choosing. * Check the book they choose isn’t the book they are borrowing. * The system will save information about a book they select into the selected table. * View the selected table again to make sure about what they want to borrow. * If they have satisfied with their selection. Click the button borrow. * The system will insert to the database that the user is logging in and id of the book they borrow, and minus the quantity of this book to 1 unit. |
| **Alternative flow(s)** | * If the select the book they are borrowing, show notification message that we refuse them to choose this book. * If they don’t satisfied with their selection. * Select the book they don’t satisfy * Click the Unselect button to “unselect” this book or “unselect all” button to remove all books they selected. |

|  |  |
| --- | --- |
| **Usecase** | Return book |
| **Short Description** | The student come to library and return book they are borrowing |
| **Pre-Conditions** |  |
| **Post-Conditions** |  |
| **Main flow** | * The user will select a book they want to return on the table Borrowing book after login. * The user will click the “return” button. * System will delete that username and bookid in the table borrow in database, update plus the quantity of book to 1 unit. * Show message that the return successful. |
| **Alternative flow(s)** | * The user can return all the book they are borrowing by click button return all * System will browse each book and do return function. |

|  |  |
| --- | --- |
| **Usecase** | Add Book |
| **Short Description** | The admin add the new book to library |
| **Pre-Conditions** |  |
| **Post-Conditions** |  |
| **Main flow** | * The user will click the “new” button to input: * Book id * Book name * Price * Status * Quantity * Author * Click the “new” button again. * The system will check this book is valid. * The system will insert the book’s information into the database and show a message that adds successfully |
| **Alternative flow(s)** | * The book is an invalid book, show the message that they need to edit it again to make it is a valid book. |

|  |  |
| --- | --- |
| **Usecase** | Update name, price, quantity, status, and author of book. |
| **Short Description** | The admin update name, price, quantity, status, and author book to library |
| **Pre-Conditions** |  |
| **Post-Conditions** |  |
| **Main flow** | * The user will click the “update” button to input: * Book name * Price * Status * Quantity * Author * Click the “Update” button again. * The system will check this book is valid. * The system will update the book’s information into the database and show a message that update successfully |
| **Alternative flow(s)** | * The book is an invalid book, show the message that they need to edit it again to make it is a valid book. |

|  |  |
| --- | --- |
| **Usecase** | Delete book |
| **Short Description** | The admin delete the book in library |
| **Pre-Conditions** |  |
| **Post-Conditions** |  |
| **Main flow** | * Admin will select the book in the table. * Click the “Delete” button. * The System will get the id of selected book on the table. * The system will call the DataProvider class to get delete book function. * After delete, show the message that “Delete successful” to admin |
| **Alternative flow(s)** | * Some things went wrong when delete, show the message “Delete fail” |

**System Architecture Diagram:**

|  |  |
| --- | --- |
| Login Form | |
| Admin Form | User Form |
| Business Object | |
| DataProvider | BookForUserData |
| Database | |

