BBM 487 – SOFTWARE ENGINEERING LABORATORY

LIBRARY BOOK LOAN SYSTEM

System-Wide Requirements Specification

GROUP II

Özlem DEMİRTAŞ	**21327901**
Umut ÖZTÜRK	**21328394**
Rahmi Berk ŞEFKATLİ	**21427402**

Library Book Loan System

System-Wide Requirements Specification

1. Introduction

This is the System-Wide Requirements Specification for the Library Book Loan System. The purpose of this document is to convey information about the application's requirements, both functional and non-functional, to the reader. This document provides:

a. A description of the environment in which the application is expected to operate.

b. A definition of the application's capabilities.

c. A specification of the application's functional and non-functional requirements.

1.1 Purpose

The Library Book Loan System is a package to be used by libraries to improve the efficiency of librarians and users.

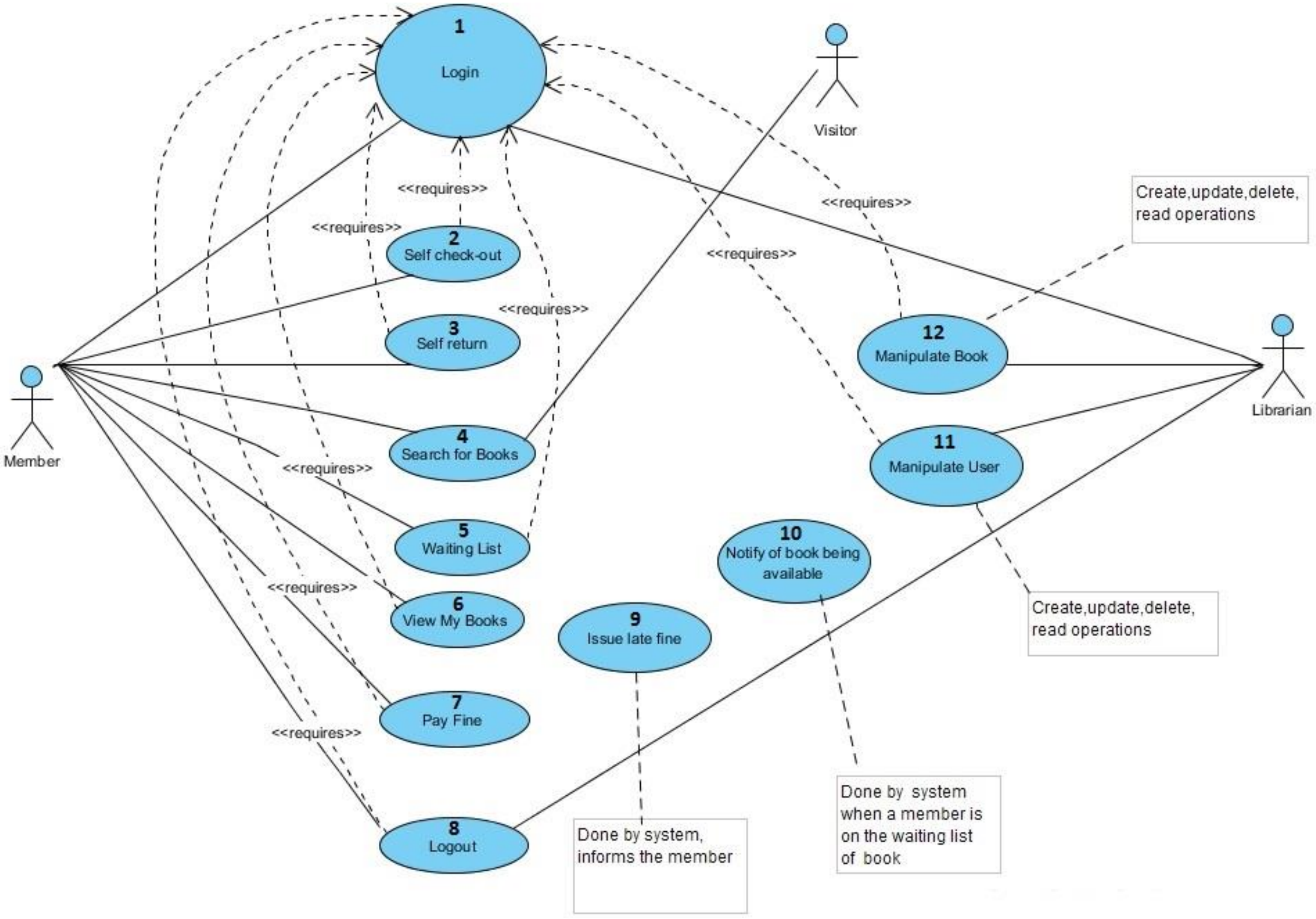
The main purpose of the system is to provide a friendly environment to maintain the details of books and library members.

1.2 Scope

The document only covers the requirements specifications for the Library Book Loan System.

All the external interfaces and the dependencies are also identified in this document.

2. System-Wide Functional Requirements



3. System Qualities

3.1 Usability

The system is designed for a user friendly environment so that student and staff of library can perform the various tasks easily and in an effective way.
Especially, the user interface will be simple and easy to understand and use. In addition, a small handbook will be prepared for users after the project.

3.2 Reliability

The system has to be very reliable due to the importance of data and the damages incorrect or incomplete data can do.
The system will accurately perform member registration, member validation, book transaction and search.

3.3 Performance

a. Response time

The system will respond to the member in less than two seconds from the time of the request submittal. The system will be allowed to take more time when doing large processing jobs.

3.4 Supportability

a. Maintainability

Database and user interface will be maintained periodically.
Changes will be verified once per day at least.

b. Documentation

The user manual will be provided to inform the user about system update.

4. System Interfaces

4.1 User Interfaces

User interfaces are developed to meet the requirements of project. In other words, they communicate between the software of project and users. These user interfaces are defined in Apdx-B document.

4.1.1 Look & Feel

The interface is clean and functional. Which makes it easier for everyone to use. It is supported with images.

4.1.2 Layout and Navigation Requirements

Navigating to the previous pages are done with the buttons on the top right corner of the window while navigating to the next pages are usually near the center.

4.1.3 Consistency

Having similar working buttons placed consistently at the same places in the window makes it easier for users to predict and adapt.

4.1.4 User Personalization & Customization Requirements

Member users can see Search for Books, View My Books and Pay Fine buttons in their main menu, while librarian users can see Manipulate Users and Manipulate Books buttons. In addition, visitor users can only search for books but members can also self-checkout or add name to the waiting list from the same page.

4.2 Interfaces to External Systems or Devices

4.2.1 Software Interfaces

The Library Book Loan System will be implemented using database management system. Therefore,

- Navicat will be used to manage the tables related project objects. It is a powerful database management and design tool.
- Java Swing will be used to develop the desktop application.
- Eclipse IDE will be used to code easily.
- MySQL will be used to persist/store the objects related to our project such as librarian, users. However, JDBC API is used to communicate between MySQL and Java Swing. The JDBC ‘s jar will be added to the class-path of project.

In shortly, Eclipse IDE, JDBC, MySQL and Java Swing are used to develop this project. In addition, we will use an external payment system to support paying fines.

4.2.2 Hardware Interfaces

There is no hardware interface in our system.

4.2.3 Communications Interfaces

There are no interfaces for communication.

5. Business Rules

5.1 Loaning Books

5.1.1 Maximum Number of Books to Loan

Each member can loan at most three books at any given time.

5.1.2 Book Loan Period

Each book can be loaned for at most 30 days.

5.2 Fine Management

5.2.1 When to Fine

When a member loans a book but does not return it within 30 days.

5.2.2 How to Calculate Fine

Fine is calculated after the book loan period is expired. For each day the book is not returned, member will be fined 5 TL.

6. System Constraints

a. The application will be operating in windows environment.

b. It will be used only as a desktop application.

c. It will be developed using Java.

d. The backend database will be SQL Server.

e. Also the application needs Eclipse, JDBC and MySQL.

7. System Compliance

7.1 Licensing Requirements

The source code and documentation are freely available to the public.

