Ain shams university
Faculty of engineering
Computer engineering and software systems programs

Course: Object Oriented Analysis & Design (CSE221)

Library System Project

Submitted by: Shehab Ahmed Hassan Kotb Id:16p6014

Hazem hamada Abdellaif Id:16p3100

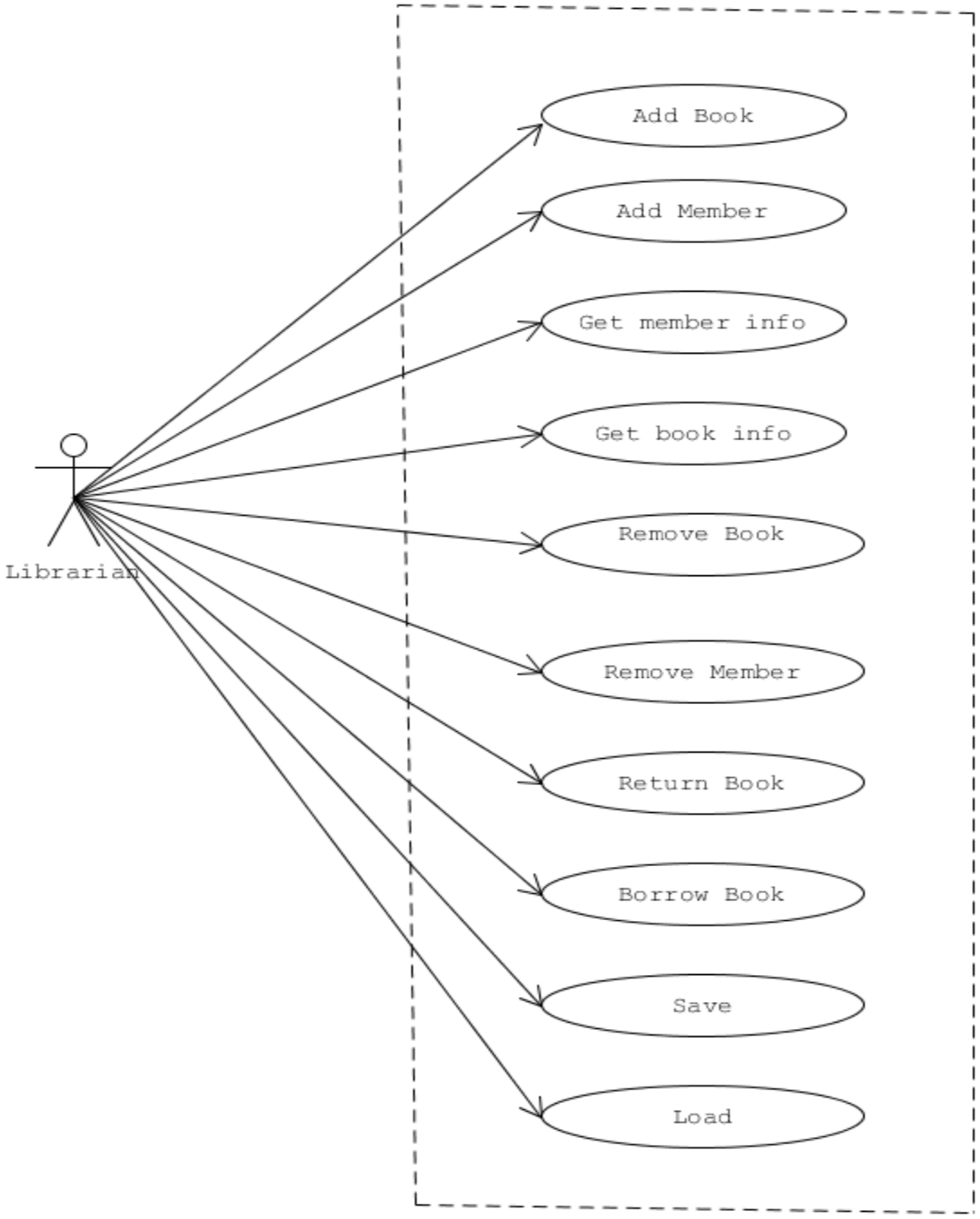
Submitted to: Dr. Islam el Elmadah
Eng. Sara

Table of Contents

1.0 Problem Statement

- A library management system is needed for a public library. The system will be used by Librarians to document the states of books in the library.
- it should provide a way to add or remove books, keep track of how many copies are available of said book.
- it should also allow the librarian to add member to the library or remove them, and keep track of books each member is currently renting.
- the system should permit the user to search the catalog, borrow books if they don't exceed borrowing limit, and return books.
- the system should be intuitive and require little to no training to be able to do basic functions.

2.0 Use Case diagram



3.0 Use Case Descriptions

3.1 Add Book

Use Case Name	Add book.
Goal in context	Add a new book to the system.
Pre-Conditions	The book is in the library.
Successful end conditions	The book successfully added to the system.
Failure End Conditions	The book failed to be added to the system.
Primary Actors	Librarian.
Trigger	The book is delivered to the library.
Main Flow	1. The librarian select catalogue. 2. The librarian selects add book. 3. The Librarian fill all the data required for the new book. 4. The System check the data. 5. The system creates a new object of book with the given data.
Extensions	4.1 no name for the book was entered. 4.2 invalid number of copies was entered.

Analysis classes:

Boundary class: CatalougeJframe, AddBookJframe

Controller class: addbkctrl.

Entity classes: Book(name, author, date, copies, renters).

