HACETTEPE UNIVERSITY

Computer Engineering
BBM487 Software Engineering Laboratory

Library Book Loan System
System-Wide Requirement Specification

GÜLŞAH ERDOĞAN	**21327953**
GÜLENDAM BUKET GÜNDÜZ	**21328026**
İREM ÖZEN	**21328337**

Book Loan System
System-Wide Requirements Specification

1. Introduction

In this project we aim to create a Book Loan System. We plan to implement a desktop application where users can take and leave books or check books and members.

In this document we describe the features that users can use, the constraints, the requirements of our system the relationships with other systems, and the interfaces we have prepared.

2. System-Wide Functional Requirements

Our project is to design a book loan system. Our goal is to ensure that operations are fast and accurate. We also aim to make it easier to use our application with a user-friendly interface.

Our system is a desktop application that we have designed for the needs of users using C # language. There are two different user types in the system. One is a member and the other is a library. Users log in to the system with their usernames and passwords from a common interface, but when they log in they are directed to different interfaces depending on the type of user.

Member can log in, log out, sign up, search a book, reservation a book by adding to waiting list, take book by self, drop back by self, see books that he/she had, pay his/her fine, communicate with librarian. Member's use cases are as following:

1. **Login:** It is providing to enter to the system with username and password.
2. **Logout:** It is providing to exit from the system with username and password.
3. **Sign up:** It allows you to create a member account.
4. **Make Reservation:** It is providing to add member to the waiting list of the selected book.
5. **Cancel Reservation:** It is providing to cancel a reservation for book
6. **Search Book:** It helps to search books with its code, name or author's name.
7. **Self Check Out:** It is providing to take a book from the system byrself.
8. **Self Return:** It is providing to return the book to the system by yourself.
9. **View My Books:** It lists all books that you have.
10. **Pay Fine:** Users pay their fine.
11. **Communication With The Librarian:** Member can communicate with the librarian, If the member encounters any problem.
12. **Waiting List:** It is providing show waiting list for a book.

Librarian can log in, log out, insert, delete and update books and list members, delete a member, list books that borrowed and fine of the members. Librarian's use cases are as following:

1. **Login:** It is providing to enter to the system with username and password.
2. **Logout:** It is providing to exit from the system with username and password.
3. **Manipulate Book:** It is providing to insert, delete and update a book.
4. **Manipulate Member:** It is providing to list members, delete a member, list books borrowed and fine.

There are two more use case done by the system itself:

1. **Notify of Book Being Available:** The system marked book available or not available. If it

is available, members can take the book.

2. **Issue Late Fine:** The system calculates the fine

3. System Qualities

The system we designed is very easy to use and will perform operations quickly and accurately. It must be protected against malware.

3.1 Usability

The interface is targeted to be simple, clear and user-friendly. The system is designed to be practical and easy to use. It consists of simple and basic steps that will not force users. If the user makes mistakes while using the system, a warning is given.

3.2 Reliability

The system protects user data and provides security. Each user is authenticated. This will ensure that books, bookings or other requests are made without errors in our library system

3.3 Performance

Our system will be prepared using ideal algorithms. At this point, users will be able to process transactions as soon as possible. Except that the response time will increase as the density of the database increases.

3.4 Supportability

The system will be implemented in modules, where it will be easier and quicker to incorporate any changes that may be needed later on in the system.

4. System Interfaces

Our system has no system interface.

4.1 User Interfaces

Our Book Loan System's interfaces are in the Appendix B, with explanations.

4.1.1 Look & Feel

The interfaces we use in our system have simple, eye-catching designs. We mostly used shades of gray and rectangular shapes. Thus, we have achieved a stylish and comfortable design.

4.1.2 Layout and Navigation Requirements

A window will open when the program runs and all operations will be done from that window. New windows will not open while any action is being taken. A new smaller window will also be opened in the middle of screen to inform you that an error has occurred or that the process has been completed.

4.1.3 Consistency

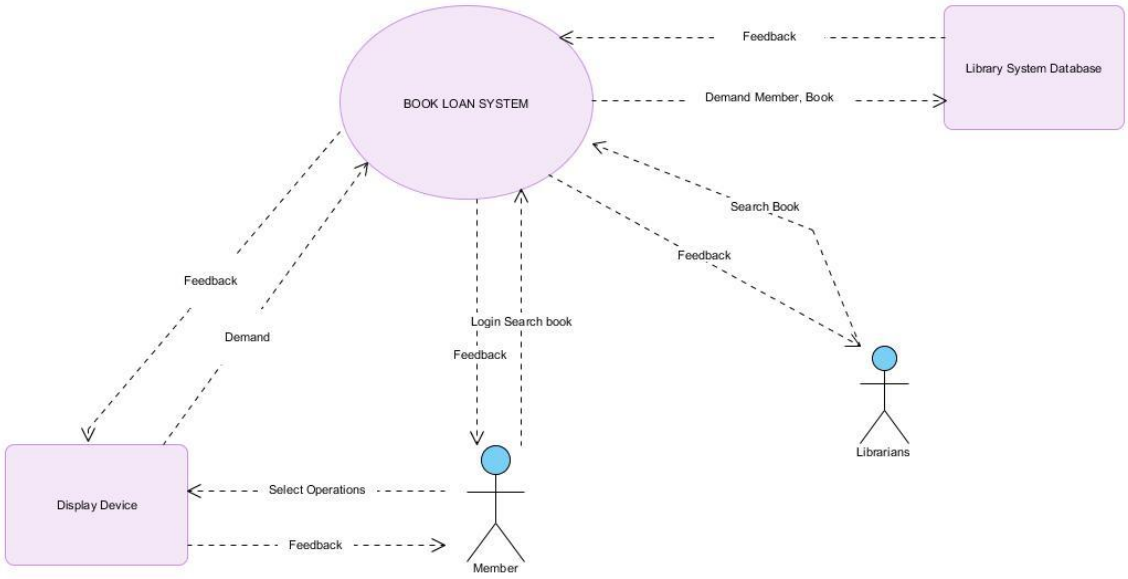
We have created each page in a similar format to ensure consistency. Colors, sizes, button shapes were created in the same way.

4.1.4 User Personalization & Customization Requirements

When the user logs in, the program meets username by typing username welcome. Librarians and member have different interfaces after they log in. The member is directed to an interface where the process of taking books, reserving, etc. is directed, while an interface is directed to the librarian, where members and books can be controlled. Apart from these, the program does not have a feature that users can customize.

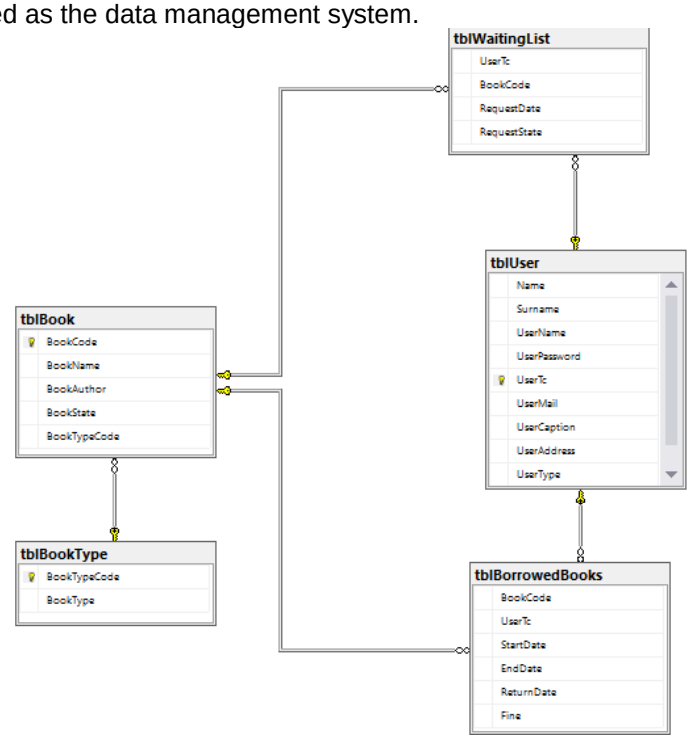
4.2 Interfaces to External Systems or Devices

It is the context diagram of our Book Loan System. Our System has a relationship with display devices and a database system.



4.2.1 Software Interfaces

When we developed the system, we designed the interfaces in Windows Form Application in Microsoft Visual Studio. We will use .Net Framework 4 to do our database operations. MySql will be used as the data management system.



4.2.2 Hardware Interfaces

The hardware contained within the PCs is sufficient for running the program. A mouse and keyboard are required for use the application.

4.2.3 Communications Interfaces

Our Book Loan System has not an communications interface. The member can communicate with the librarian registered in the library system through the link mail address. The librarian's e-mail address is in the system. This is related to member's "Communication With The Librarian" use case.

5. Business Rules

5.1 <Taking a book >

5.1.1 <book limit>

A member cannot take more than five books.

5.2 <leave a book >

5.2.1 <time limit >

A member must return book within 30 days.

5.2.2 <fine >

If the member does not leave the book within the time limit, 0.5 kuruş fine will be imposed for each delayed day.

5.3 <Creating new account >

5.3.1 <Same username >

There will be no librarian with the same username. This will be valid for members.

6. System Constraints

The system is a desktop application. An infrastructure is needed to provide the necessary hardware requirements such as sufficient storage space and processor power in the computer to be used. The system is designed to run only in Windows environment, it is not expected to work in any other operating system. It is expected that the computer to be used will have infrastructure components that can run C # programs.

7. System Compliance

7.1 Licensing Requirements

We do not need a license for the programming language and database that we use to improve our system. You can find it on the internet for free.

7.2 Legal, Copyright, and Other Notices

This document is intended for clients who are interested in preparing the Book Loan System and project team. Reproduction of the Software Design Report by anyone other than the responsible person is prohibited. We are prohibited from changing the system's codes and sharing them with third parties.

