

Library System Project

Three Users --> Librarian, Admin, Both (Super User with all privileges)

Basic Use cases

Librarian

--> Checkout Books

Admin

--> Add/ Edit/ Delete / Update Library members

--> Add new books to the collection

--> Add copies to the existing collection

Both

--> Checkout Books

--> Add/ Edit/ Delete / Update Library members

--> Add new books to the collection

--> Add copies to the existing collection

Book --> Book Copy – Bidirectional Relationship

Example:

Java 8 Fundamentals - 10 copies

CheckoutRecord for the Library member

CheckoutRecord --> Multiple CheckoutEntries

1005 - Member id

Checkout - Two books

Checkout Entry 1

1005, Book Copy 1 (Title, Copy num, due date), Book details

Checkout Entry 2

1005, Book Copy 2 (Title, Copy num, due date), Book details

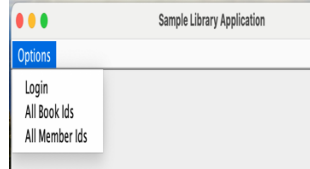
Default User credentials from the Project startup code

"101", "xyz", Auth.LIBRARIAN

"102", "abc", Auth.ADMIN

"103", "111", Auth.BOTH

Startup Code Layers

| Business | Data access | Storage | UI Layer |
|--|---|---|--|
| <ul style="list-style-type: none"> business <ul style="list-style-type: none"> Address.java Author.java Book.java BookCopy.java ControllerInterface.java LibraryMember.java LibrarySystemException.java LoginException.java Main.java Person.java SystemController.java <p>Need to add CheckoutRecord, CheckoutEntry in the Business layer.</p> | <ul style="list-style-type: none"> dataaccess <ul style="list-style-type: none"> Auth.java DataAccess.java DataAccessFacade.java TestData.java User.java | <ul style="list-style-type: none"> dataaccess.storage <ul style="list-style-type: none"> BOOKS MEMBERS USERS | <ul style="list-style-type: none"> librarysystem <ul style="list-style-type: none"> AllBookIdsWindow.java AllMemberIdsWindow.java LibrarySystem.java LibWindow.java LoginWindow.java Main.java Util.java  |

How Library System Interacts with different layers.

ControllerInterface can have the behaviours related to Usecases.

```
public void login(String id, String password)
public List<String> allMemberIds();
public List<String> allBookIds();
```

These methods must implements in the SystemController class.

How UI layer interact with the Business layers

Inside SystemController

@Override

```
public List<String> allMemberIds() {
    DataAccess da = new DataAccessFacade();
    List<String> retval = new ArrayList<>();
    retval.addAll(da.readMemberMap().keySet()); // Retrieve member ids
    return retval;
}
```

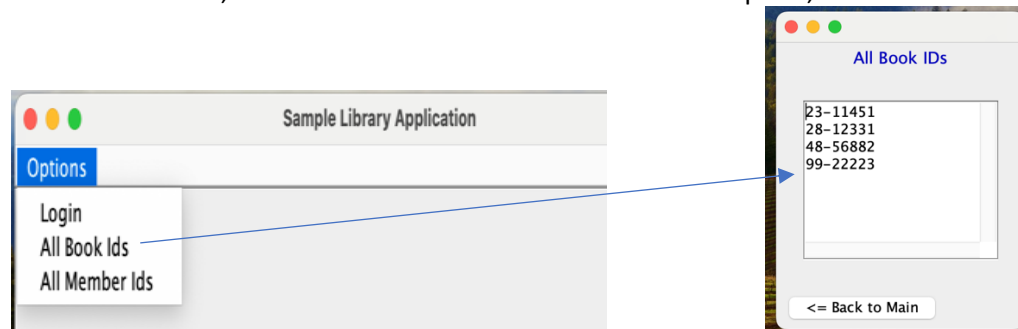
DataAccess is an Interface, DataAccessFascade implements it and give all the database related operations.

Inside DataAccessFascade.java

```
public HashMap<String, LibraryMember> readMemberMap() {  
    //Returns a Map with name/value pairs being  
    // memberId -> LibraryMember  
    return (HashMap<String, LibraryMember>) readFromStorage(  
        StorageType.MEMBERS);  
}
```

If the SystemController need a DataAccess, contact it through DataAccessFascade. [Connection between Business Controller to DataAccess.

From the Main UI, If the user selects the All Books Ids menu option, it will show all the Book Ids.



UI Layer contacts the ControllerInterface to interact DataAccess and Business layers.

Inside LibrarySystem(UI Layer), clicking the All Book ids menu item

```
ControllerInterface ci = new SystemController();  
  
public void actionPerformed(ActionEvent e) {  
    AllBookIdsWindow.INSTANCE.init();  
  
    List<String> ids = ci.allBookIds();  
    Collections.sort(ids);  
    StringBuilder sb = new StringBuilder();  
    for(String s: ids) {  
        sb.append(s + "\n");  
    }  
    System.out.println(sb.toString());  
    AllBookIdsWindow.INSTANCE.setData(sb.toString());  
}
```

Inside AllBookIdsWindow(UI Window)

```
public void setData(String data) {  
    textArea.setText(data);  
}
```

Video Links for Java Swing and Project

1. Java Swing and Project Description

<https://mum0.sharepoint.com/:v:/s/CS401-2022-12A-12D-01/EbsZEVlbaLJlt4mJhLgqPhABoV1AMssUi7XYtoy6Z3CEnQ?e=Xoo5Ds>

2. Project Use cased Explanation.

https://mum0.sharepoint.com/:v:/s/CS401-2022-12A-12D-01/EdDpuF4jKR5CtjCGIzq9cl4B_GB0nBzA5ttBIF2uNQjXTg?e=VuvvNi

3. JTable(CRUD) Demo and Project Startup code Explanation

https://mum0.sharepoint.com/:v:/s/CS401-2022-12A-12D-01/EVhGz7tpPolDtkBbzHaxNvgBGp_RsW-FyqlRuzH7jzHXMw?e=M6wq6k

Here are the demos help your Library project available under Sakai Resources, Democode & Project

1. If you want to keep your window center of your screen, use this code:
setLocationRelativeTo(null);
2. If you want to work with list, to select ids, isbn, authours etc., **Refer: JList Packages**
3. If you want to how to work with Menus, Tables, working with multiple screens
Refer: GUI Package
4. If you want to design a Split screen design view, **Refer: splitpane package**
5. How to work with Table and its CRUD operation **Refer: bookdemo package**