7.2 Legal, Copyright, and Other Notices

Only non-commercial redistribution or modification of the source code for personal use is allowed.

7.3 Applicable Standards

The project “Library Book Loan System” follows Java coding standards.

8. System Documentation

The product will include user manual. The user manual will include product overview, complete configuration of the used software, technical details, backup procedure and contact information which will include email address. There will be no online help for the product at this moment.

Also, a Read Me file is typically included as a standard component. The Read Me includes a “What’s New With This Release” section, and a discussion of compatibility issues with earlier releases. Most users also appreciate documentation defining any known bugs and workarounds in the Read Me file.

APPENDIX A

Use case 1: Login	
Actors	Librarian, Member
Pre-Condition	1. If the librarian or member wants to login then he/she should have been first registered.
Post-Condition	1. Librarian or member is logged in.
Main (happy) path	1. To login the librarian or member should first open the Login page. 2. They should then enter their usernames and passwords. 3. Once the username and password are verified, they are moved to the main menu where they can perform other operations.
Alternative path	1. The login fails after entering user name and password. 2. The system will show a message indicating it. 3. User will be able to enter another user name and password.

Use case 8: Logout	
Actors	Librarian, Member
Pre-Condition	1. The user is logged in.
Post-Condition	1. The user is logged out.
Main (happy) path	1. The user clicks on the logout button. 2. The system logs the user out and invalidates the session. 3. The system redirects to the login page.
Alternative path	N/A

Use case 12: Manipulate book (Add, delete and update)	
Actors	Librarian
Pre-Condition	1. Librarian must be logged in. 2. To delete and update the book, the book must be in the library.
Post-Condition	1. The new book record is added to the system. 2. The old book record is removed from the system. 3. The old book record is updated.
Main (happy) path	1. To add record, librarian selects the menu option to add record of the book. 2. Enters the name, Author's name, barcode, printing date, number of pages. 3. Adding record is successful. 4. To delete the record, librarian selects the book from the list and presses Delete Book button. 5. Deleting record is successful. 6. To update the record, librarian selects the book from the list and presses Edit Book button. 7. Librarian updates the specific record. 8. Updating record is successful.
Alternative path	1. When updating the barcode of a book, if there is a book with the same barcode system will not allow it. 2. Librarian must enter a unique barcode or cancel the editing process.

Use case 11: Manipulate user (Add, delete and update)	
Actors	Librarian
Pre-Condition	1. The member to be deleted or updated should have been added in the database of the library management system. 2. Librarian must be logged in.
Post-Condition	1. The new member record is added to the system. 2. The old member record is removed from the system. 3. The old member record is updated.
Main (happy) path	1. To add member, librarian selects the menu option to create new member. 2. Enters the username, name, surname, password and clicks the create new member button. 3. Adding member is successful. 4. To delete a member, librarian selects the member from the list and presses Delete Member button. 5. Deleting record is successful. 6. To update the member record, librarian selects the member from the list and presses Edit Member button. 7. Librarian updates the specific member record. 8. Updating member record is successful.
Alternative path	1. If the new member record is already in the database, librarian can't add it again. 2. Librarian must enter a unique member record or cancel the process.

Use case 5: Add name to book waiting list	
Actors	Member
Pre-Condition	1. Member must be logged in. 2. The book that the member wants to borrow should have been borrowed by another user.
Post-Condition	1. The member adds her/his name to waiting list to borrow the book when it is available.
Main (happy) path	1. Member searches for specific book. 2. Member finds that the book he/she is searching is taken by someone else. 3. Member clicks the add name to waiting list button. 4. System adds member's name to the list.
Alternative path	N/A

Use case 4: Search for books	
Actors	Member, visitor
Pre-Condition	N/A
Post-Condition	1. The user finds the specific book(s).
Main (happy) path	1. Member or visitor enters the book name or barcode or author name and presses search. 2. If the search is successful then that book is displayed on the screen.
Alternative path	1. If the searched book is not found, user may search for another book.

Use case 6: View my books	
Actors	Member
Pre-Condition	1. Member must be logged in.
Post-Condition	1. Member views her/his books which are borrowed from the library.
Main (happy) path	1. Member clicks on the view my books button. 2. Records are found and displayed on the screen.
Alternative path	1. If the member did not borrow any book from the library, then book(s) cannot be viewed.

Use case 7: Pay fine	
Actors	Member
Pre-Condition	1. Member must be logged in. 2. Member borrows book from the library and does not return book on time.
Post-Condition	1. Member pays fine.
Main (happy) path	1. Member clicks the pay fine button and uses credit card. 2. Member enters credit card password and pays fine.
Alternative path	1. If the credit card number is invalid, an error message displays to the user and the user is asked to re-enter the credit card information or cancel the payment process.

Use case 3: Self-return	
Actors	Member
Pre-Condition	1. Member must be logged in. 2. Member should have loaned a book before.
Post-Condition	1. Member returns the book. 2. Member can see returned books in View My Books page.
Main (happy) path	1. Member chooses the book he/she wants to return from the View My Books page. 2. User clicks the self-return button. 3. System marks the books as returned and the book can be seen as returned in the list.
Alternative path	N/A

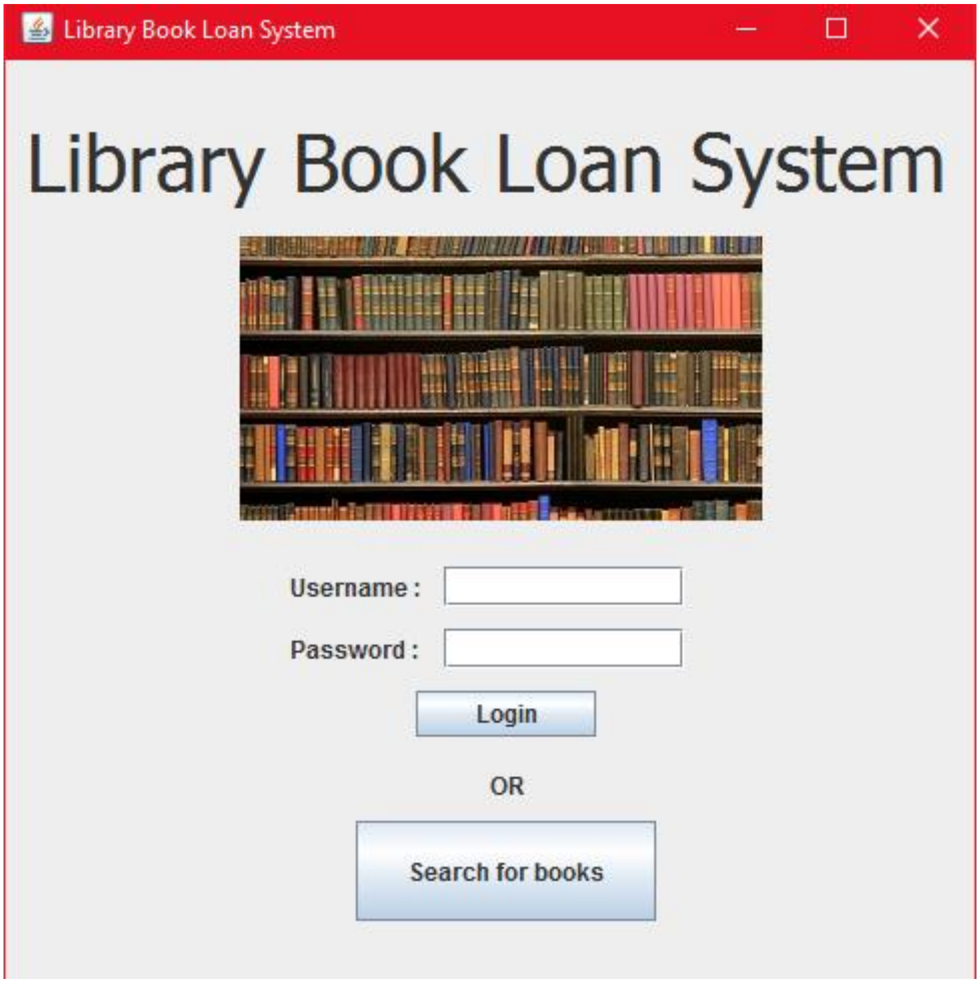
Use case 2: Self-checkout	
Actors	Member
Pre-Condition	1. Member must be logged in. 2. The book user wants to loan should be available.
Post-Condition	1. Member loans the book. 2. Loaned book can be seen in the View My Books page.
Main (happy) path	1. Member chooses the book he/she wants to loan from the Search for Books page. 2. Member clicks the self-checkout button. 3. Member agrees to the 30 days rule. 4. System marks the books as returned and the book can be seen as returned in the list.
Alternative path	1. Member does not agree to the 30 days rule. 2. Member is redirected back to the Search for Books page.