7.3 Applicable Standards

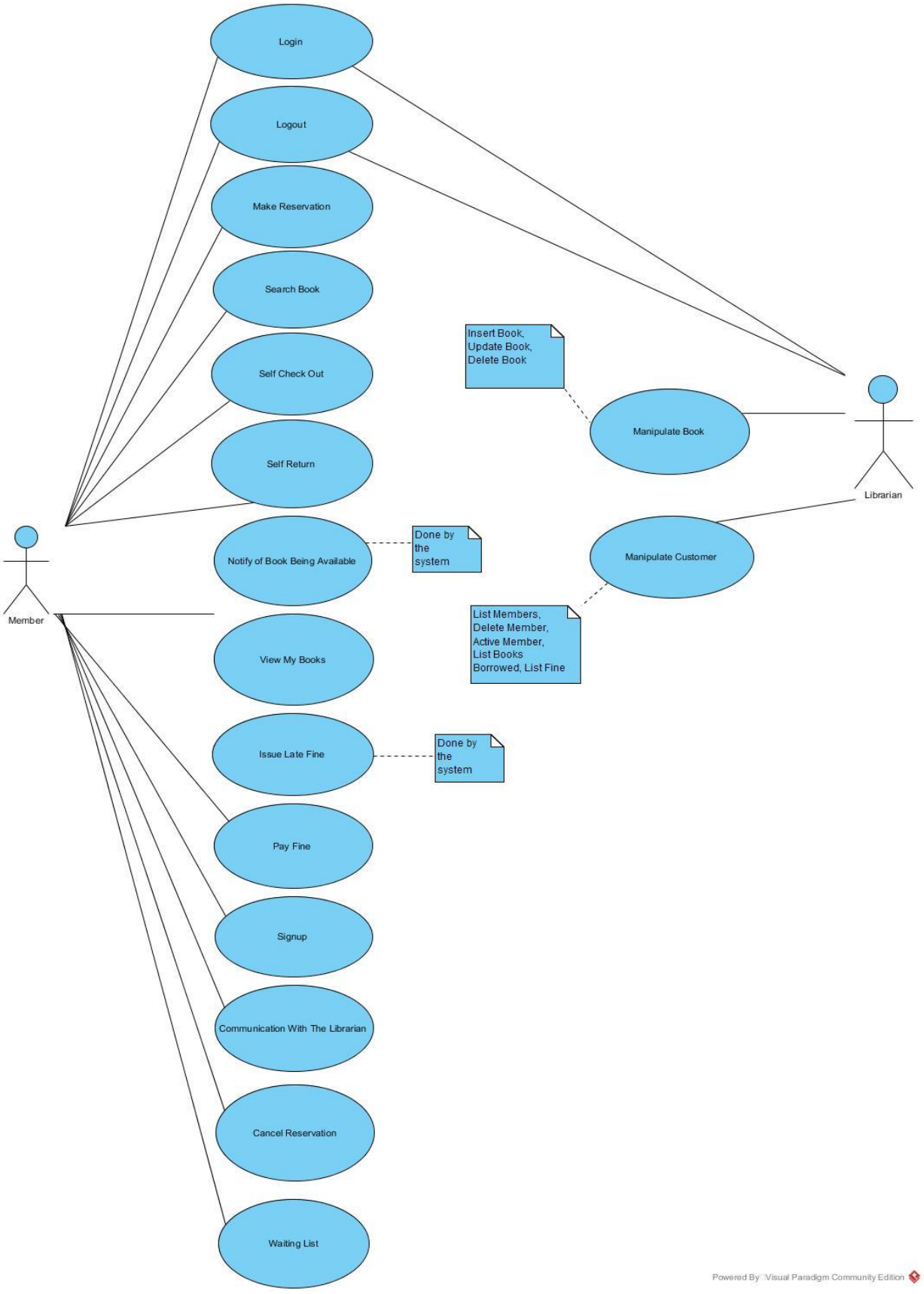
Our system follows C# coding standards and MySQL coding standards.

8. System Documentation

A guide will be provided by the project team on how to use our project. This guide will include information about how the library will manage books and members, and how users can sign up, take and leave books.

Appendix A

Our Book Loan System’s use case diagram is as following.


Login
Logout
Make Reservation
Search Book
Self Check Out
Self Return
Notify of Book Being Available
View My Books
Issue Late Fine
Pay Fine
Signup
Communication With The Librarian
Cancel Reservation
Waiting List
Member
Librarian
Manipulate Book
Manipulate Customer
Insert Book, Update Book, Delete Book
Done by the system
List Members, Delete Member, Active Member, List Books Borrowed, List Fine
Done by the system
Powered By Visual Paradigm Community Edition

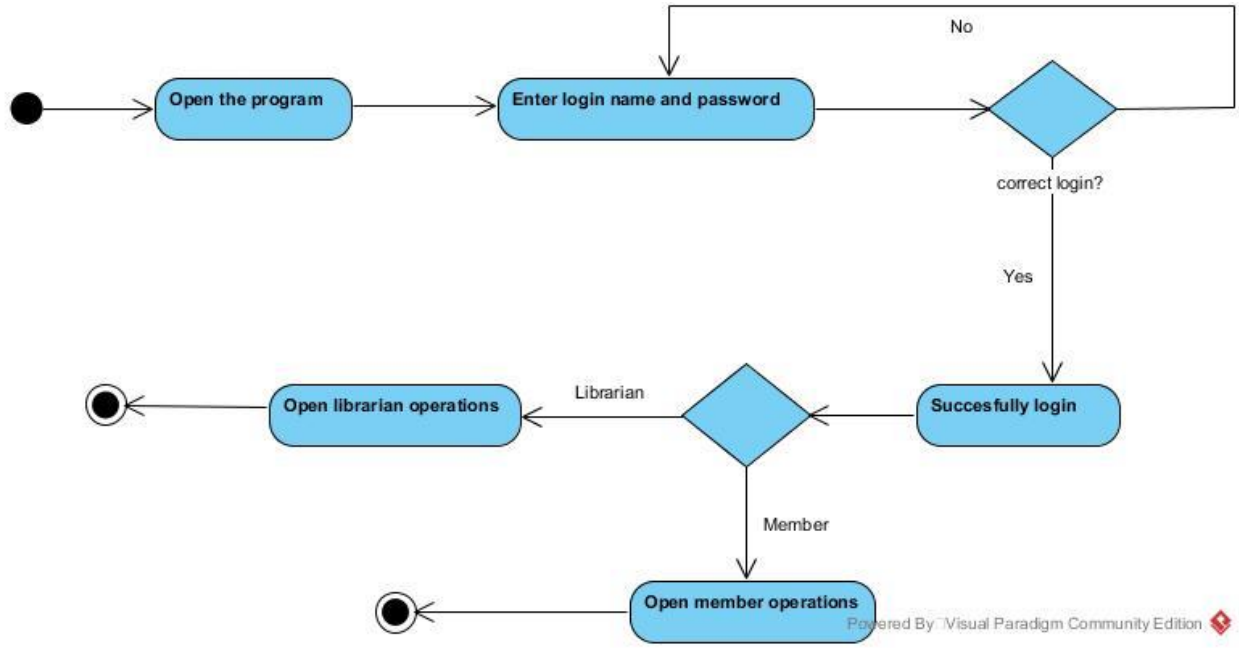
Use case: Login
Scenario:

1. Open the program.
2. Enter username and password
3. The system checks the entered user name and password from the stored information.
4. The system delivers that username and password are correct.
5. The system checks who is login.
6. If the librarian who is logging in, system opens librarian operations screen. If the member who logging in, system opens member operations.

Alternative Scenario:

1. If users entered wrong username or password, system give a warning and directs the user to the login screen.

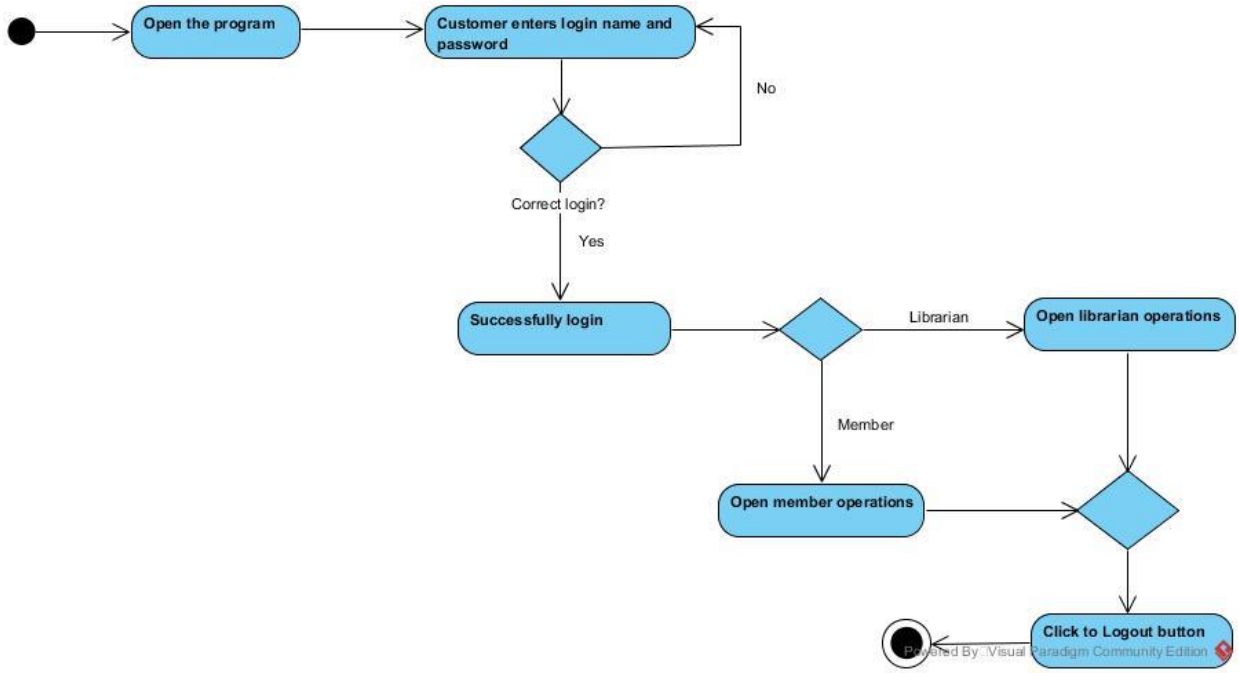
Activity Diagram: Login



Use case: Logout
Scenario:

1. Open the program.
2. Enter username and password
3. The system checks the entered user name and password from the stored information.
4. The system delivers that username and password are correct.
5. The system checks who is login.
6. If the librarian who is logging in, system opens librarian operations screen. If the member who logging in, system opens member operations.
7. User clicks to Logout button

Activity Diagram: Logout



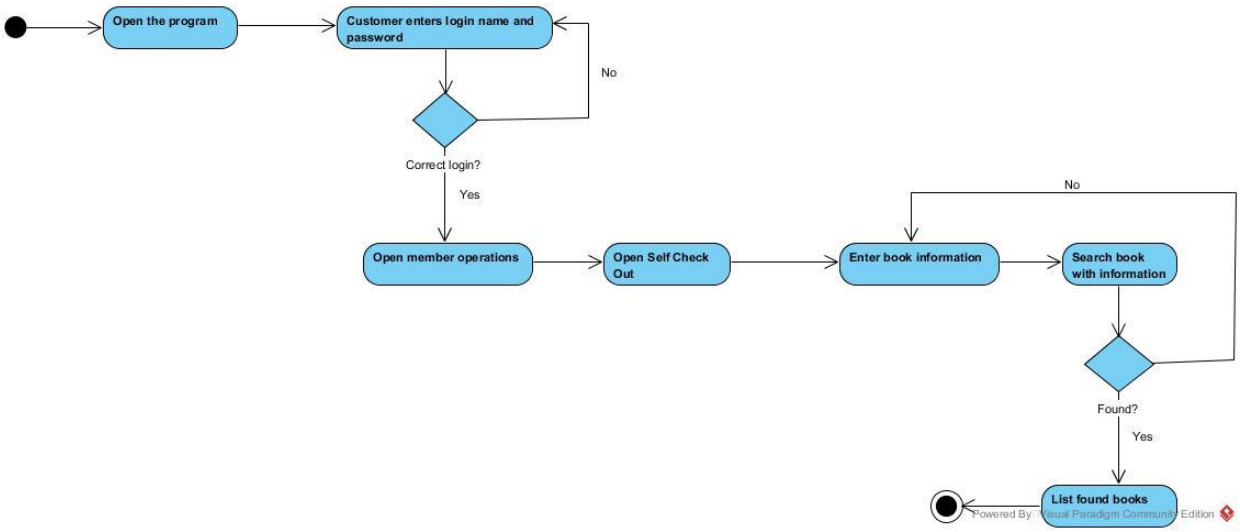
Use Case: Search Book
Scenario:

1. Member opens the program.
2. Member enters username and password.
3. The system checks username and password's correctness.
4. The system delivers that username and password are correct.
5. The system opens member operations screen.
6. Member opens reservation screen.
7. Member enters book information.
8. The system search book with information.
9. System finds book and list it.