3.2 Add Member

Use Case Name	Add member.
Goal in context	Add a new member to the system.
Pre-Conditions	The member is in the library.
Successful end conditions	The member is successfully added to the system.
Failure End Conditions	The member failed to be added to the system.
Primary Actors	Librarian.
Trigger	The member requests to be added to the system.
Main Flow	1. The librarian select members. 2. The librarian selects add member. 3. The Librarian fill all the data required for the new member. 4. The System check the data. 5. The system created a new object of member with the given data.
Extensions	4.1 no name was entered for member. 4.2 member already in the system. 4.3 invalid data was entered.

Analysis classes:

Boundary class: MembersJframe, AddMemberJframe.

Controller class: addmbrctrl.

Entity classes: Member(name , address, phone, books rented).

3.3 Get Member info

Use Case Name	Get member information.
Goal in context	Get the chosen member information.
Pre-Conditions	Select the desired member.
Successful end conditions	The system display the chosen member information.
Failure End Conditions	The system doesn't display the chosen member information.
Primary Actors	Librarian.
Trigger	The librarian ask the system for the member information.
Main Flow	1. The librarian selects members. 2. The librarian selects the desired member from the members list. 3. The system will display the member information in the member info panel along with his borrowed books in book rented panel.
Extensions	2.1 The librarian didn't find the member in the list.

Analysis classes:

Boundary class: MembersJframe,.

Controller class: Minfoctrl.

Entity classes: Member(name , address, phone, books rented).

3.4 Get Book Info

Use Case Name	Get book information.
Goal in context	Get the chosen book information.
Pre-Conditions	Select the desired book.
Successful end conditions	The system display the chosen book information.
Failure End Conditions	The system doesn't display the chosen book information.
Primary Actors	Librarian.
Trigger	The librarian ask the system for the book information.
Main Flow	1. The librarian selects catalogue. 2. The librarian selects the desired book from the book list. 3. The system will display the book information in the book info panel.
Extensions	2.1 The librarian didn't find the book in the list.

Analysis classes:

Boundary class: CatalougeJframe.

Controller class: Binfoctrl.

Entity classes: Book(name, author, date, copies, renters).

3.5 Remove Member

Use Case Name	Remove member.
Goal in context	Remove the member from the system.
Pre-Conditions	1. The member is on the system. 2. The member has no borrowed books.
Successful end conditions	The librarian is successfully removed the member from the system.
Failure End Conditions	The librarian is failed to be removed the member from the system.
Primary Actors	Librarian.
Trigger	The Member requests to be removed from the system.
Main Flow	1. The librarian select members. 2. The librarian choose the member. 3. The librarian selects remove member. 4. The system removes the member from the system.
Extensions	2.1 The librarian didn't find the member in the list. 4.1 the member still has book rented.

Analysis classes:

Boundary class: MembersJframe.

Controller class: removembrctrl.

Entity classes: Member(name , address, phone, books rented).

3.6 Remove Book

Use Case Name	Remove book.
Goal in context	Remove the book from the system.
Pre-Conditions	The book is on the system.
Successful end conditions	The librarian successfully removed the book from the system.
Failure End Conditions	The librarian failed to remove the book from the system.
Primary Actors	Librarian.
Trigger	The librarian requests from the system to remove the book.
Main Flow	1. The librarian select catalogue. 2. The librarian choose the book. 3. The librarian selects remove book. 4. The system removes the book from the book from the system.
Extensions	2.1 The librarian didn't find the book in the list. 4.1 The book is rented by a member.

Analysis classes:

Boundary class: CatalougeJframe.

Controller class: removebkctrl.

Entity classes: Book(name, author, date, copies, renters).

3.7 Return Book

Use Case Name	Return Book.
Goal in context	The member returns his borrowed book.
Pre-Conditions	The member and the book should be in the system, the member should has borrowed the book before.
Successful end conditions	The member successfully returns the borrowed book.
Failure End Conditions	The member fails to return the borrowed book.
Primary Actors	Librarian.
Trigger	The member brings the borrowed book to the Librarian.
Main Flow	1. The member bring the borrowed book to the Librarian. 2. The Librarian look for the state of the book on the system to check that the book is on the system and is it borrowed or not. 3. Liberian click on members button. 4. librarian searches for member in the list. 5. select member. 6. search for borrowed book in rented books list of member. 7. click return book. 8. The system removes the book from the list of the borrowed books of the member. 9. The system updates the number free copies of the book and the number of the member's borrowed books.
Extensions	1.1 The member returns the book with some defects or some pages are cut. 4.1 member isn't found in list. 6.1 book isn't presented in the borrowed list.

Analysis classes:

Boundary class: MembersJframe.

Controller class: rtrnbkctrl.

Entity classes: Member(name , address, phone, books rented), Book(name, author, date, copies, renters).

3.8 Borrow Book

Use Case Name	Borrow book.
Goal in context	The member borrows the desired book.
Pre-Conditions	The member and the book both exist in the system.
Successful end conditions	The member successfully borrows the desired book.
Failure End Conditions	The member fails to borrow the desired book.
Primary Actors	Librarian.
Trigger	The member asks the librarian to borrow the desired book.
Main Flow	1. The Librarian selects members. 2. The Librarian selects the member from the members list. 3. The system will open another frame for the Librarian to choose the book to borrow. 4. The Librarian selects the desired book from the list. 5. The system checks if the member exceeds the limit of the borrowed books(5 books). 6. If the member doesn't exceeds his limit the system add the book to the member's borrowed books. 7. Then updates the number of the free copies of the book and the number of member's borrowed books.
Extensions	1.1 librarian doesn't find member. 4.1 the desired book isn't available. 5.1 The client exceeds the limit of the borrowed books.