User Interface

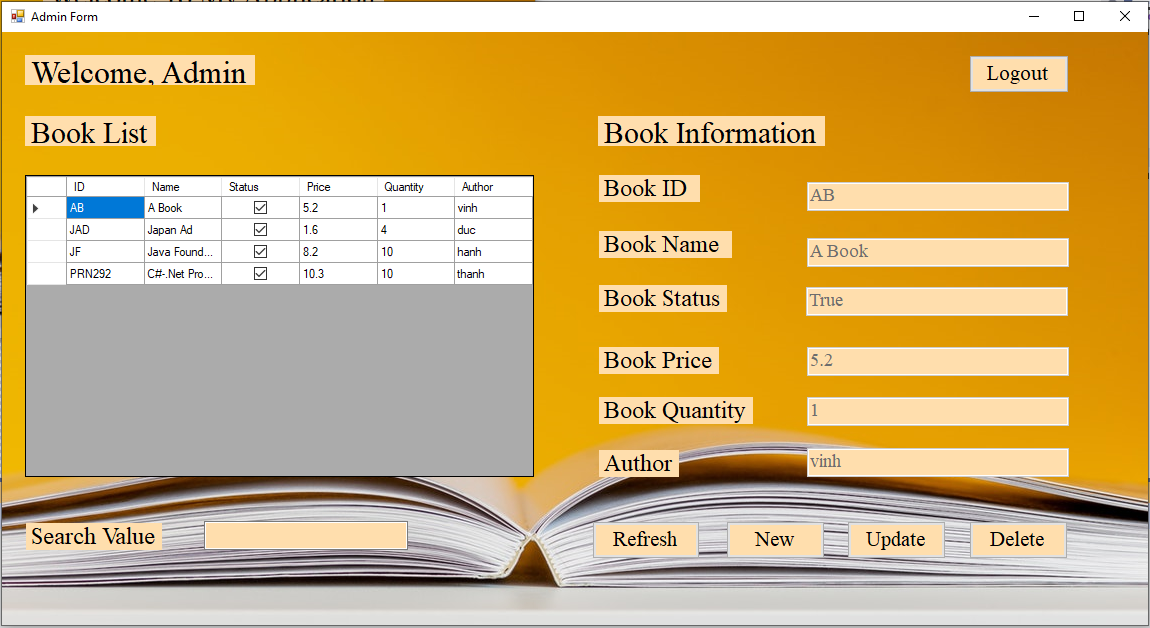
**Interface Design**

**1. Login Form:**



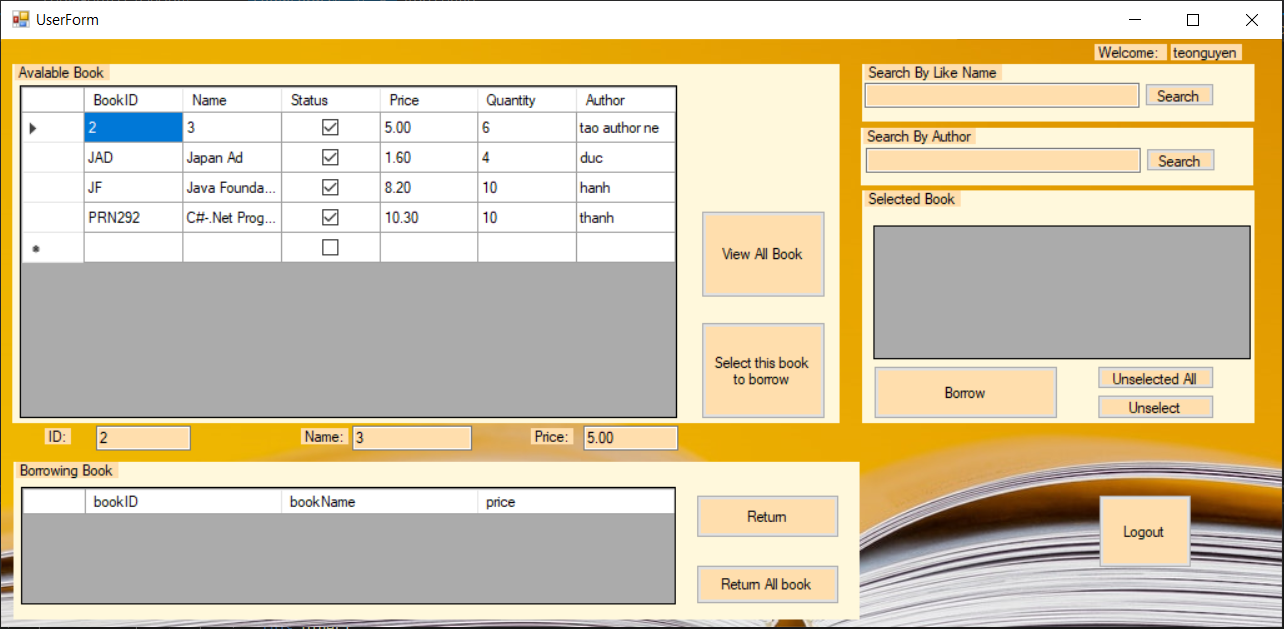
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Event | Validation | Description |
| LblTitle | Label |  |  | Show data |
| LblUsername | Label |  |  | Show data |
| LblPassword | Label |  |  | Show data |
| TxtUsername | Textbox |  | String, not null | Input username |
| TxtPassword | Textbox |  | String, not null | Input password |
| BtnLogin | Button | Click |  | Log in |
| BtnReset | Button | Click |  | Click data in textboxes |

**2. Admin Form**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Event | Validation | Description |
| lblWelcomeAdmin | Label |  |  | Show data |
| lblBookList | Label |  |  | Show data |
| lblBookInformation | Label |  |  | Show data |
| lblBookID | Label |  |  | Show data |
| lblBookName | Label |  |  | Show data |
| LblBookStatus | Label |  |  | Show data |
| LblBookPrice | Label |  |  | Show data |
| LblBookQuantity | Label |  |  | Show data |
| LblAuthorName | Label |  |  | Show data |
| lblSearchValue | Label |  |  | Show data |
| DgvShowBook | Data Grid View |  |  | Show all information of book |
| TxtBookID | Textbox |  | String, not null | Input book id |
| TxtBookName | Textbox |  | String, not null | Input book name |
| TxtBookStatus | Textbox |  | String, not null | Input book status |
| TxtBookPrice | Textbox |  | Double, not null | Input book price |
| TxtBookQuantity | Textbox |  | Int, not null | Input book quantity |
| TxtAuthorName | Textbox |  | String, not null | Input author |
| TxtSearchValue | Textbox | Text changed | String | Search data |
| BtnRefresh | Button | Click |  | Refresh all data, table, textboxes |
| BtnNew | Button | Click |  | New a book |
| BtnUpdate | Button | Click |  | Update information book |
| BtnDelete | Button | Click |  | Delete a book |
| BtnLogout | Button | Click |  | Logout account admin |

1. **User Form:**



|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Even | Description |
| dvgAvalable | Data Grid View |  | Show all information of ready book |
| dvgSelected | Data Grid View |  | Show id, name, price of selected book |
| dvgBorrowing | Data Grid View |  | Show the book user is borrowing |
| lbUser | Label |  | Show fullname of user is logging in |
| txtID | TextBox |  | Show id of book is selecting on available book |
| txtName | TextBox |  | Show name book is selecting on available book |
| txtPrice | TextBox |  | Show price book is selecting on available book |
| txtSearchLikeName | TextBox |  | For user input the name of book to search |
| txtSearchAuthor | TextBox |  | For user input the author of book to search |
| btnViewAllBook | Button |  | View all book after search somethings |
| btnSelect | Button | Click | For user to select their book |
| btnSearchLikeName | Button | Click | Click this button to see what book have the name like what you input at txtSearchLikeName |
| btnSearchAuthor | Button | Click | Click this button to see what book have the author like what you input at txtSearchAuthor |
| btnBorrow | Button | Click | Click this button to borrow all book you have selected |
| btnUnselectedAll | Button | Click | Click this button to unselected all the book in selected table. |
| btnUnselected | Button | Click | Click this button to unselected that the book you are selecting in the “Seleted Book” table. |
| btnReturn | Button | Click | Click this button to return the book you are selecting in the “borrowing Book” table |
| btnReturnAll | Button | Click | Click this button to return all of book in “borrowing book” table |