Use case 10: Notify of book being available	
Actors	Member
Pre-Condition	1. Member must have queued for a previously unavailable book.
Post-Condition	1. Member gets notified that the book is now available for loaning.
Main (happy) path	1. When a member returns a book. System checks if anybody is waiting for that book in the waiting list. 2. If there is one member waiting for the book, he/she will be notified. 3. Member will be able to self-checkout the book.
Alternative path	1. If there are more than one members waiting for the same book, then the first one in the list will get the notification and will be able to self-checkout.

Use case 9: Issue late fine	
Actors	Member
Pre-Condition	1. Member’s book loan period must be expired.
Post-Condition	1. Member gets fined.
Main (happy) path	1. System checks if any member’s book loan has expired, every day. 2. When it detects such a member, system will count the days until the book is returned. 3. System multiplies the days with 5 TL and specifies the total amount. 4. Member is expected to pay the fine.
Alternative path	1. There are no members with expired loan periods. 2. System keeps checking every day.

APPENDIX B

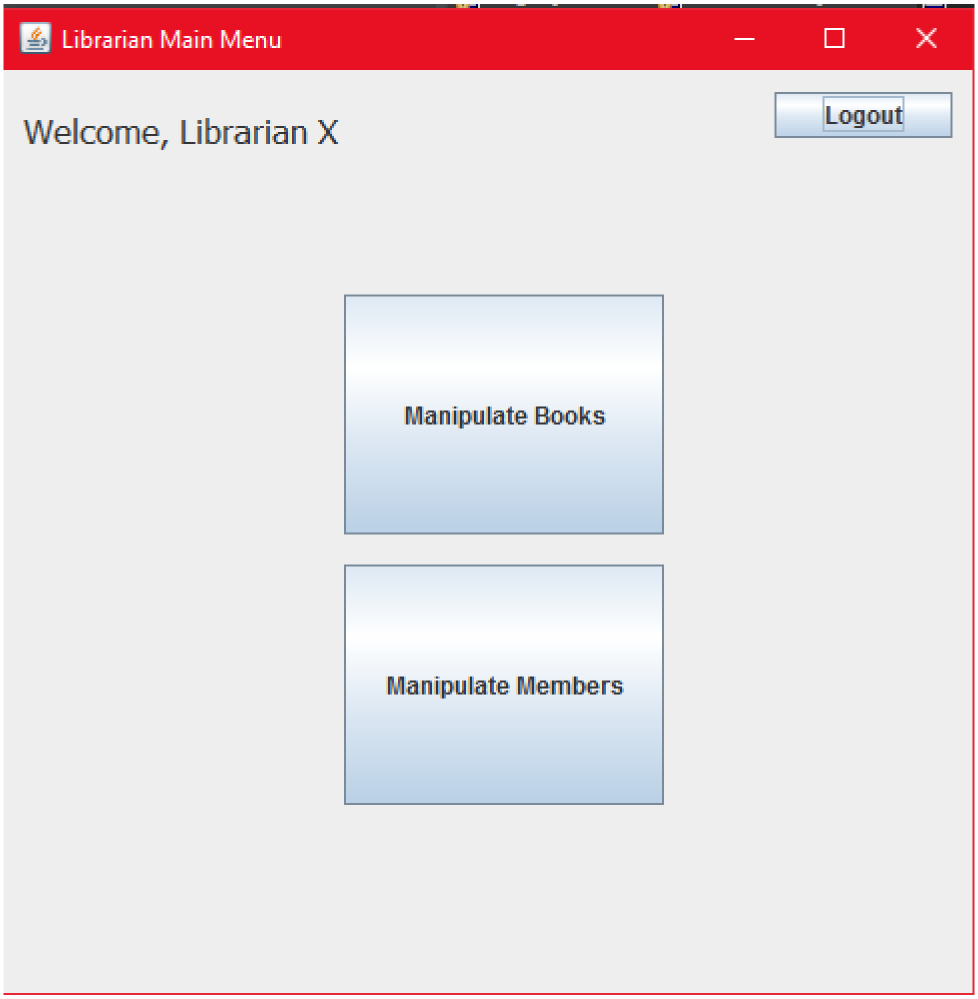
User Interface



Login page

Related use cases:

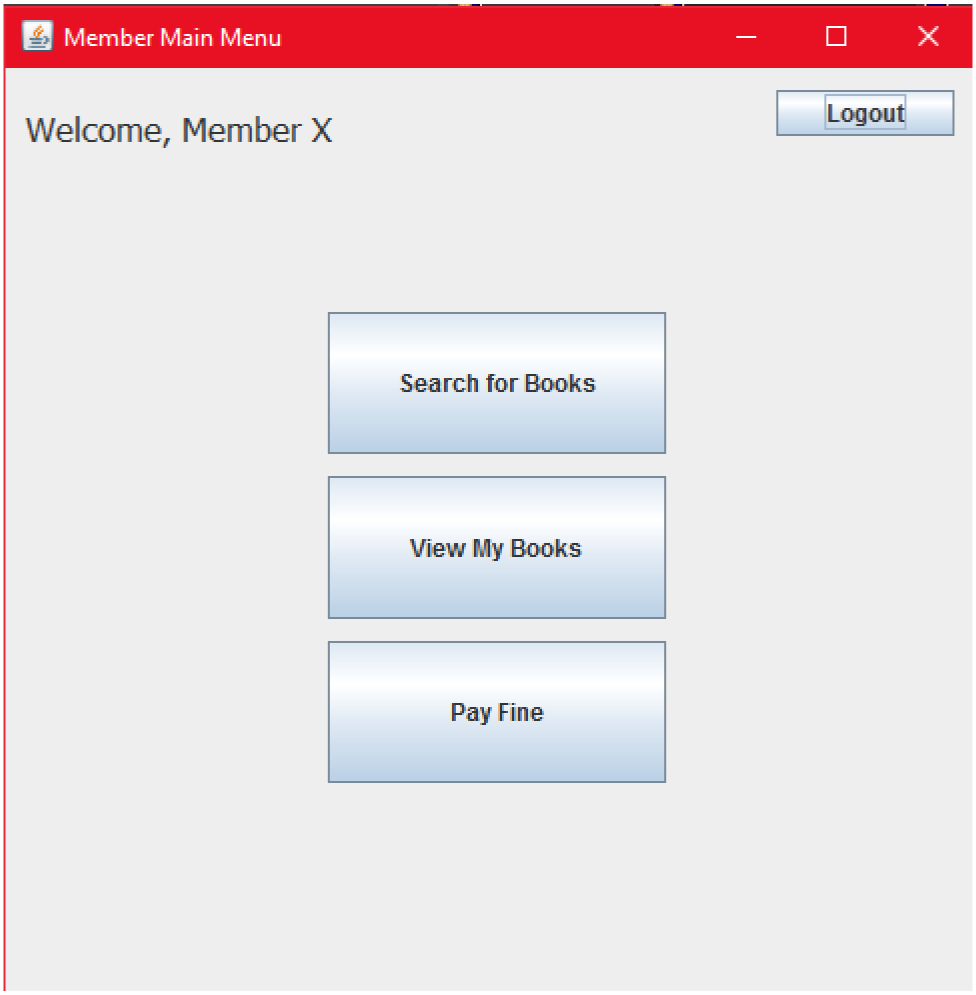
- **Login**
- **Search for books**



Main menu page for librarians

Related use cases:

- **Manipulate Books**
- **Manipulate Members**
- **Logout**

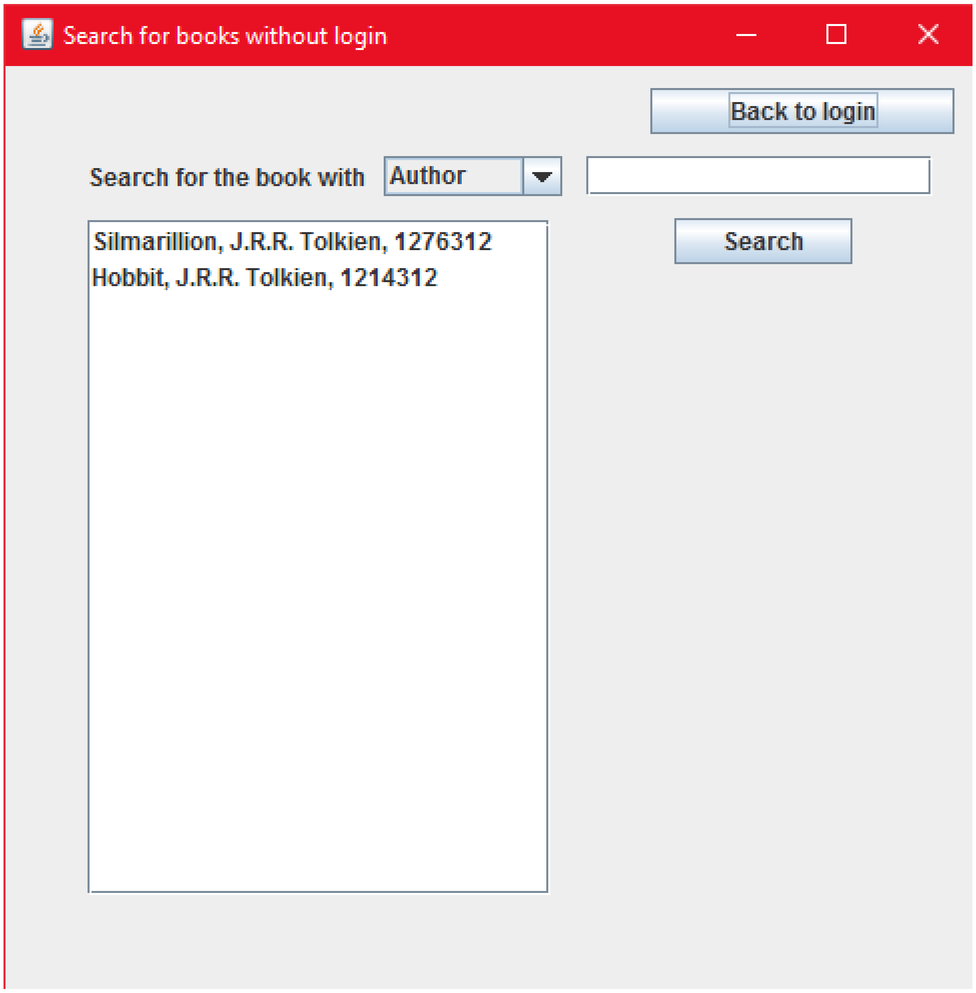


Welcome, Member X

Main menu page for members

Related use cases:

- **Search for books**
- **View my books**
- **Pay fine**
- **Logout**

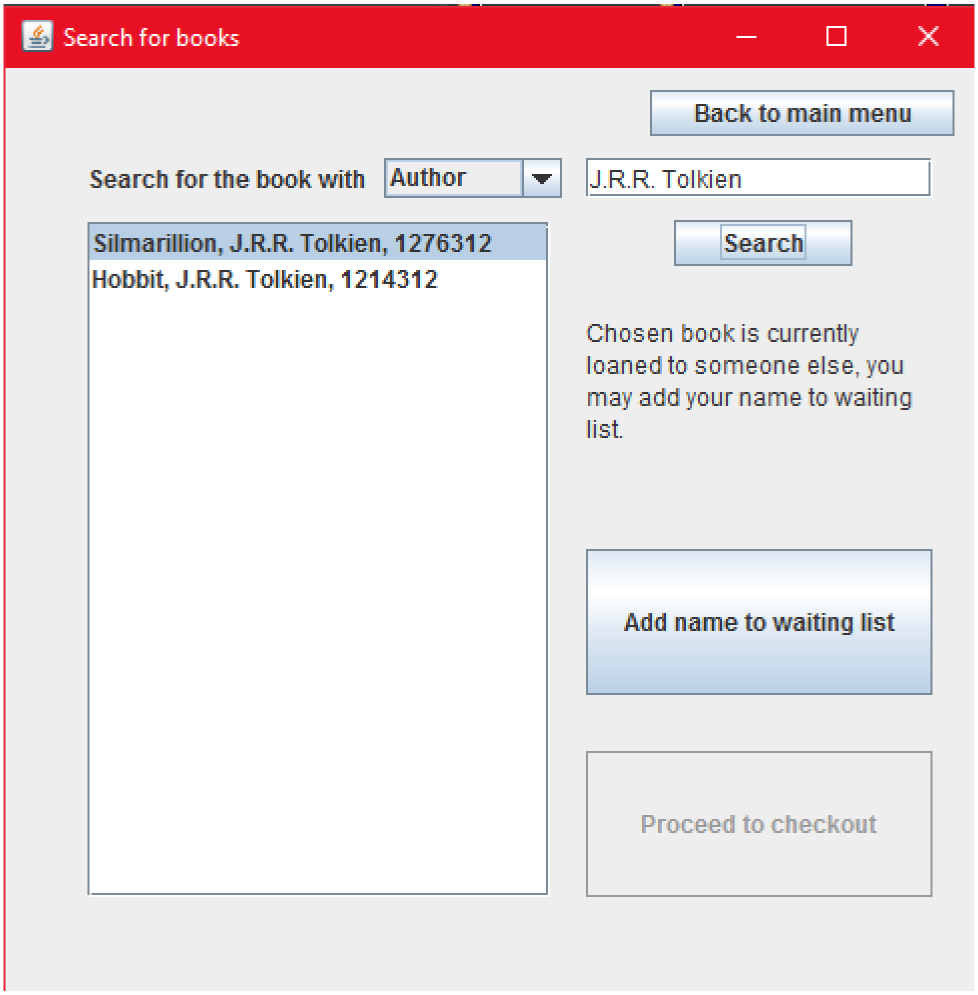


Silmarillion, J.R.R. Tolkien, 1276312
Hobbit, J.R.R. Tolkien, 1214312

Search for books page for visitors

Related use cases:

- **Search for books**



Search for the book with Author J.R.R. Tolkien

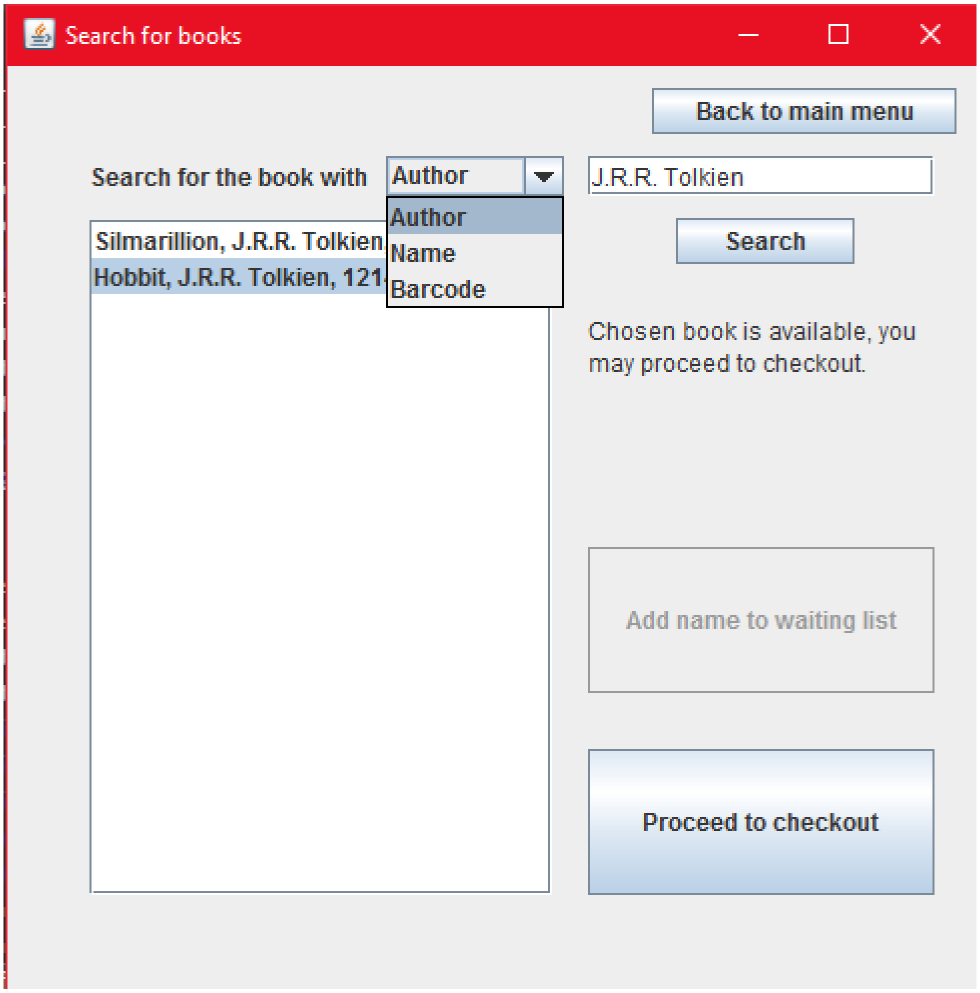
Silmarillion, J.R.R. Tolkien, 1276312
Hobbit, J.R.R. Tolkien, 1214312

Chosen book is currently loaned to someone else, you may add your name to waiting list.

Search for books page for members

Related use cases:

- **Search for books**
- **Add name to waiting list**
- **Self-checkout**



Search for the book with Author

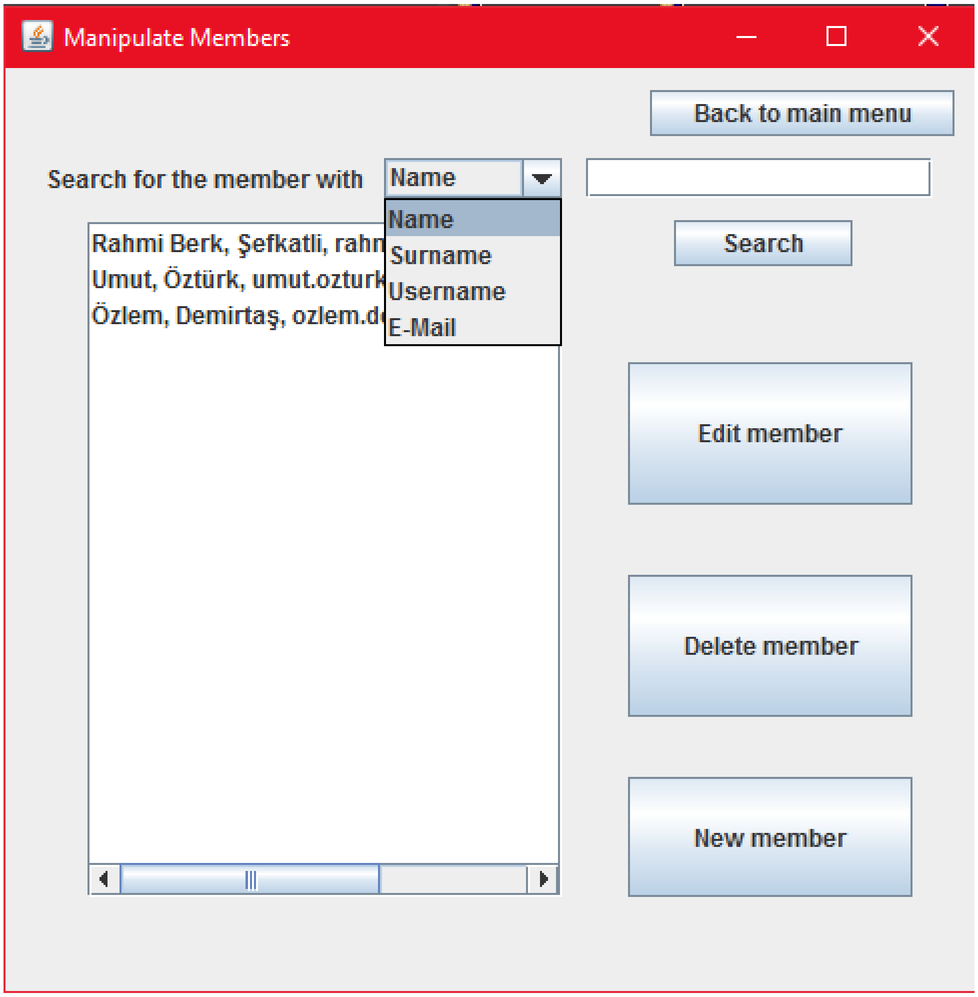
J.R.R. Tolkien

Silmarillion, J.R.R. Tolkien

Hobbit, J.R.R. Tolkien, 121

Chosen book is available, you may proceed to checkout.

Search for books page for members cont.

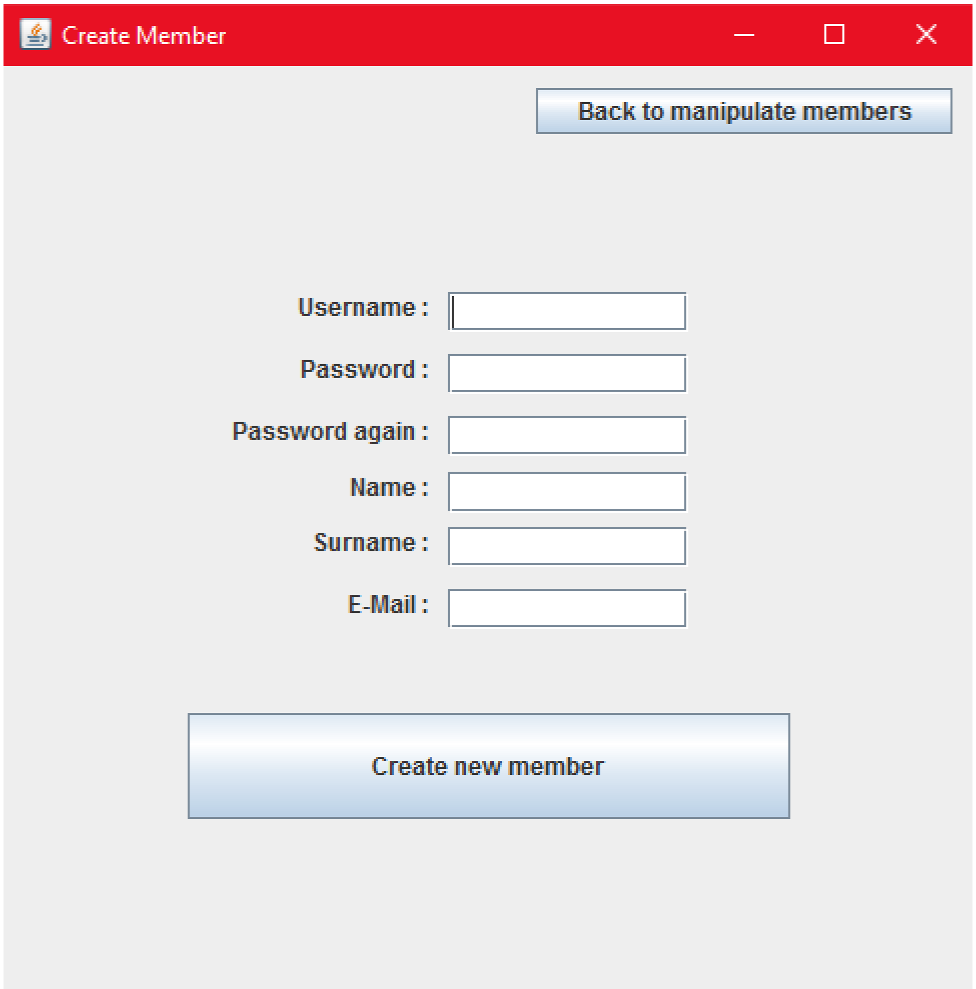


Rahmi Berk, Şefkatli, rahn
Umut, Öztürk, umut.ozturk
Özlem, Demirtaş, ozlem.d

Manipulate members page for librarians

Related use cases:

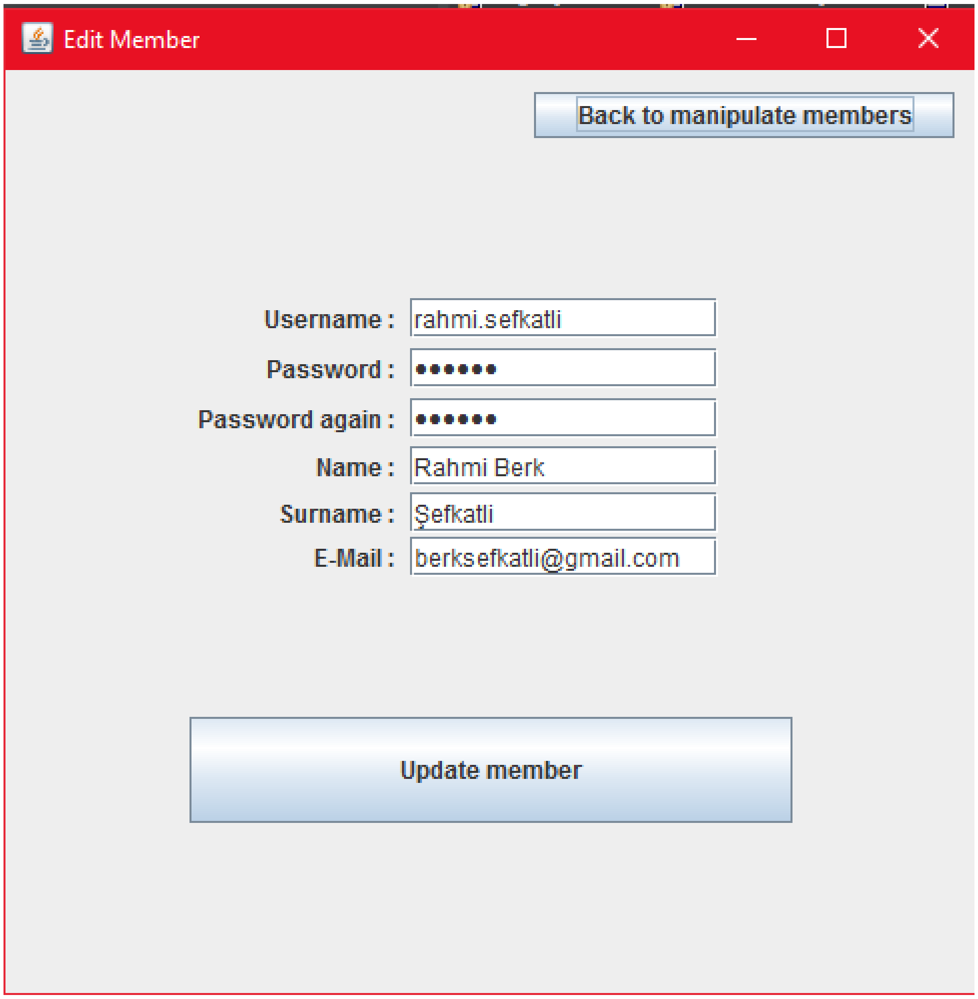
- **Manipulate Members(View, Edit, Delete, Create)**



Create member page for librarians

Related use cases:

- **Manipulate Members(Create)**



Username : rahmi.sefkatli

Password : ••••••

Password again : ••••••

Name : Rahmi Berk

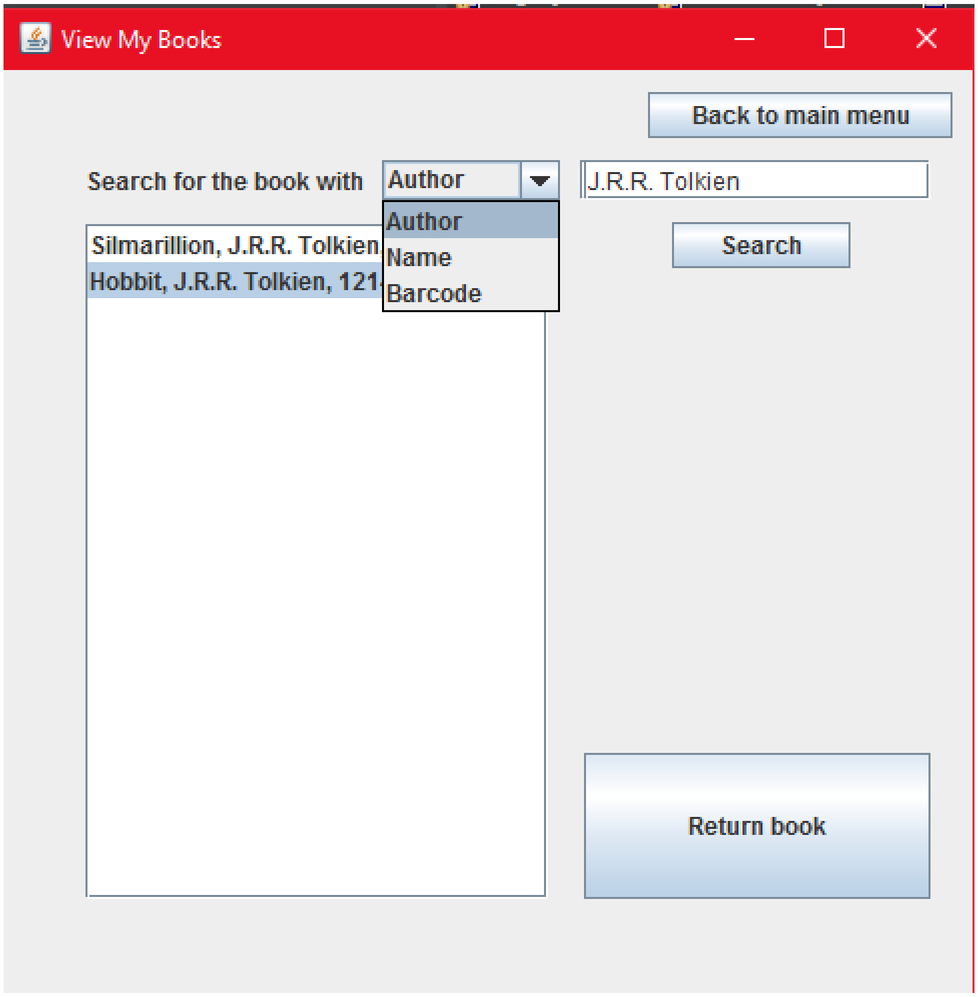
Surname : Şefkatli

E-Mail : berksefkatli@gmail.com

Update member page for librarians

Related use cases:

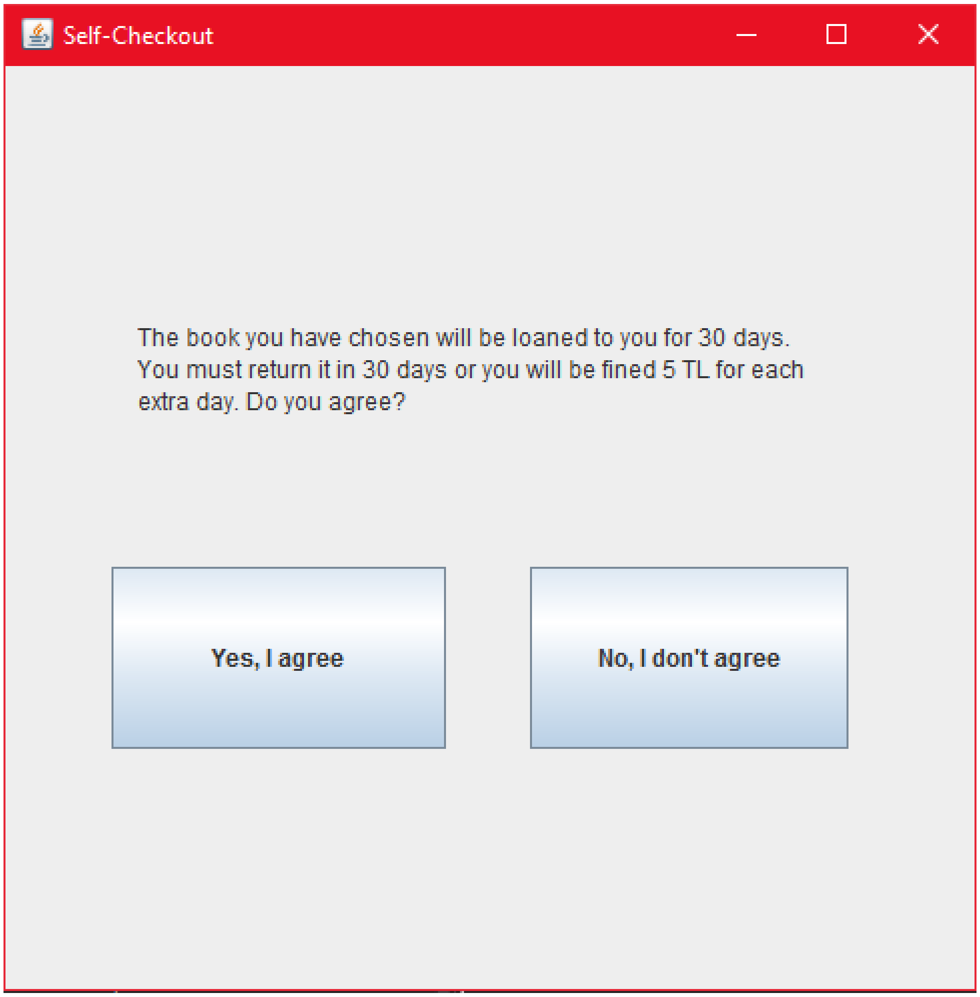
- **Manipulate Members(Edit)**



View my books page for members

Related use cases:

- **View my books**
- **Self-return**

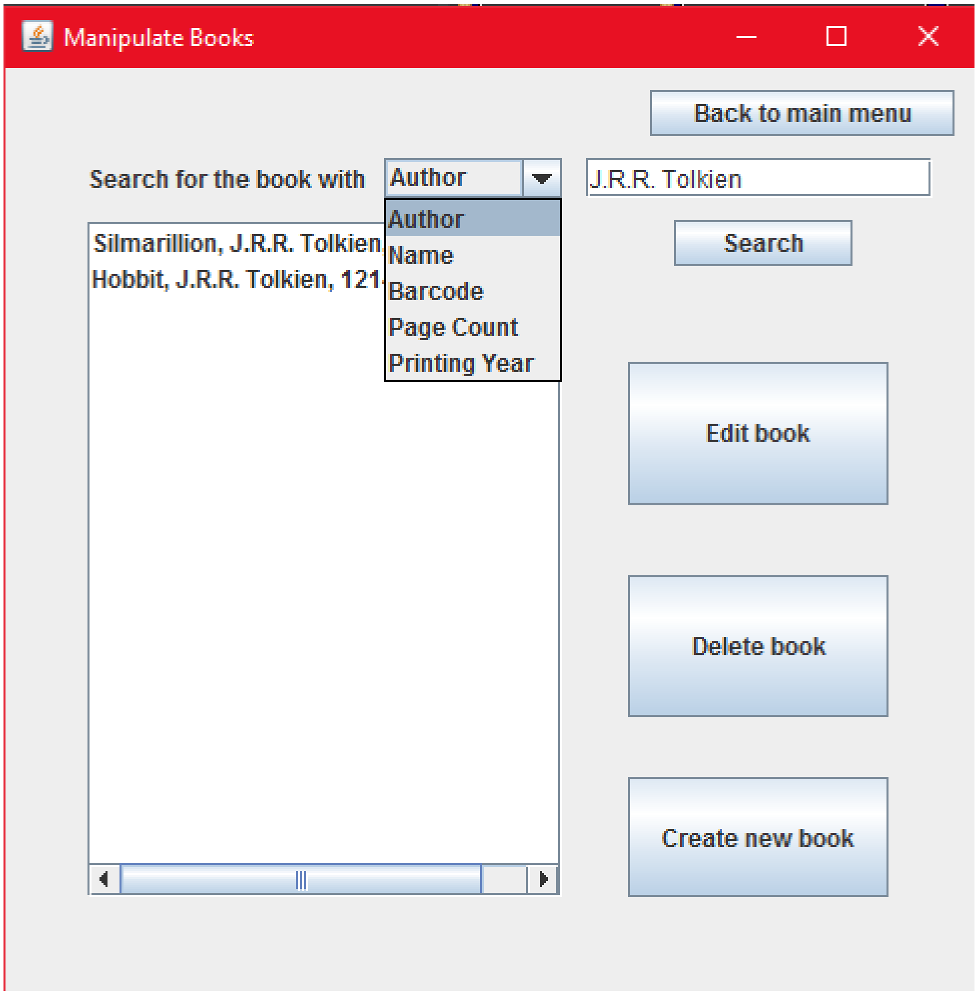


The book you have chosen will be loaned to you for 30 days. You must return it in 30 days or you will be fined 5 TL for each extra day. Do you agree?

Self-checkout page for members

Related use cases:

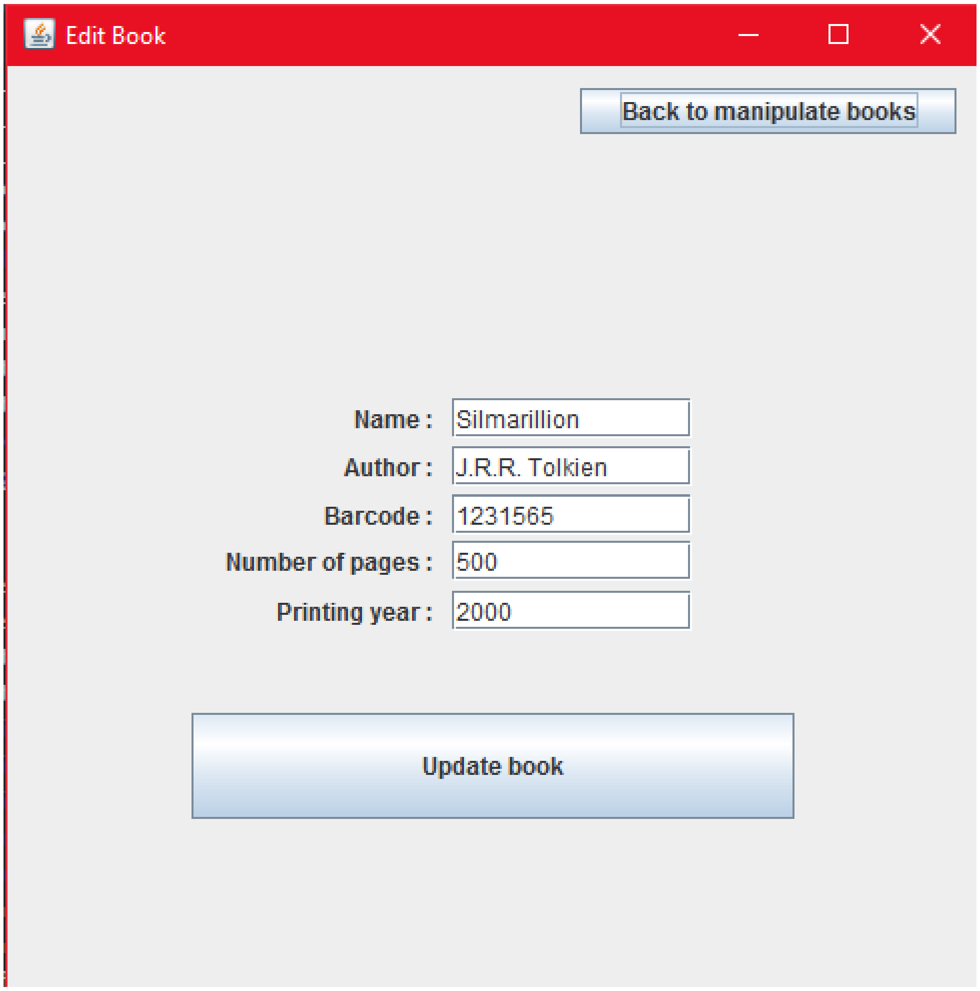
- **Self-checkout**



Manipulate books page for librarians

Related use cases:

- **Manipulate Books(View, Edit, Delete, Create)**



Name : Silmarillion

Author : J.R.R. Tolkien

Barcode : 1231565

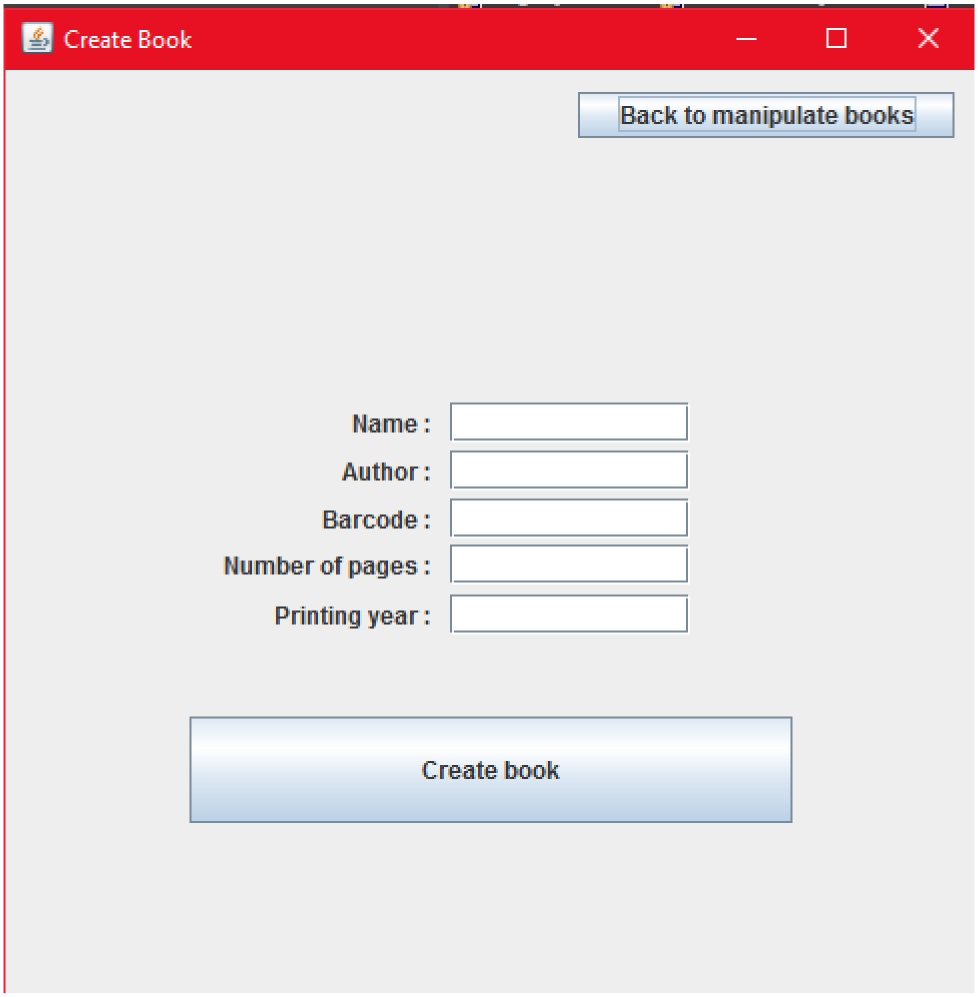
Number of pages : 500

Printing year : 2000

Edit book page for librarians

Related use cases:

- **Manipulate Members(Edit)**



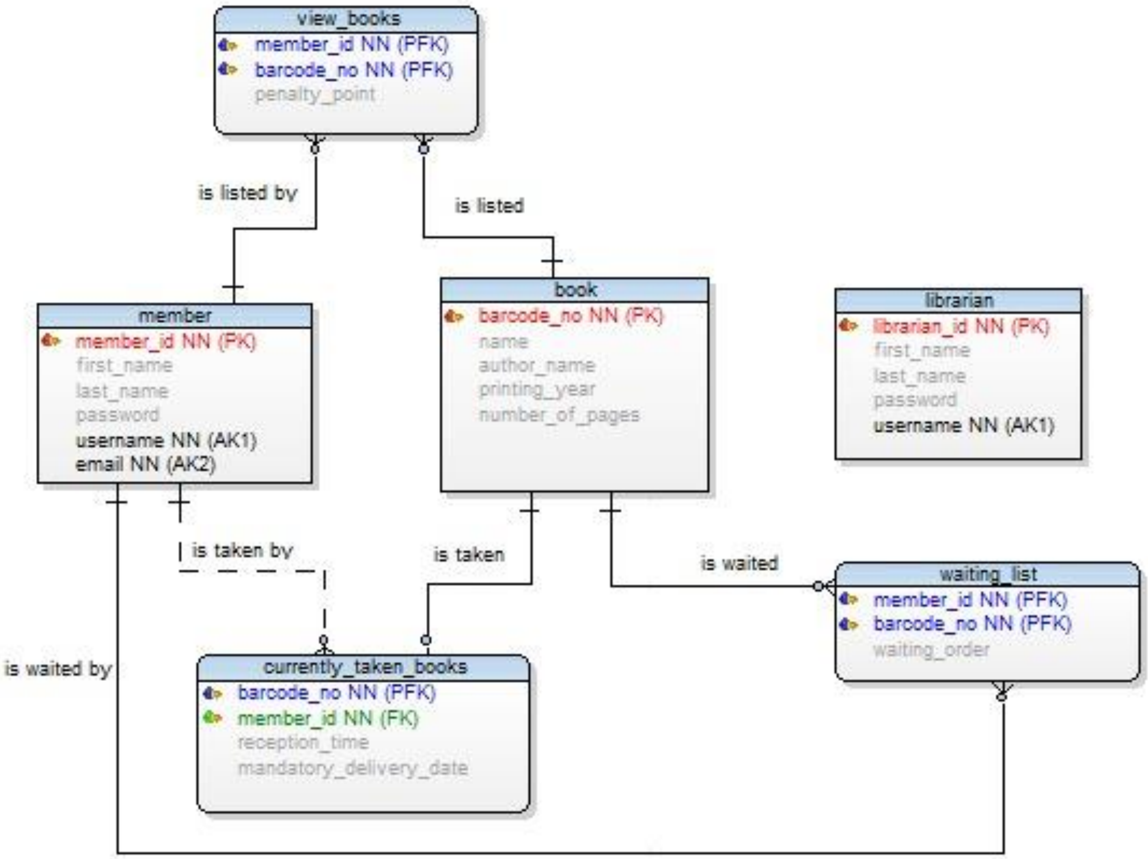
Create book page for librarians

Related use cases:

- **Manipulate Members(Create)**

APPENDIX C

ER Diagram



view_books
member_id NN (PFK)
barcode_no NN (PFK)
penalty_point
is listed by
is listed
member
member_id NN (PK)
first_name
last_name
password
username NN (AK1)
email NN (AK2)
book
barcode_no NN (PK)
name
author_name
printing_year
number_of_pages
librarian
librarian_id NN (PK)
first_name
last_name
password
username NN (AK1)
is taken by
is taken
is waited
waiting_list
member_id NN (PFK)
barcode_no NN (PFK)
waiting_order
is waited by
currently_taken_books
barcode_no NN (PFK)
member_id NN (FK)
reception_time
mandatory_delivery_date