Analysis classes:

Boundary class: MemberJframe, BorrowBookJframe.

Controller class: brwbkctrl.

Entity classes: Member(name , address, phone, books rented), Book(name, author, date, copies, renters).

3.9 Save

Use Case Name	Save.
Goal in context	Save the current state of the system.
Pre-Conditions	Changes are made in the system.
Successful end conditions	The system saves the new member's information and book's information.
Failure End Conditions	The system fails to save the new member's information and book's information.
Primary Actors	Librarian.
Trigger	The librarian press the save button.
Main Flow	1. The librarian selects save button. 2. The system will save the new changes in the new member's information and book's information in specific files. 3. If the files are not found the system create new files and save in it
Extensions	2.1 the system can't create the files needed to save in.

Analysis classes:

Boundary class: WelcomeJframe.

Controller class: savectrl.

Entity classes: Member(name , address, phone, books rented), Book(name, author, date, copies, renters).

4.0 Analysis Classes

After analyzing classes needed for each use case we will list all of them here.

Boundary Classes:

- WelcomeJframe
- CatalogueJframe
- MembersJframe
- AddBookJframe
- AddMemberJframe
- BorrowBookJframe

Control Classes:

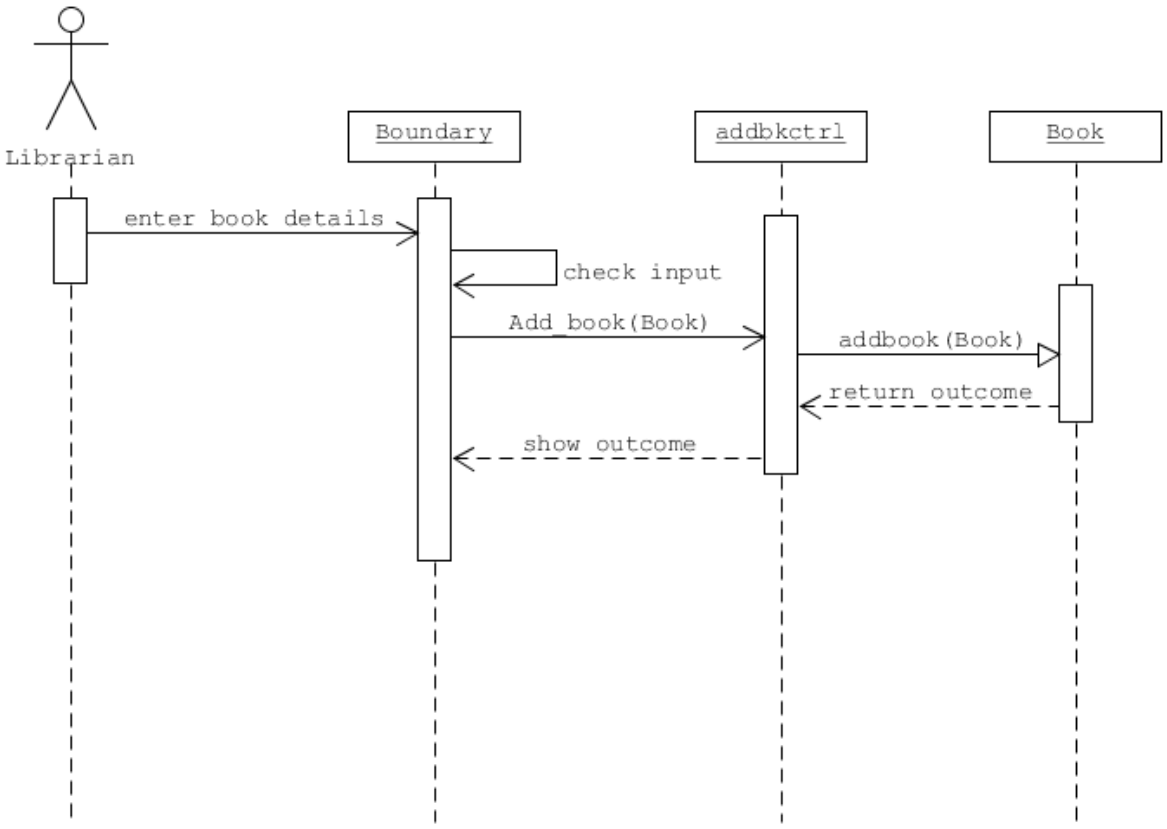
- AddBookControl
- AddMemberControl
- GetMemberInfoControl {minfoctrl in code}
- GetBookInfoControl {binfoctrl in code}
- RemoveBookControl
- RemoveMemberControl
- RetrunBookControl
- BorrowBookControl
- SaveControl

Entity Classes:

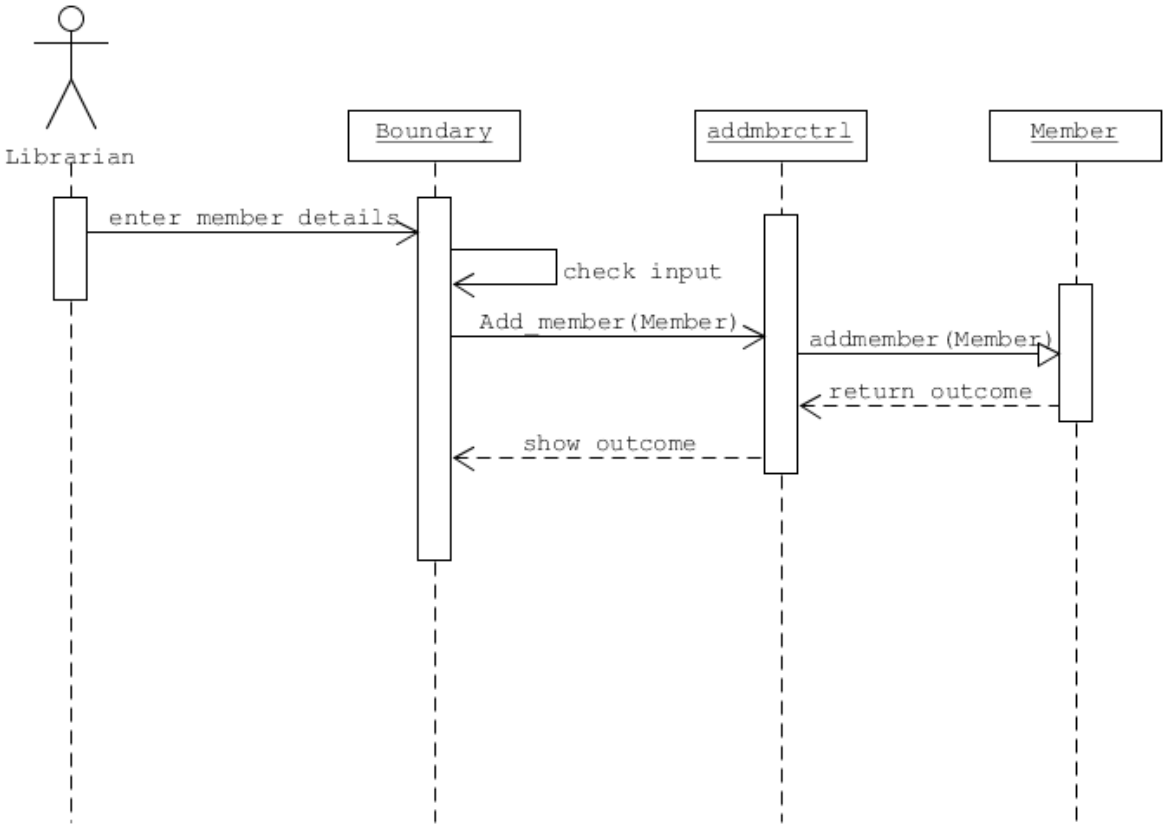
- Member
- Book

5.0 Sequence Diagrams

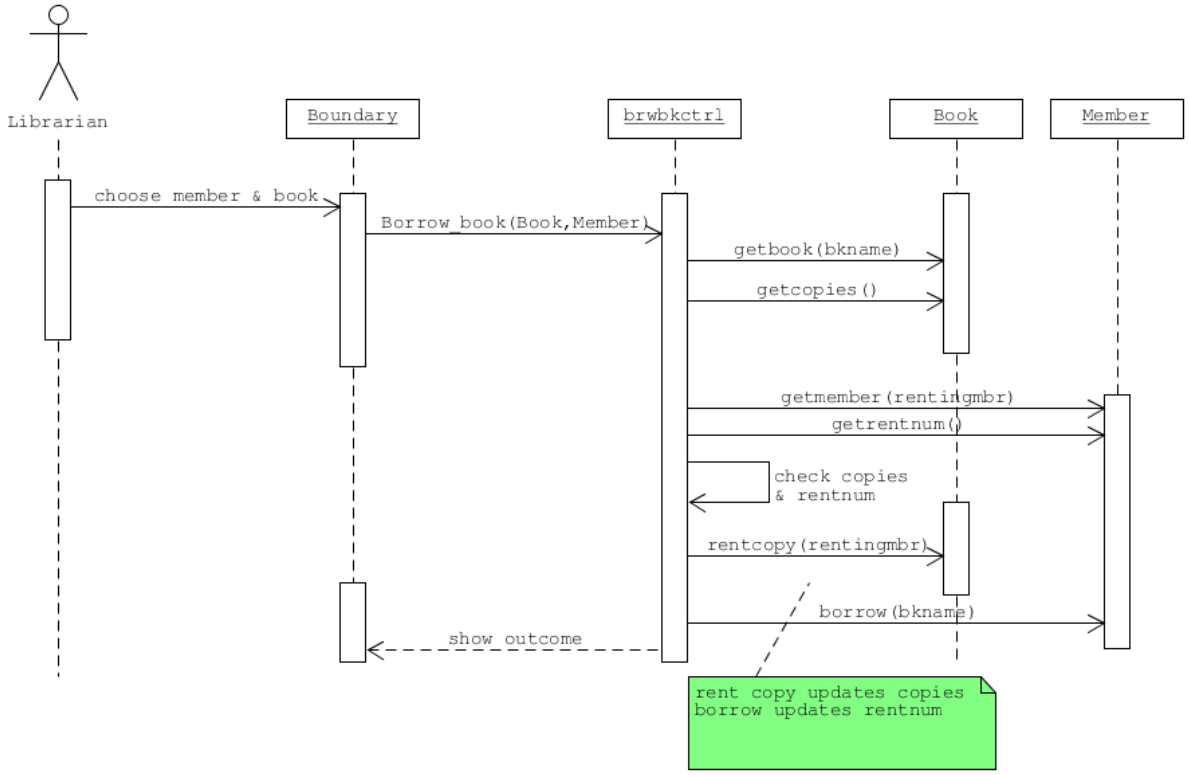
5.1 AddBook



5.2 AddMember



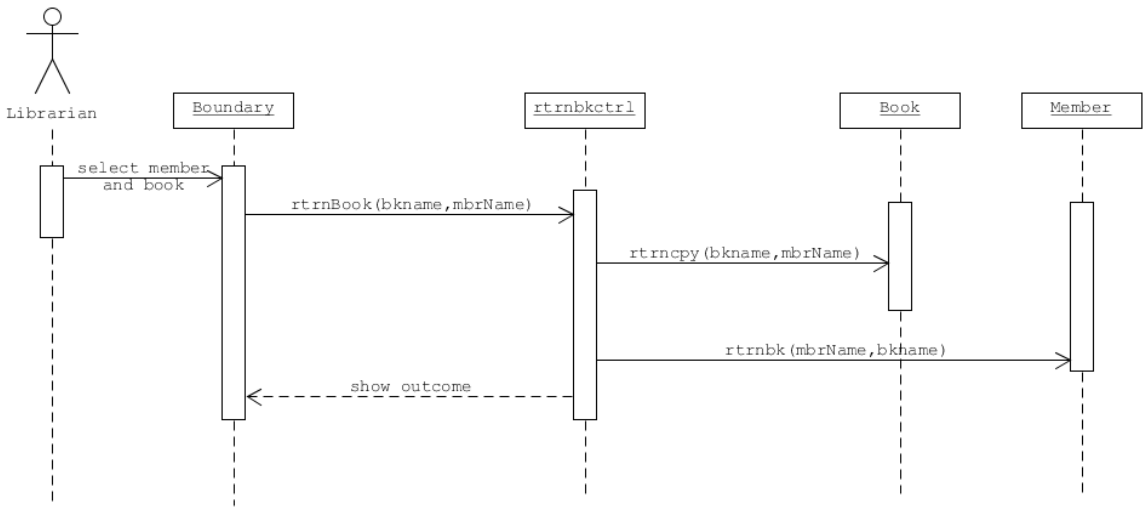
5.3 BorrowBook



The reason why the controller calls one method for each Book, and member is because