Alternative Scenario:

1. The system can't find the book and directs member to the enter book information screen.

Activity Diagram: Search Book



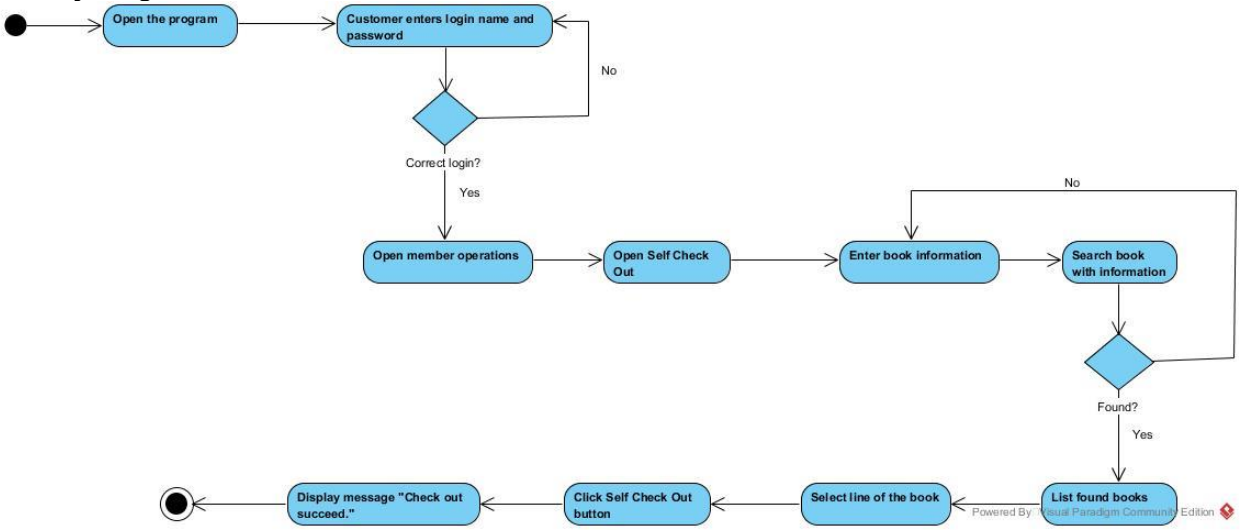
Use Case: Self Check Out
Scenario:

1. Member opens the program.
2. Member enters username and password.
3. The system checks username and password's correctness.
4. The system delivers that username and password are correct.
5. The system opens member operations screen.
6. Member opens Self Check Out screen.
7. Member enters book information.
8. The system search book with information.
9. System finds book and list it.
10. Member selects the line of the book.
11. Member clicks Self Check Out button.
12. System displays message "Check out succeed."

Alternative Scenario:

1. If found book is not available, member uses Make Reservation use case.

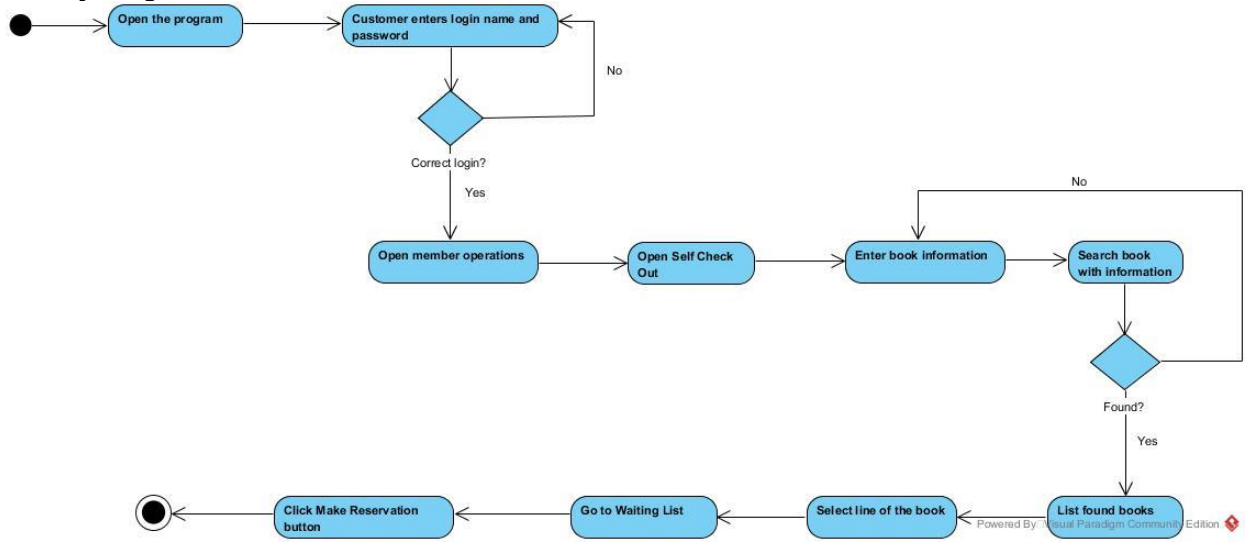
Activity Diagram: Self Check Out



Use Case: Make Reservation
Scenario:

1. Member opens the program.
2. Member enters username and password.
3. The system checks username and password's correctness.
4. The system delivers that username and password are correct.
5. The system opens member operations screen.
6. Member opens Self Check Out screen.
7. Member enters book information.
8. The system search book with information.
9. System finds book and list it.
10. Member selects the line of the book.
11. Member clicks Go to Waiting list button.
12. Member clicks Make Reservation button.

Activity Diagram: Make Reservation



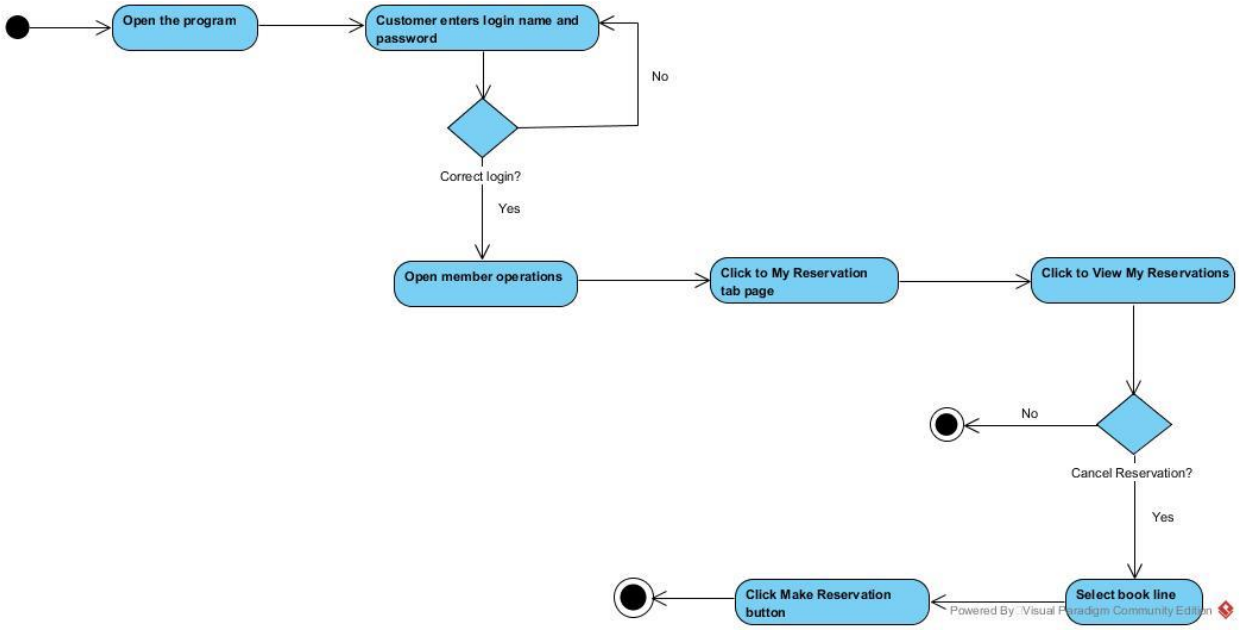
Use Case: Cancel Reservation
Scenario:

1. Member opens the program.
2. Member enters username and password.
3. The system checks username and password's correctness.
4. The system delivers that username and password are correct.
5. The system opens member operations screen.
6. Member opens reservation screen.
7. Member clicks on My Reservation tab page.
8. Members view their reservations by clicking My Reservation button.
9. Member selects book line from listed reservations.
10. Member clicks on Cancel Reservation button.

Alternative Scenario:

1. Member wants to see only reservations and leaves the screen.

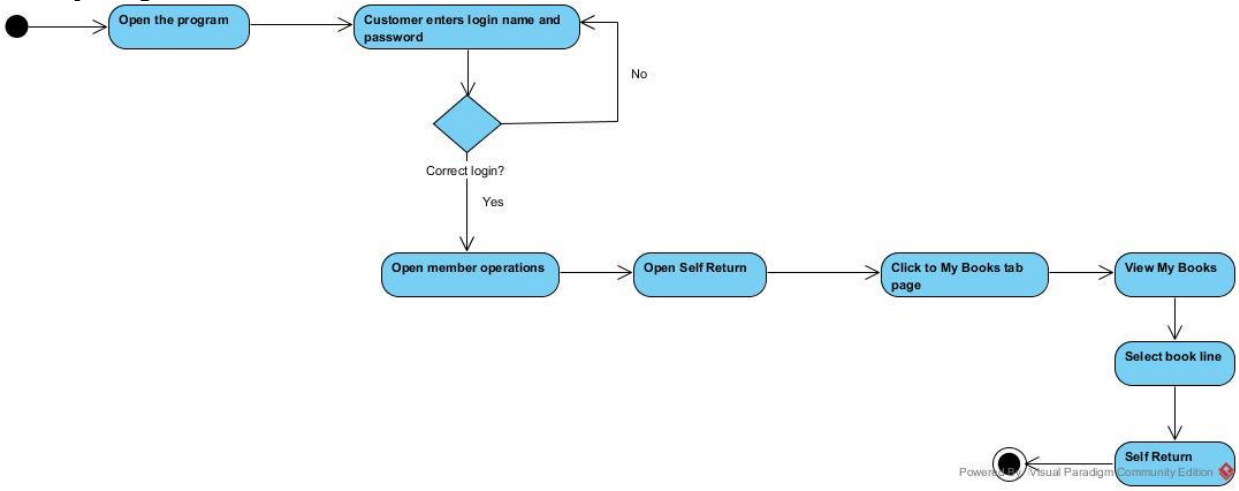
Activity Diagram: Cancel Reservation



Use Case: Self Return
Scenario:

1. Member opens the program.
2. Member enters username and password.
3. The system checks username and password's correctness.
4. The system delivers that username and password are correct.
5. The system opens member operations screen.
6. Member opens Self Return.
7. Member clicks to My Books tab page.
8. Member clicks View My Books button.
9. System list the book that member has.
10. Member selects book line.
11. Member clicks to Self Return button.
12. System displays message "Self Return succeed."