in each class there is variables that need to be updated.

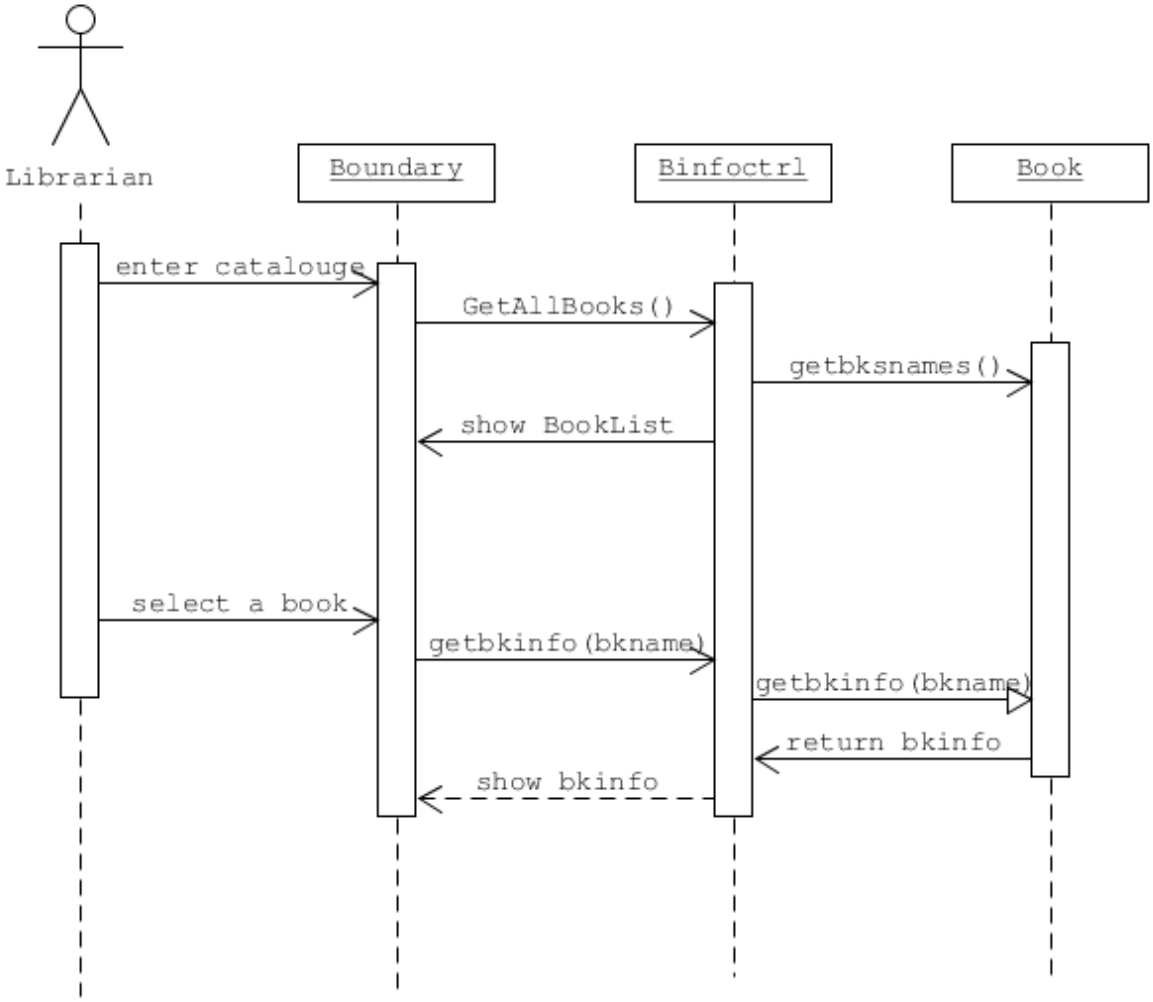
- In Book the number of copy and list of renters of that book are updated by **rentcopy.**
- In Member the rent number and the list of rented books for the member need to be updated by **borrow.**

5.4 ReturnBook

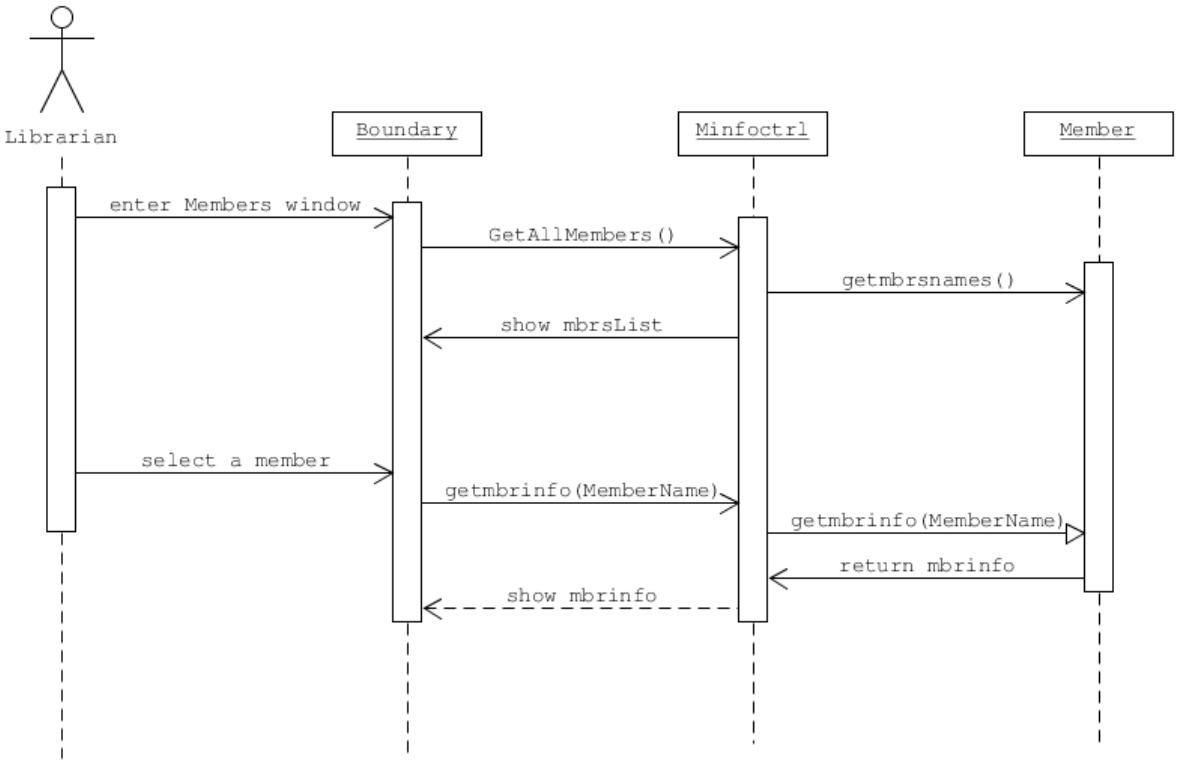


The same happens here but in reverse like BorrowBook usecase.