Activity Diagram: Self Return



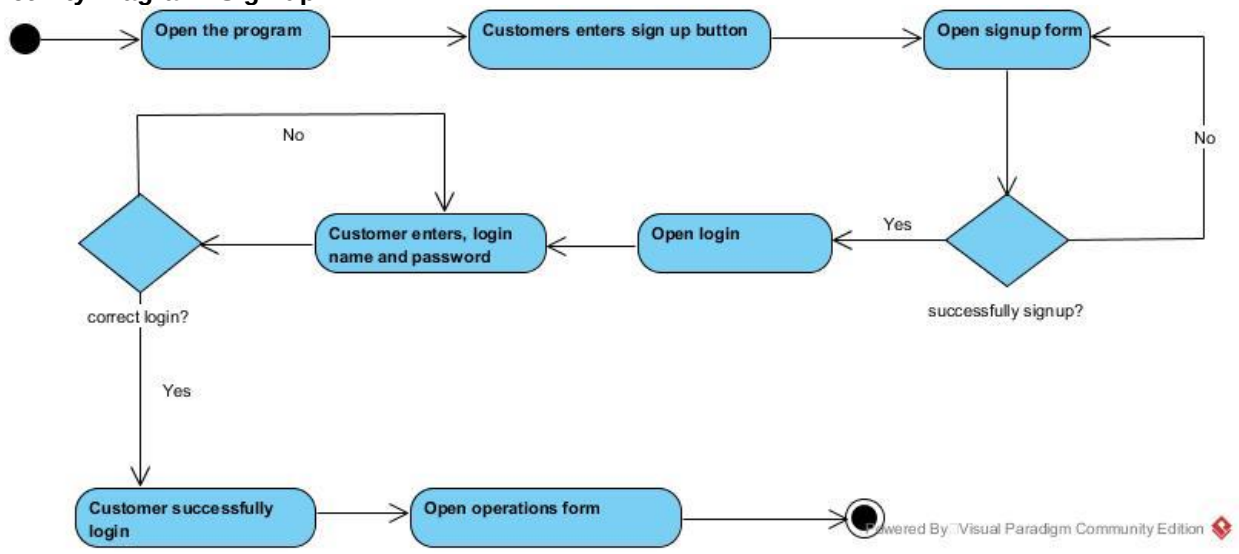
Use Case: Sign up
Scenario:

1. User opens the program.
2. User enters Sign up button.
3. The system opens sign up form.
4. User fill the form and the system directs users to login screen.
5. User enters username and password.
6. User successfully login and the system opens operations form.

Alternative Scenario:

1. If user failed to sign up, the system directs user to the sign up form again.

Activity Diagram: Sign up



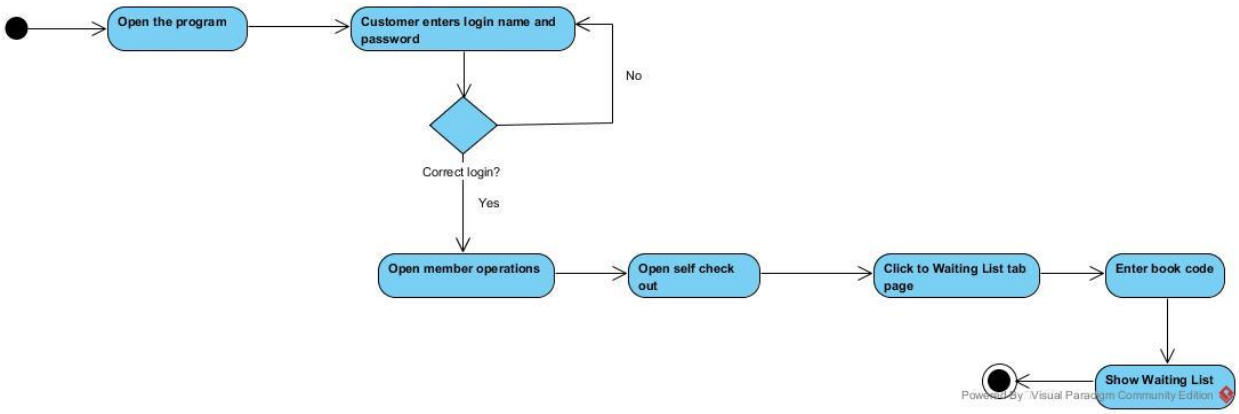
Use Case: Waiting List
Scenario:

1. Member opens the program.
2. Member enters username and password.
3. The system checks username and password's correctness.
4. The system delivers that username and password are correct.
5. The system opens member operations screen.
6. Member opens Self Check Out.
7. Member clicks to Waiting List tab page.
8. Member enters book code.
9. Member clicks Show Waiting List button.
10. The system lists Waiting List for the book.

Alternative Scenario:

1. If there is no one waiting for the book, the system doesn't list anything.

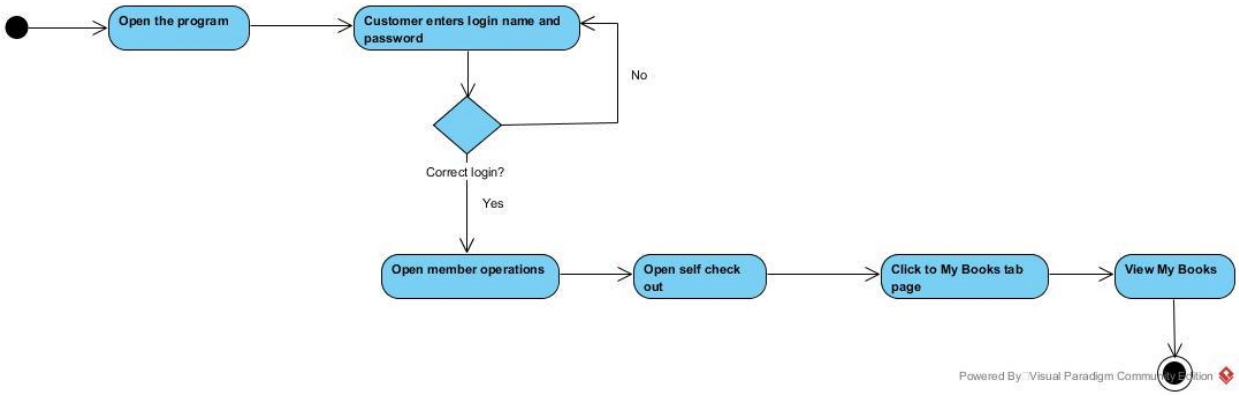
Activity Diagram: Waiting List



Use Case: View My Books
Scenario:

1. Member opens the program.
2. Member enters username and password.
3. The system checks username and password's correctness.
4. The system delivers that username and password are correct.
5. The system opens member operations screen.
6. Member opens Self Check Out.
7. Member clicks to My Books tab page.
8. Member clicks View My Books button.
9. The system lists books that member has.

Activity Diagram: View Books

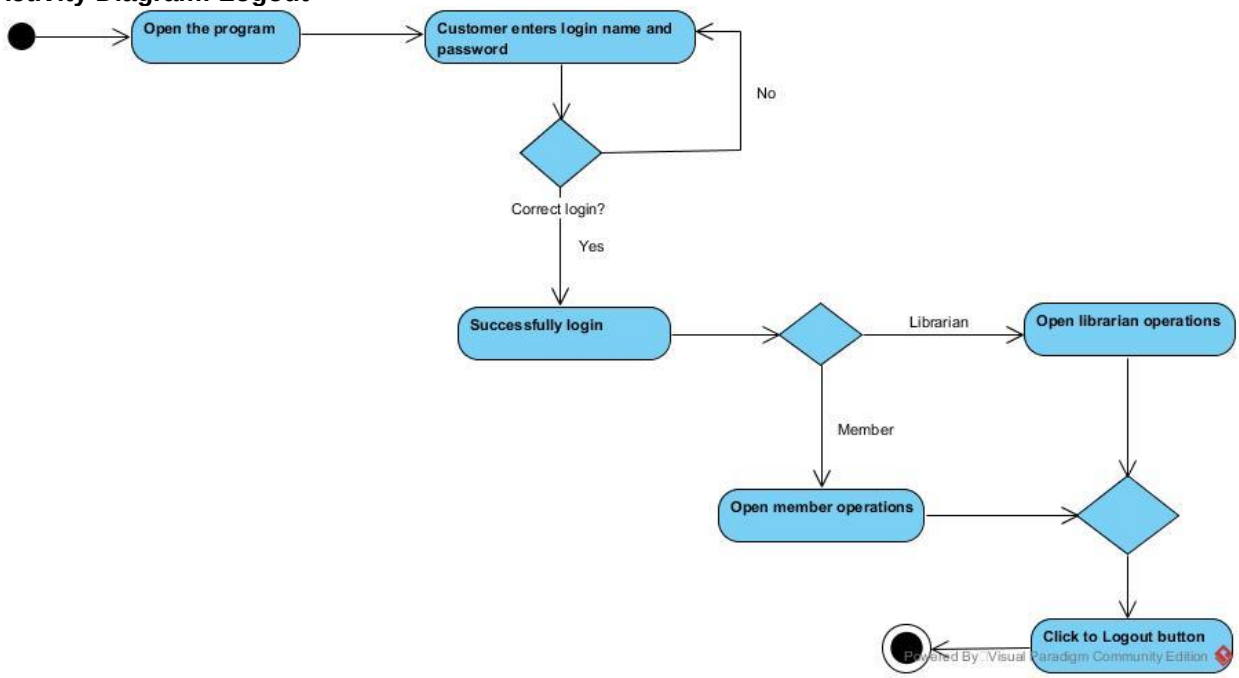


Use Case: Logout
Scenario:

1. Member opens the program.
2. Member enters username and password.
3. The system checks username and password's correctness.
4. The system delivers that username and password are correct.
5. User logins successfully.
6. If librarian logins, the system opens librarian operations.
7. If member logins, the system opens member operations.

8. Users click Logout button.

Activity Diagram: Logout



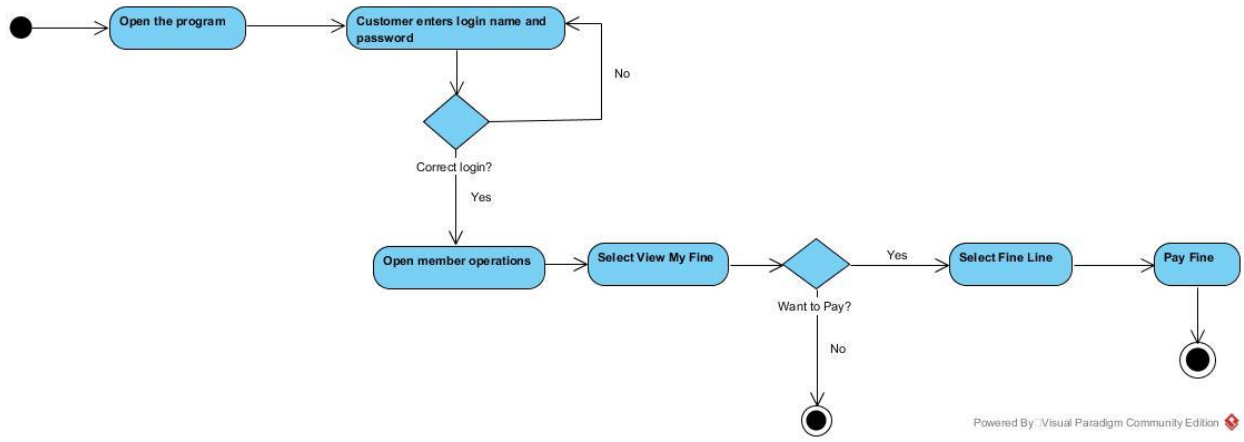
Use Case: Pay Fine
Scenario:

1. Member opens the program.
2. Member enters username and password.
3. The system checks username and password's correctness.
4. The system delivers that username and password are correct.
5. The system opens member operations screen.
6. Member select View My Fine button.
7. Member selects fine line.
8. Member clicks Pay Fine button and fine will be paid.