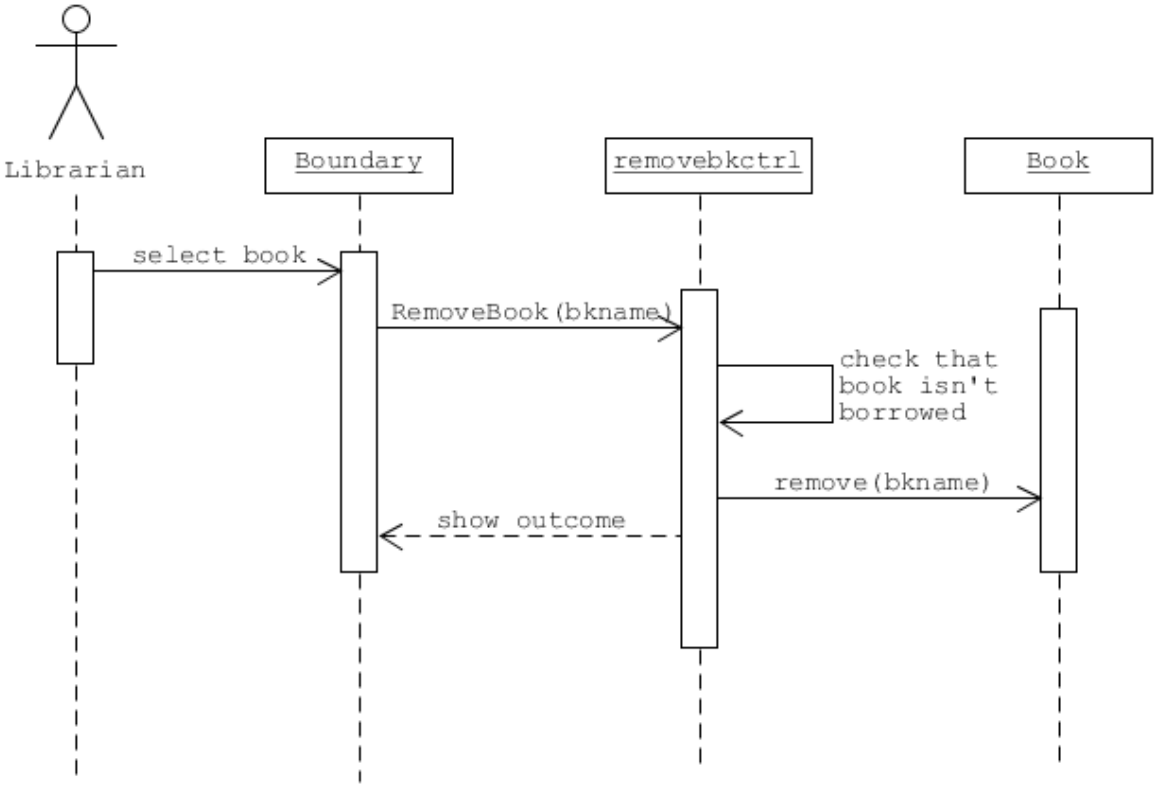
5.5 GetBookInfo



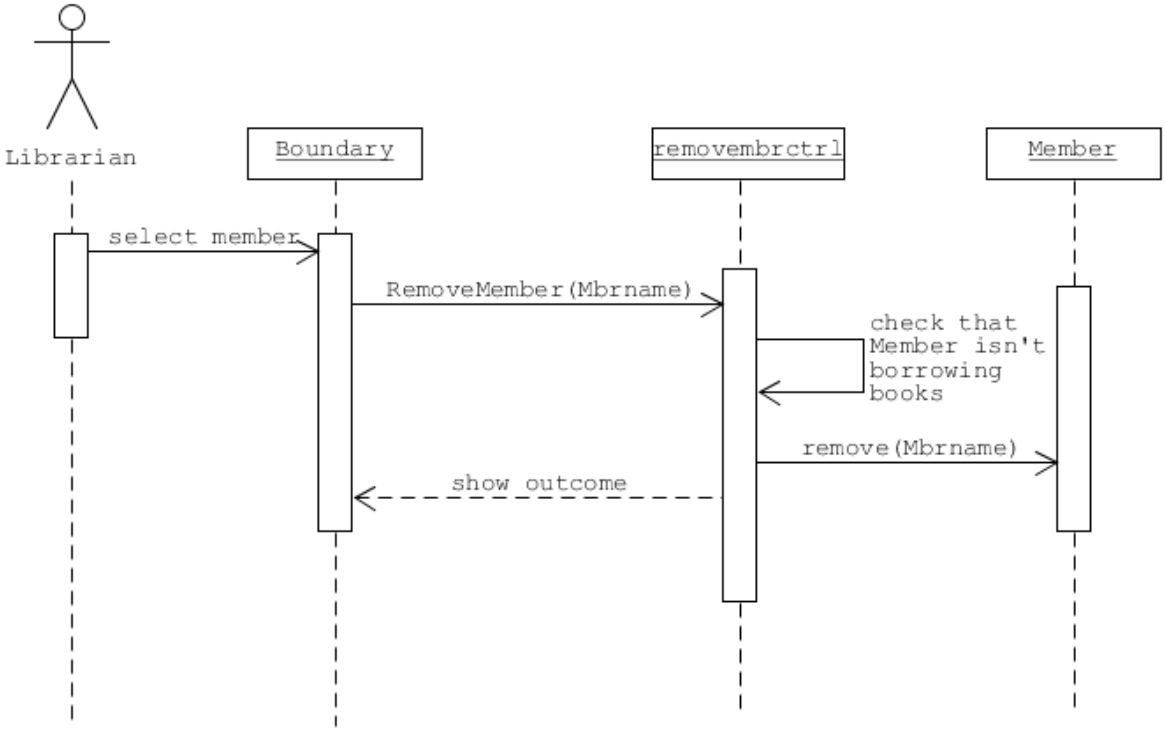
5.6 GetMemberInfo



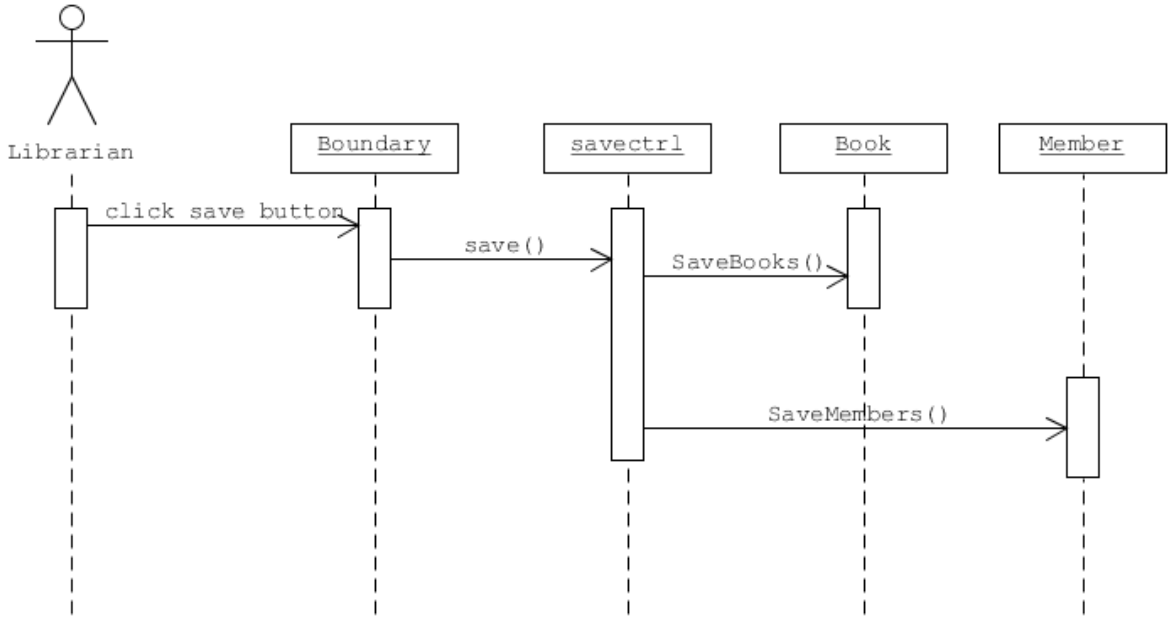
5.7 RemoveBook



5.8 RemoveMember

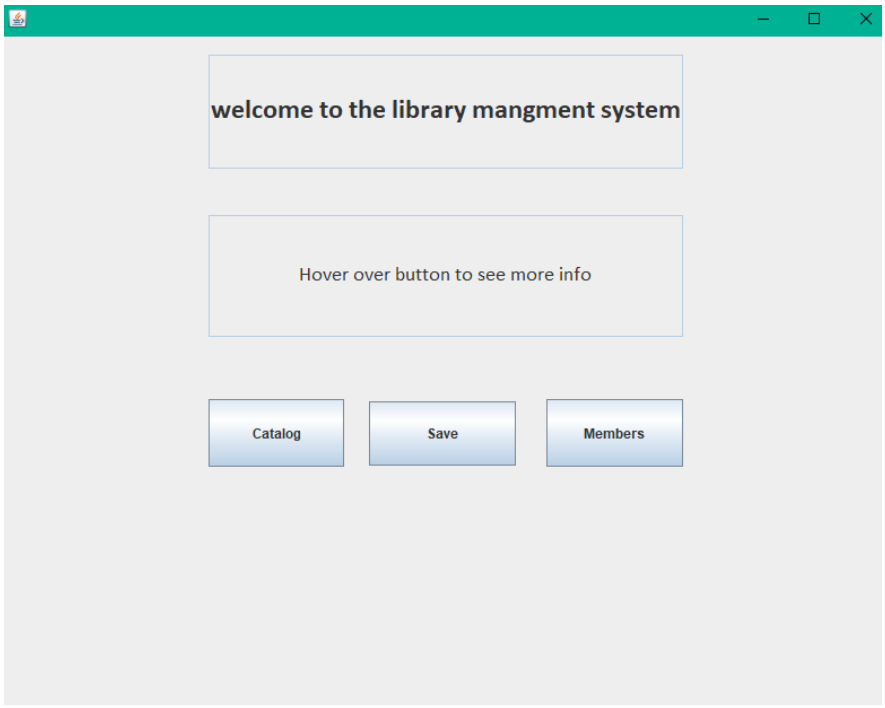