Alternative Scenario:

1. If member doesn't want to pay fine, process will be done.

Activity Diagram: Pay Fine



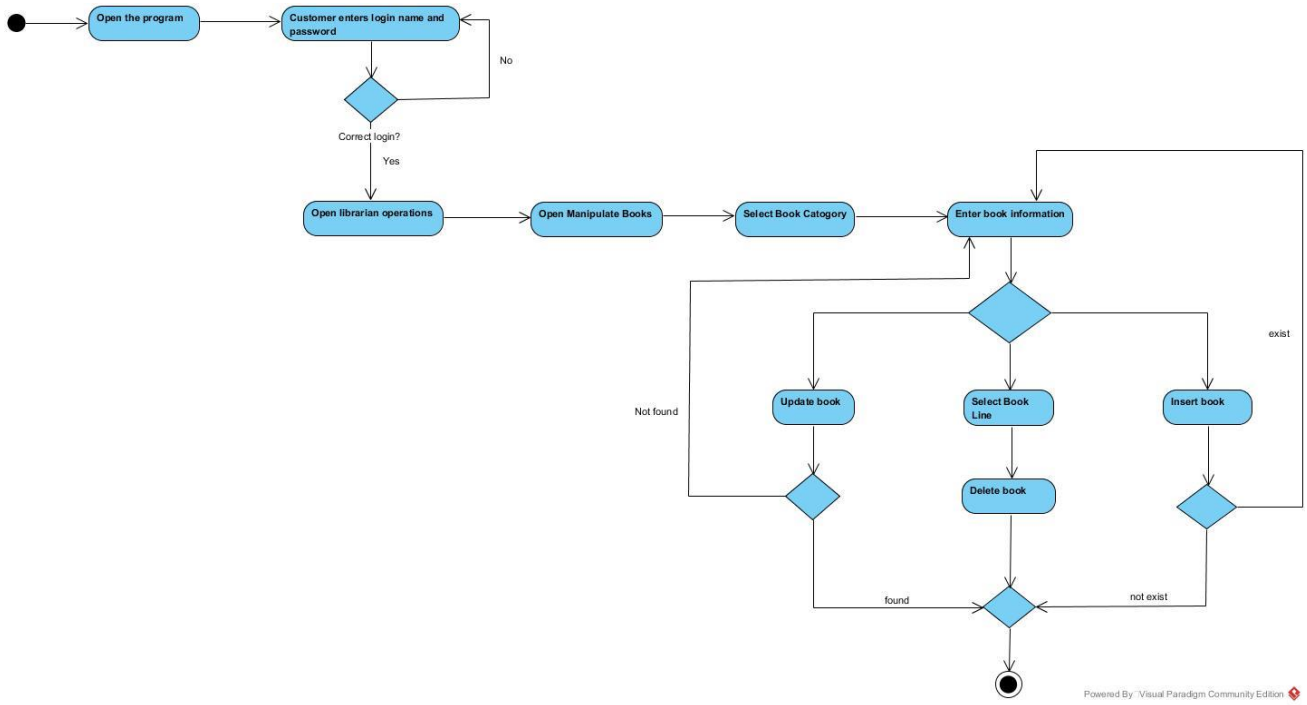
Use Case: Manipulate Books
Scenario:

1. Librarian opens the program.
2. Librarian enters username and password.
3. The system checks username and password's correctness.
4. The system delivers that username and password are correct.
5. The system opens librarian operations screen
6. Librarian opens Manipulate Books.
7. Librarian selects a book category.
8. Librarian enters book information.
9. If librarian wants to update a book, clicks Update Book button.
10. If librarian wants to delete a book, clicks Delete Book button.
11. If librarian wants to insert a book, clicks Insert Book button.

Alternative Scenario:

1. If the book exists, the system directs the librarian to the stage in which book information is entered.
2. If the book doesn't found, the system directs the librarian to the stage in which book information is entered.

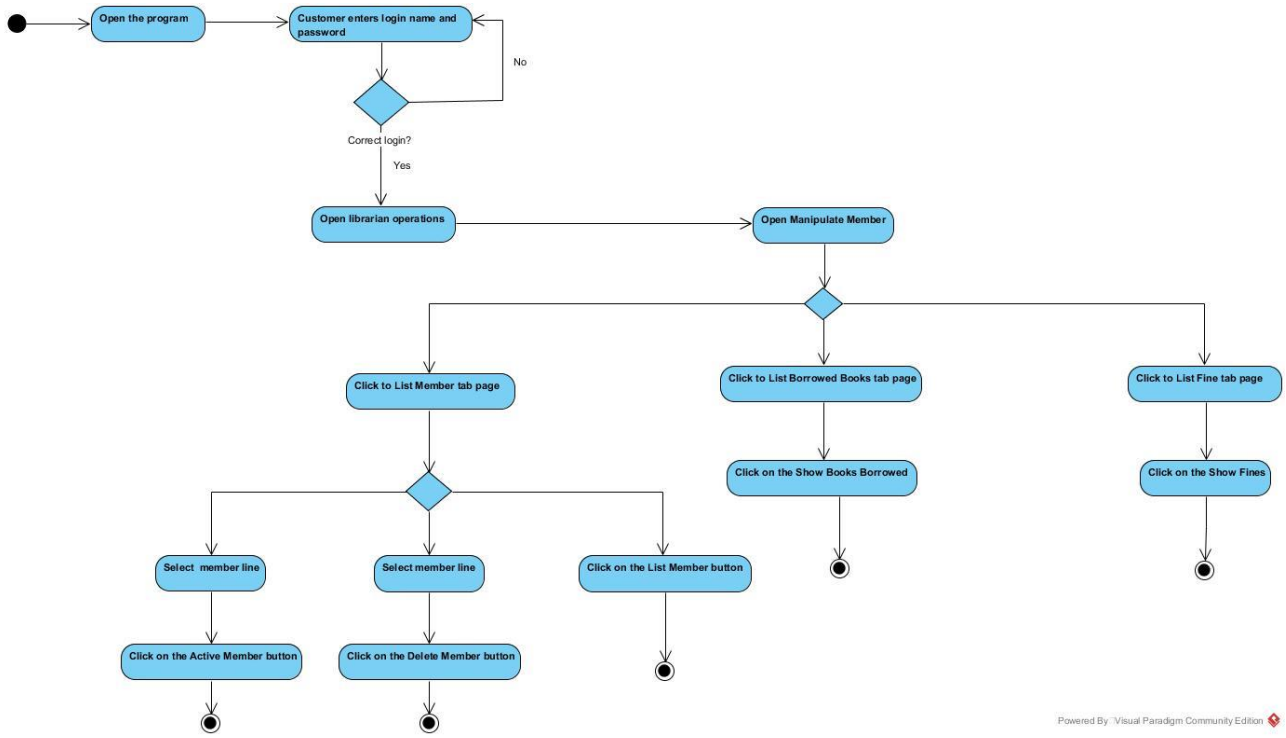
Activity Diagram: Manipulate Books



Use Case: Manipulate Members
Scenario:

1. Librarian opens the program.
2. Librarian enters username and password.
3. The system checks username and password's correctness.
4. The system delivers that username and password are correct.
5. The system opens librarian operations screen
6. Librarian opens Manipulate Members.
7. If librarian wants to see all member, librarian clicks to List Member tab page and clicks on the List Member button.
8. If librarian wants to delete a member, librarian selects member line and clicks on the Delete Member button.
9. If librarian wants to make active a member, librarian selects line and clicks on the Active Member button.
10. If librarian wants to see all books that borrowed, clicks to List Borrowed Books tab page and clicks on the Show Books Borrowed button.
11. If librarian wants to see all fine, clicks to List Fine tab page and clicks on the Show Fines button.

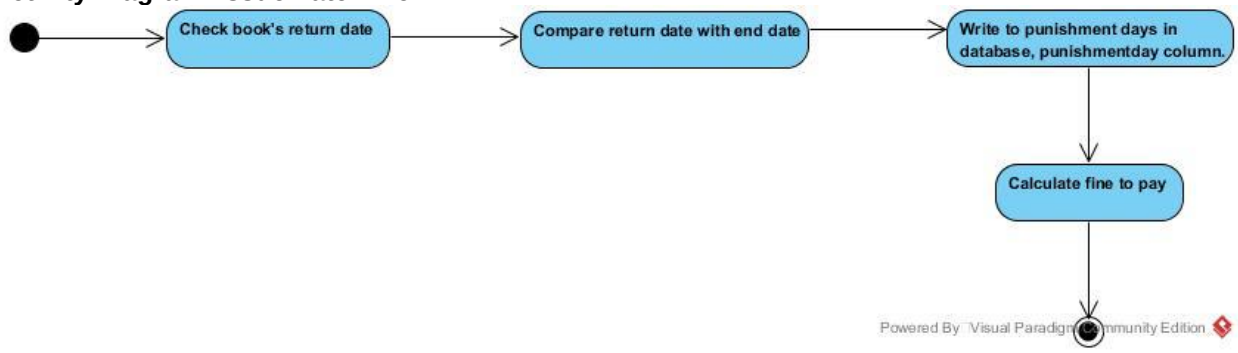
Activity Diagram: Manipulate Members



Use Case: Issue Late Fine
Scenario:

1. The system check books return date.
2. The system compares return date and end date of the book.
3. The system writes punishment day in database.
4. The system calculates fine to pay.

Activity Diagram: Issue Late Fine



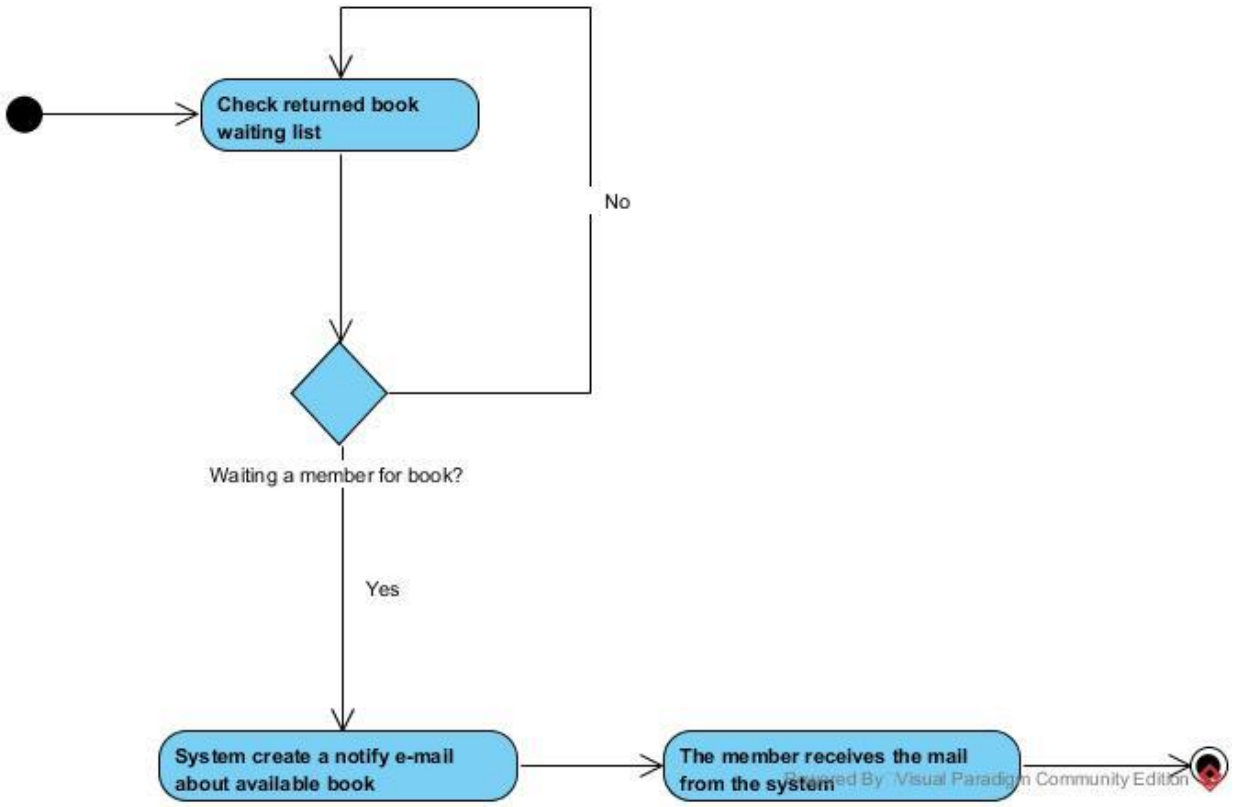
Use Case: Notify of Book Being Available
Scenario:

1. The system checks returned book on waiting list.
2. The system create a notify e-mail about available book.
3. The member receives the mail from the system.

Alternative Scenario:

1. If there is no member to wait the book, the system returns checks returned book waiting list step.

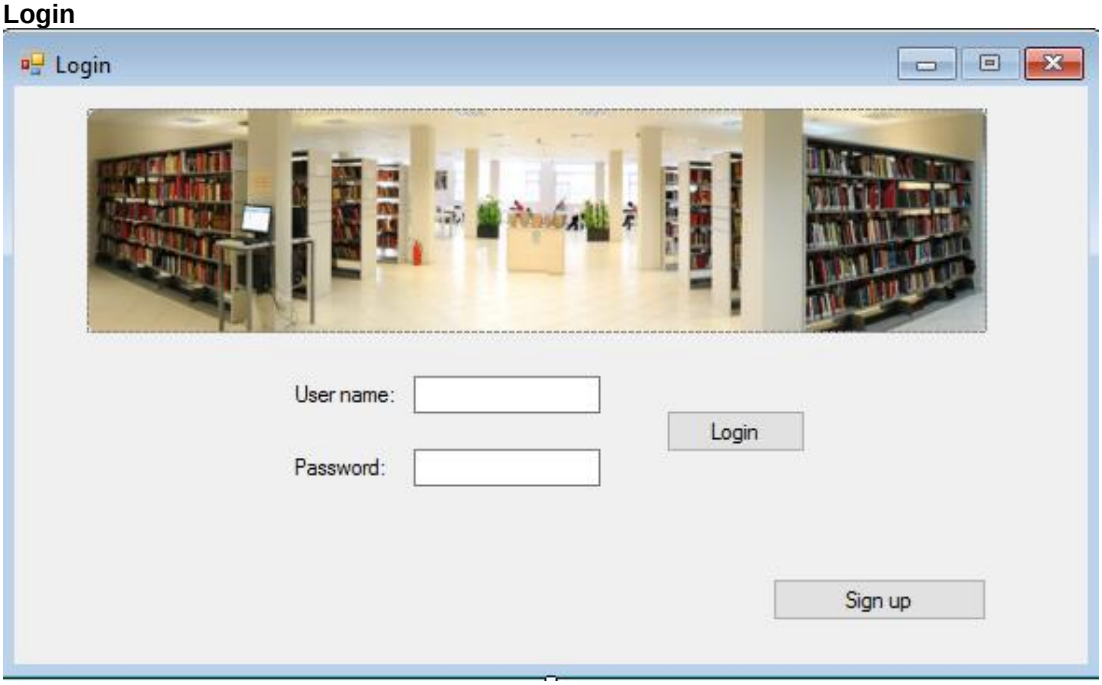
Activity Diagram: Notify of Book Being Available



Appendix B

This section contains the user interface of our system.

1. **Login**



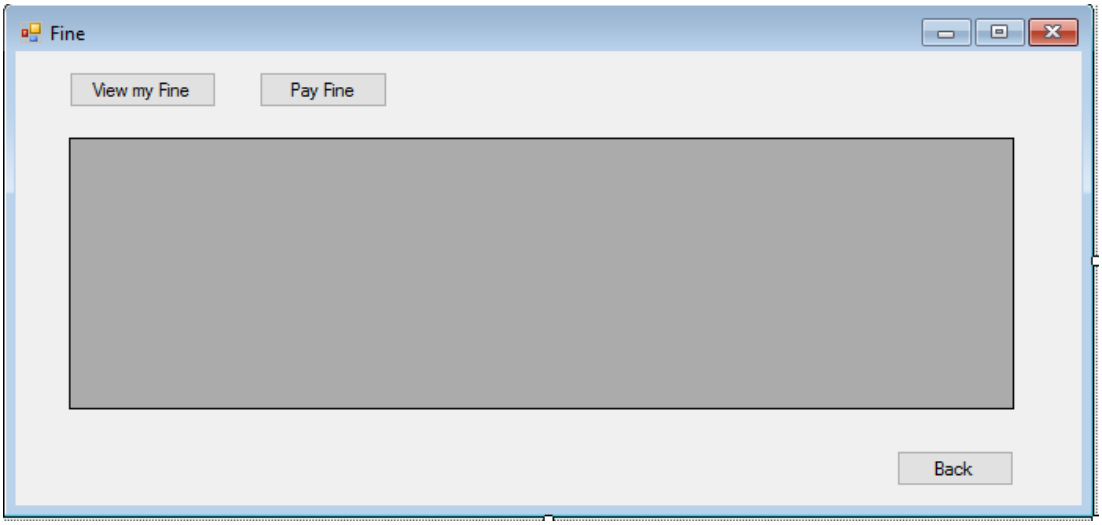
When the program executed users meet the main screen. This screen satisfies Login and Sign up

use cases.

2. **Member Operations**

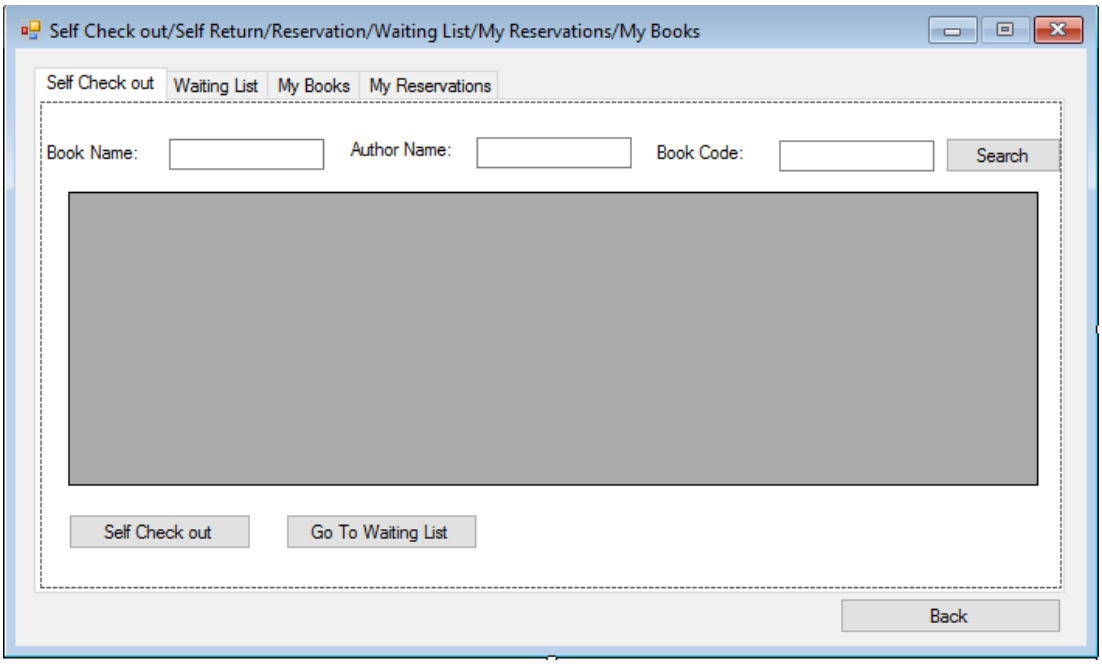
When the members login, meet this screen. This screen is an interface that directs the work that the members want such as take a book or pay fine. Also, members can logout from the system by using this interface. Thus it satisfies Logout use case.

3. **Fine**



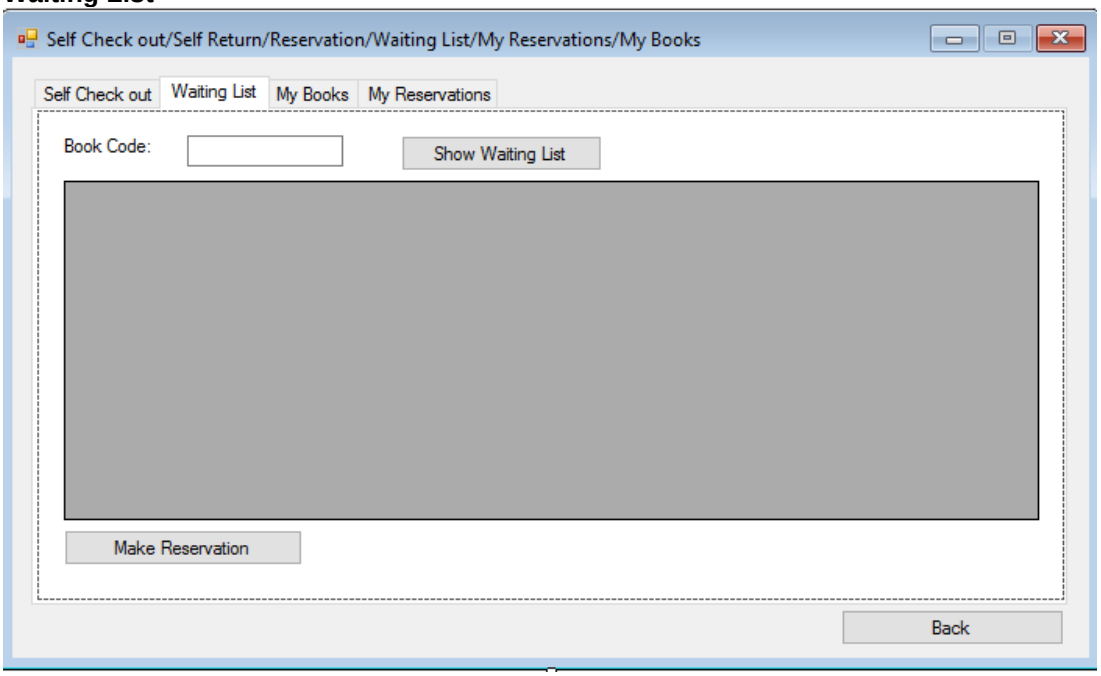
In this interface, members can see fine and pay it. It satisfy Pay Fine use case. Issue Late Fine is done by system and members can see it from this interface. With Back button user can go back to the previous screen.

4. **Search Book, Self Check Out**



In this interface there are several operations. While members take a book, write book's information and search on the system. Then, if members find the book select the line of the book and click Self Check Out button. By following these steps, Self Check Out will be done. tab. It also satisfy Self Check Out and Search Book use cases. If the book is not available but members still want, then members click on the Go To Waiting List button. With Back button members can go back to the previous screen. So, Self Check Out use case will be satisfied.

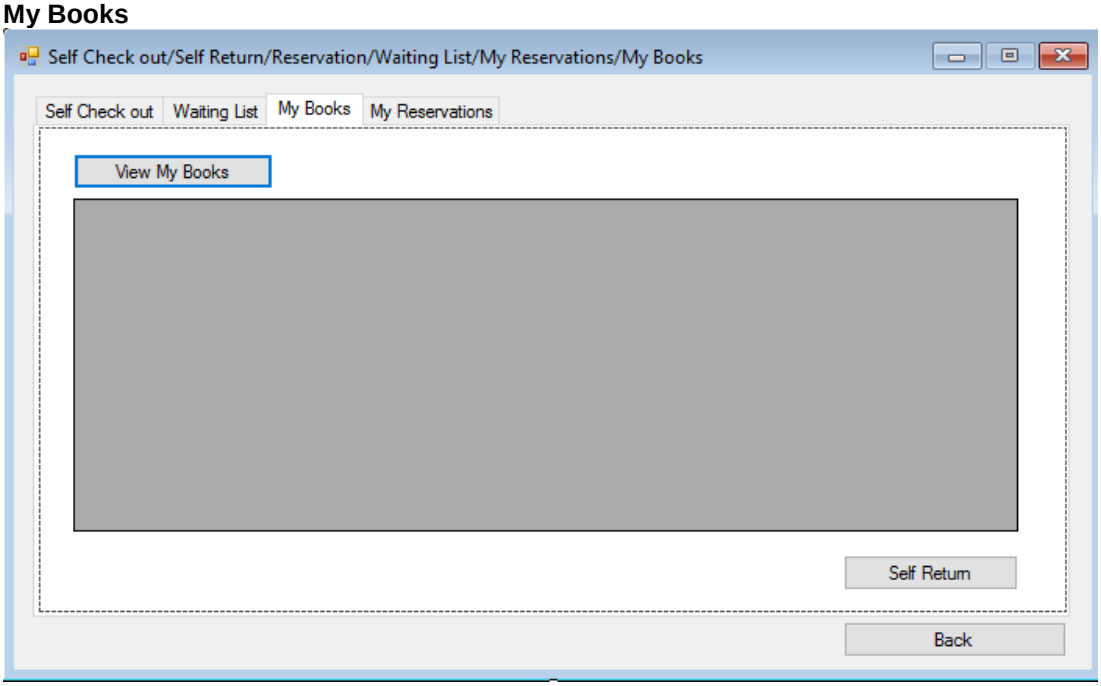
5. Waiting List



Members come to this screen from the Self Check Out tab by clicking Go To The Waiting List button so that the book code is entered automatically or enter the code of the book by themself. After that, members click the Show Waiting List and see the waiting list of the book and with Make Reservation button, members can add themself to the waiting list. With Back button

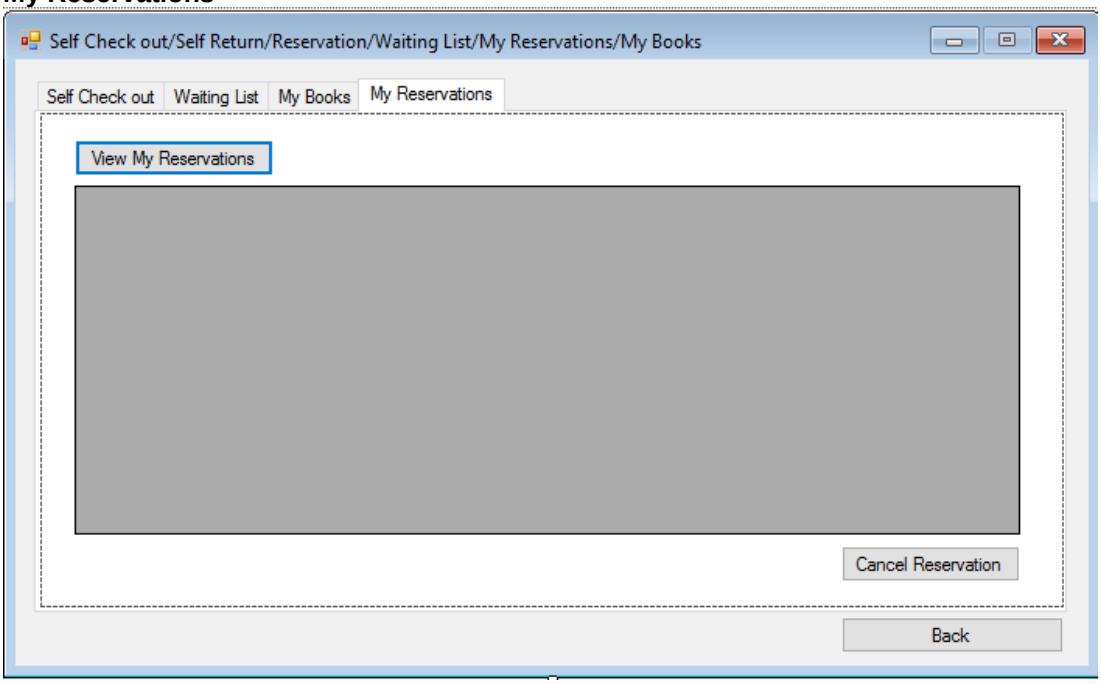
members can go back to the previous screen. So, Reservation Book use case will be satisfied.

6. My Books



In this interface, members see the books they have received. Then, members if select the line of a book and click the Self Return button, they return the book to the system. With Back button members can go back to the previous screen. So, Self Return use case will be satisfied.

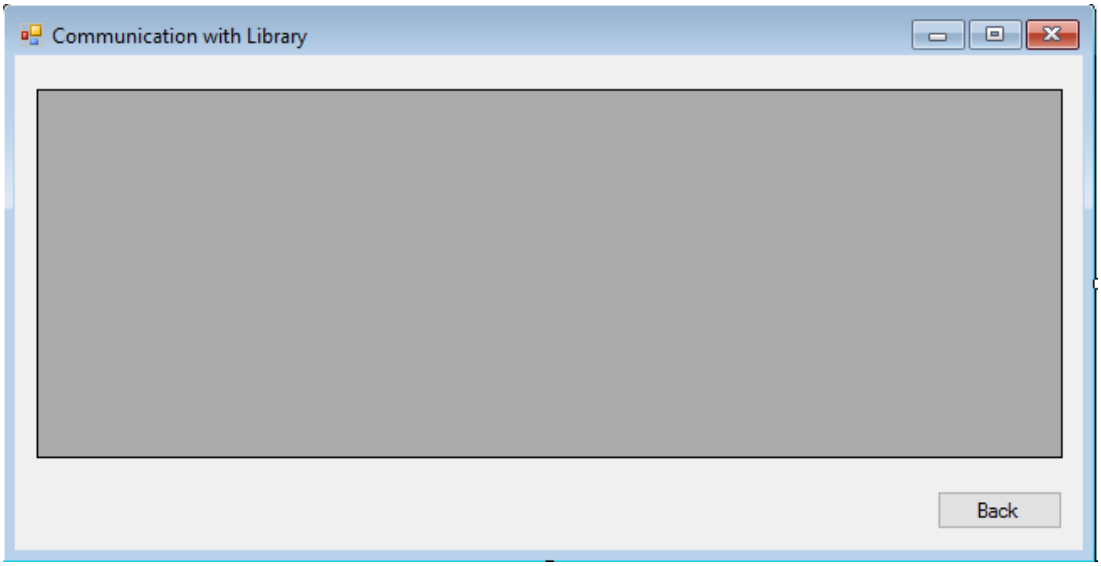
7. My Reservations



In this interface, members see their reservations. Then, members if select the line of a book and

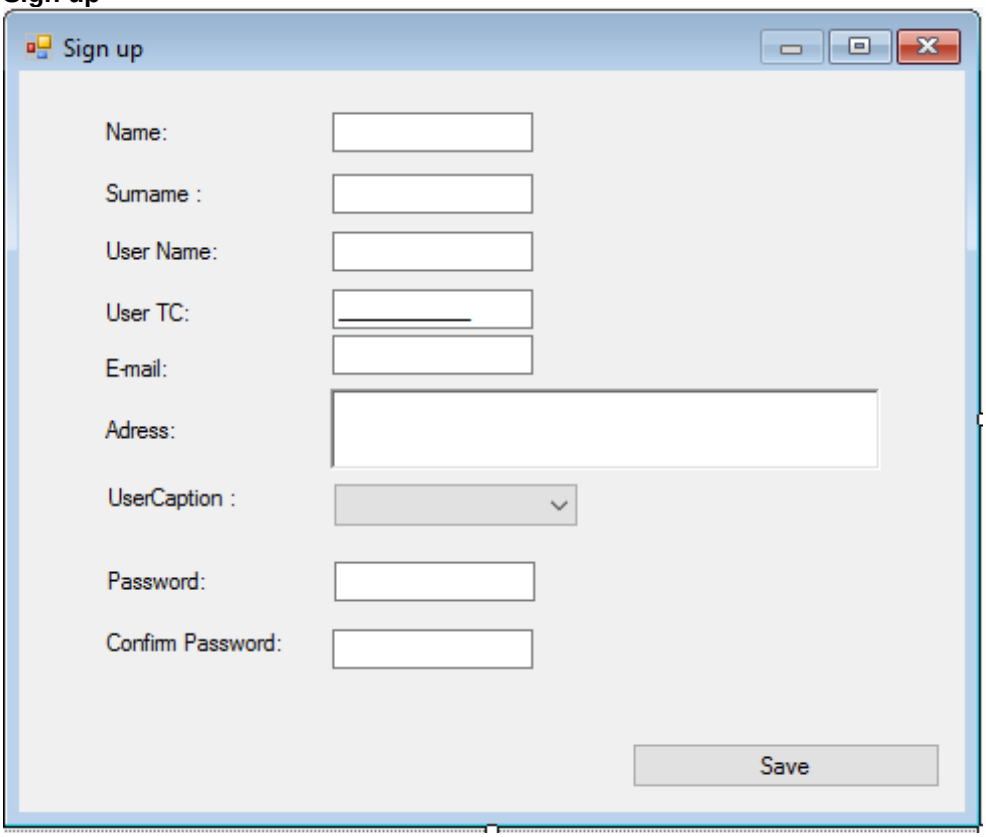
click the Cancel Reservation button, they cancel the reservation. With Back button members can go back to the previous screen.

8. **Communication With The Librarian**



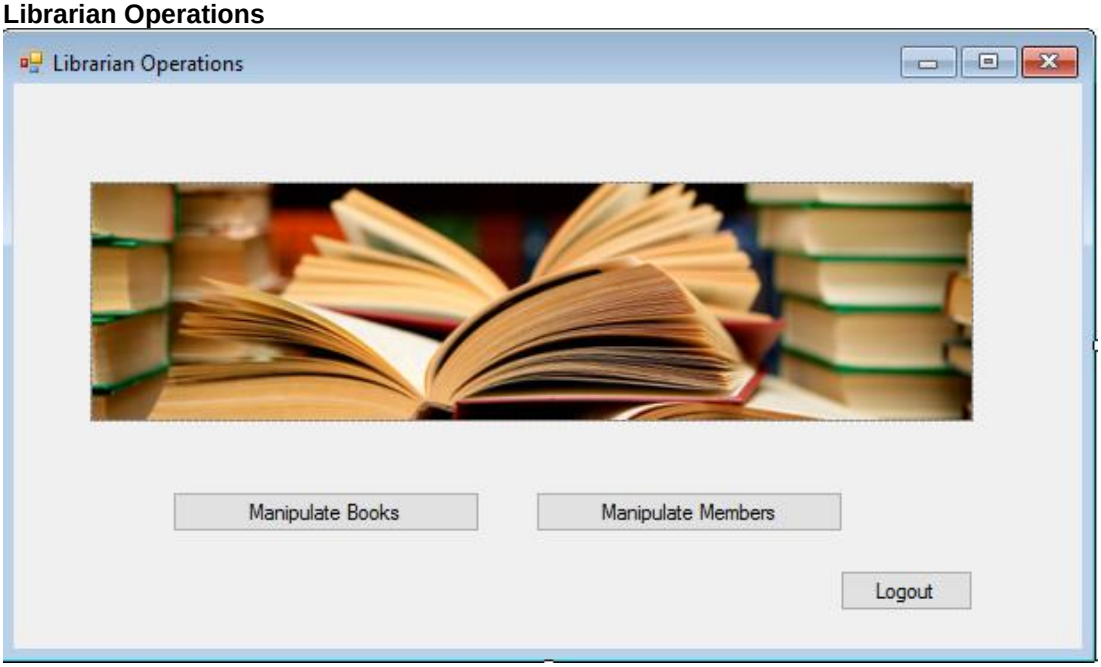
Members can communicate with the librarian using this interface. With Back button members can go back to the previous screen. It satisfy Communication With Librarian use case.

9. **Sign up**



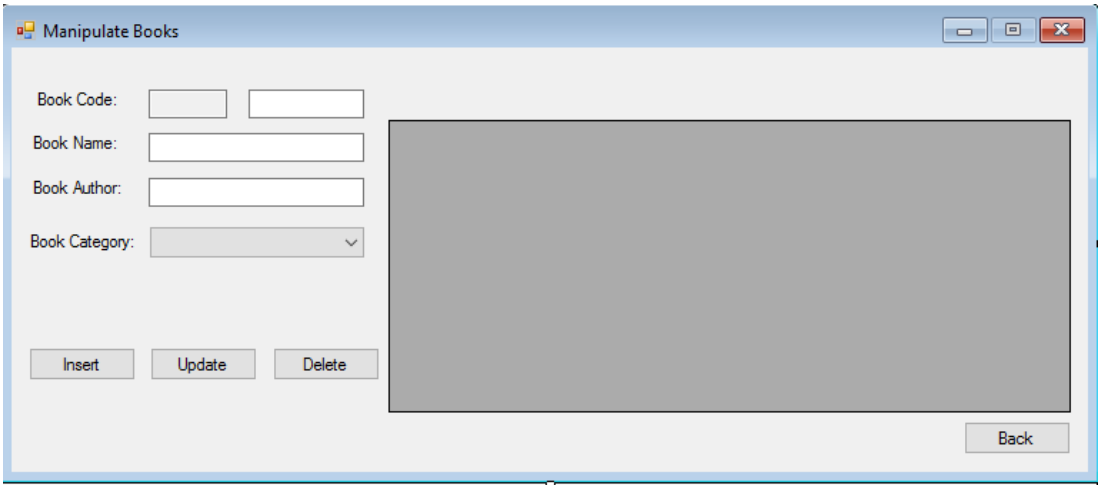
In this interface, users if not a member of the system, fill this form and sign up to our system. So, Sign up Use case will be satisfied.

10. Librarian Operations



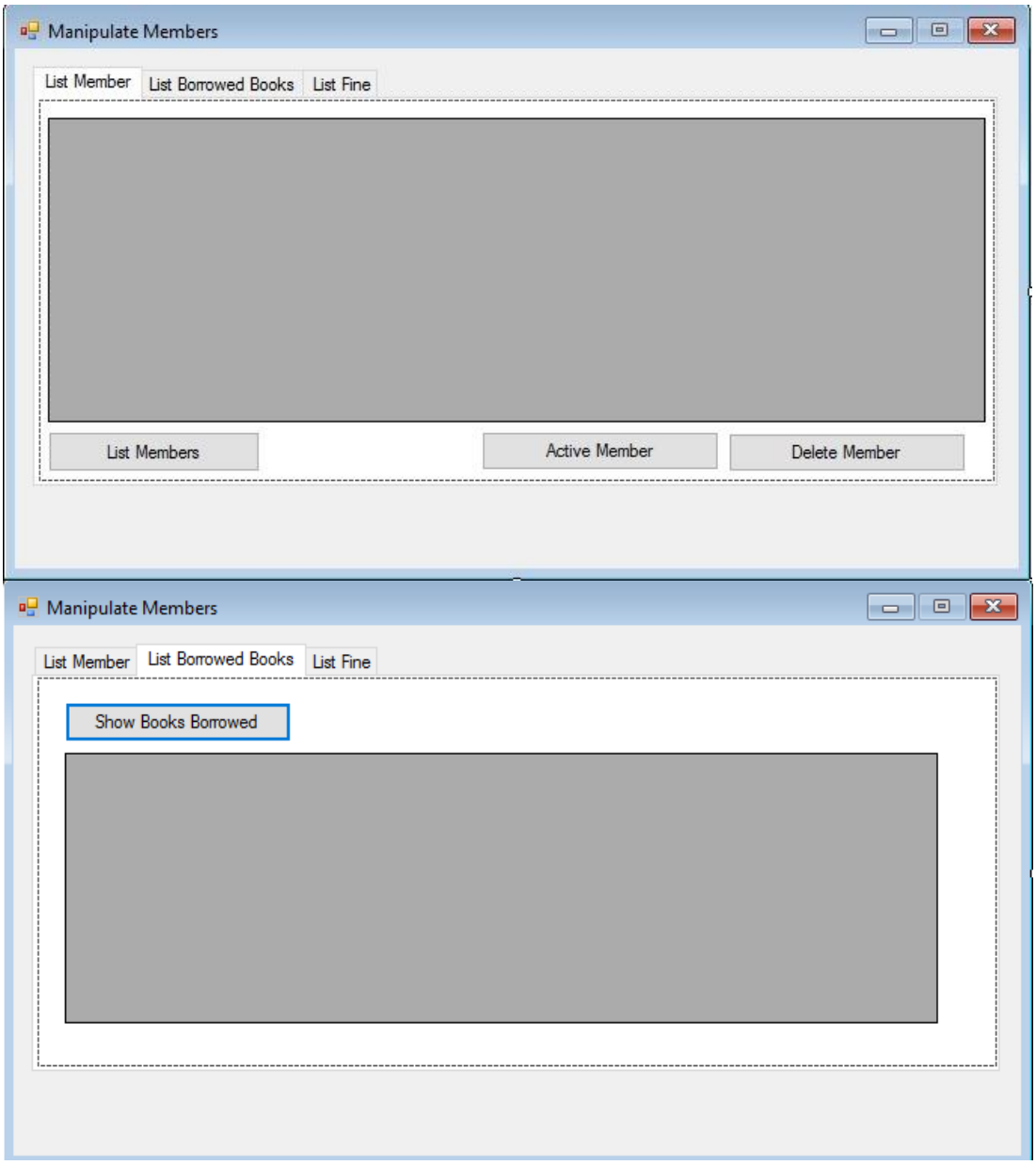
If the librarians login the system, they meet with this interface. From this interface, librarians are directed to the operations that they have authority to do so.

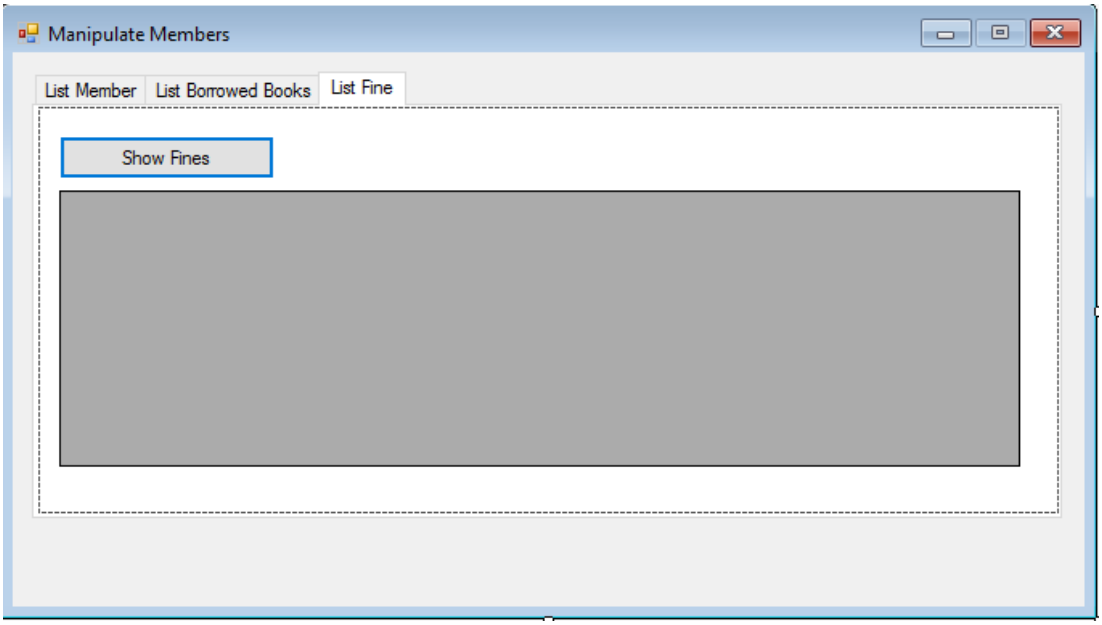
11. Manipulate Books



In this interface, the librarians enter the book information they have and system lists the books. They can Delete or Update book by select line of the book and insert a book by clicking Insert button. So, Manipulate Books use case will be satisfied.

12. Manipulate Members


Manipulate Members
List Member
List Borrowed Books
List Fine
List Members
Active Member
Delete Member
Manipulate Members
List Member
List Borrowed Books
List Fine
Show Books Borrowed



In this interface, the librarians managed members issues. They can list members, borrowed books and fine. Delete Member button make member's status inactive and Active Member button make the member active again. With Show Books Borrowed button, system lists books that borrowed and with Show Fine button, the system show fines.

Appendix C

E/R Diagram of our system is as following.