5.9 Save

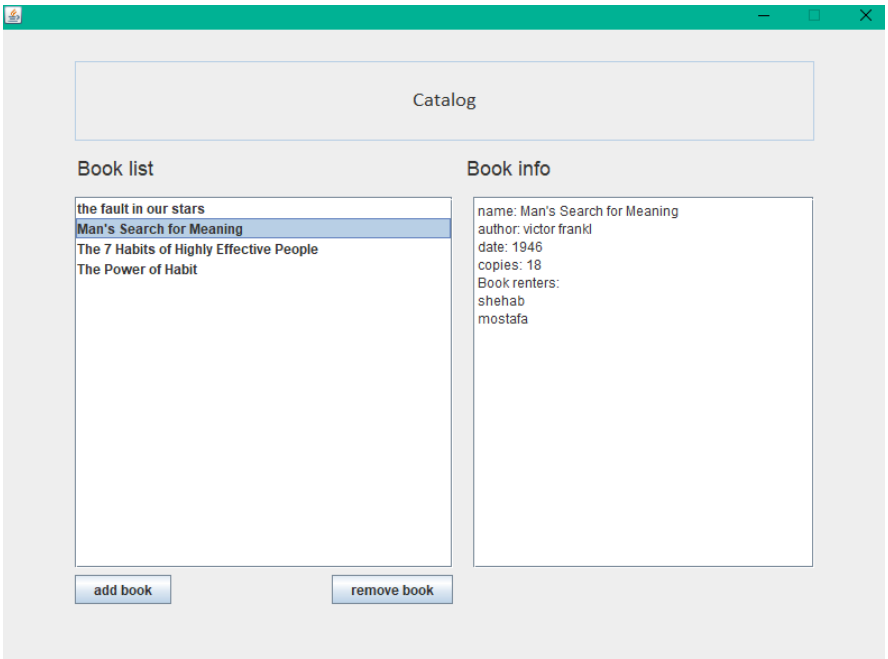

Librarian
Boundary
savectrl
Book
Member
click save button
save()
SaveBooks()
SaveMembers()

6.0 Screen shots of the program

Welcome Screen:



Catalogue Screen:



Members Screen: