HACETTEPE UNIVERSITY

Computer Engineering
BBM487 Software Engineering Laboratory

Library Book Loan System
Software Design Description

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

Book Loan System
Software Design Description

1. Revision History

Version	*Date*	*Author*	*Change Description*
1.0	19.04.2017	Gülendam Buket Gündüz	ER Diagram prepared.
1.1	23.04.2017	Gülşah Erdoğan	Sequence diagrams added.
1.2	25.04.2017	Gülendam Buket Gündüz Gülşah Erdoğan İrem Özen	Class diagram and traceability matrix is added.

2. INTRODUCTION

2.1 Purpose and Scope

This document includes the software details, detailed user interfaces and detailed data model of our project which we planned earlier. Also, in this document, there are class diagram and sequence diagrams of our system. We aim to have a better understanding of the system with this document.

2.2 Document Overview

This document contains information that will make the stage of our system more understandable by the stakeholders.

In the 3rd chapter of this document, we explained the restrictions and decisions of our application.

In the 4th chapter, there are class diagram, sequence diagrams of every use case, ER diagram of our system and interfaces of use cases.

In the 5th chapter of this document, there is requirement-class traceability matrix

2.3 System Overview

Our system is designed as a desktop application that can be easily used by university students, academicians and librarians and will reduce the troubles that they have experienced while manually taking books, giving books, managing members, etc.

Within the constraints we have specified, we aimed to enable users to quickly handle library works automatically. We have defined operations according to the user types in our system which is designed based on MVC system architecture so that the library work will be more organized.

2.4 Definitions, Acronyms, and Abbreviations

etc: et cetera, it is used to avoid giving a complete list
MVC: Model-View-Controller software architectural pattern.
SRS: System-wide Requirement Specification document

2.5 References

The following documents have been referred to in this document:

1. Project Plan and Vision, Version No: 1.0 Version Date: 7 March 2017
2. System-wide Requirements Specification, Version No: 2.0 Version Date: 21 March 2017
3. Architecture Notebook, Version No: 3.0 Version Date: 4 April 2017
4. https://msdn.microsoft.com/en-us/library/ff649643.aspx

3. Design Constraints and Decisions

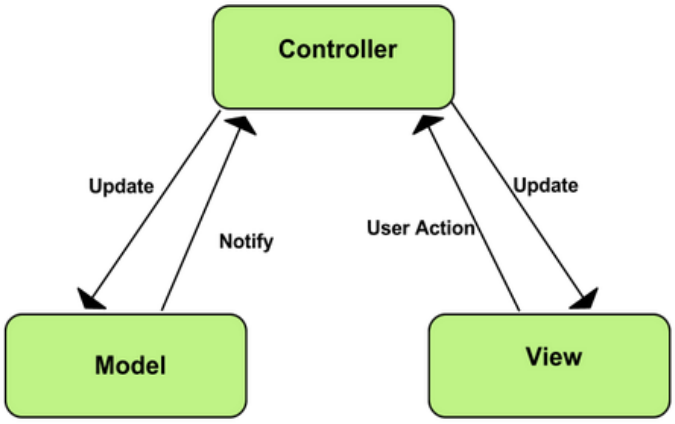
Our system will be designed to easily add and update features later on. The interfaces will be similar to each other, so that a simple and user-friendly interface will be designed so that the user can easily understand and use the system.

Our system will be made as a desktop application that can be used 7/24 in Windows operating system. We will use the MVC architectural pattern. Our system depends on the MySQL database. Therefore, the performance of our system will depend on the speed of queries retrieved from the database and on the algorithm of the code written. In case of wrong password entry etc., the system will give error message and guide the user.

4. Design Details

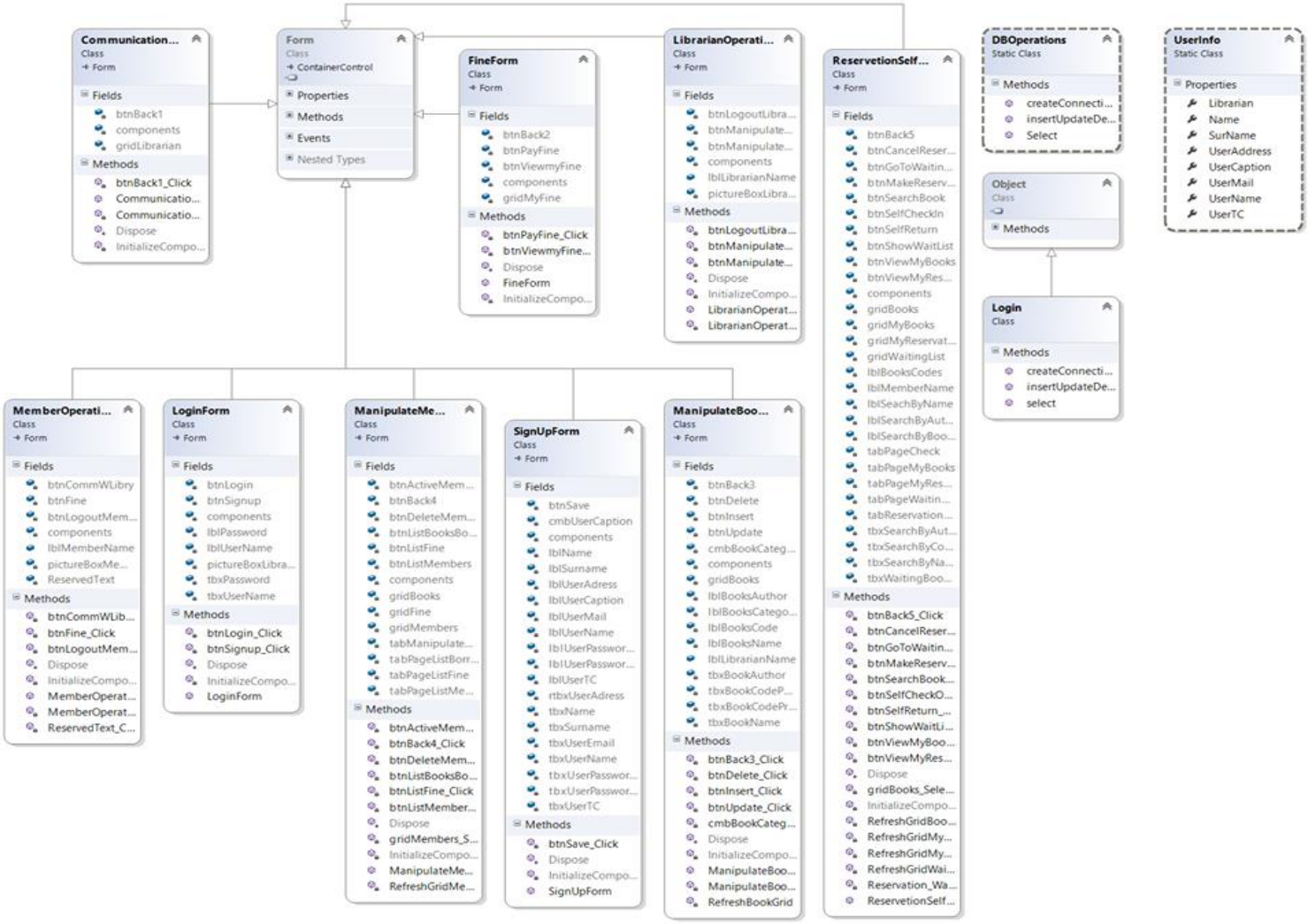
We use Model-View-Controller (MVC) pattern to describe our Book Loan System. MVC is a software architectural pattern for implementing user interfaces on computers. It divides a given application into three interconnected parts.

Model: The model manages the behavior and data of the application domain, responds to requests for information about its state (usually from the view), and responds to instructions to change state (usually from the controller).
View: The view manages the display of information.
Controller: The controller interprets the mouse and keyboard inputs from the user, informing the model and/or the view to change as appropriate.



4.1 Software Components

Class diagram of our system is as following:

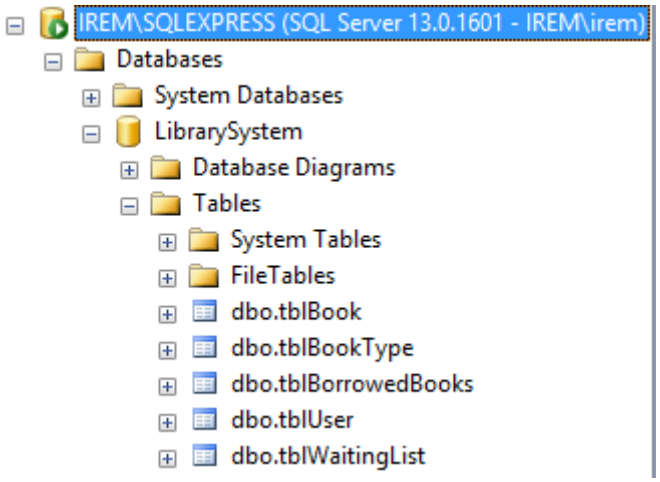


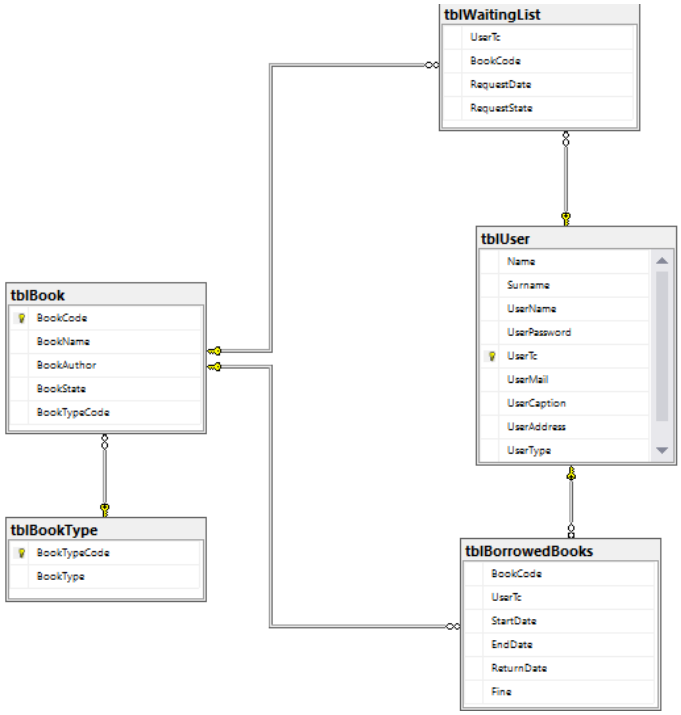
4.2 Software Behavior

The sequence diagrams of the use cases we have described in the System-wide Requirement Specification(SRS) document are found in Appendix A.

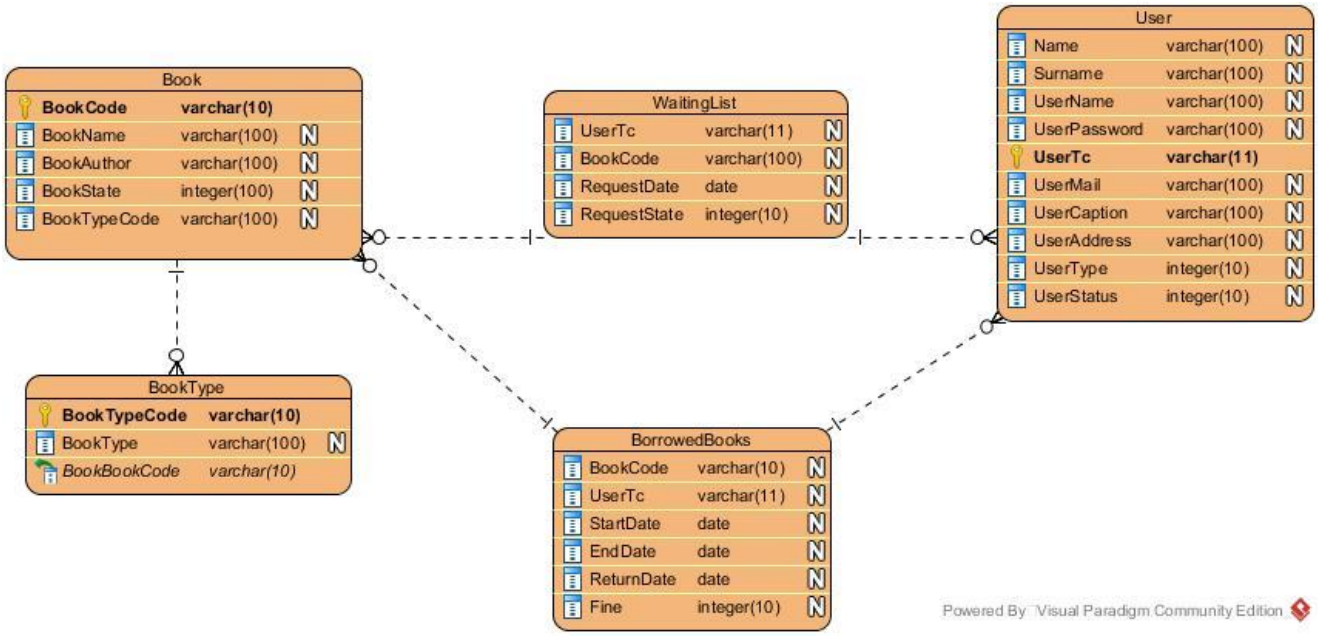
4.3 Data Model (E-R Diagram)

Our database tables are as following.





E/R diagram of our system is not changed. It is as following.



4.4 User Interface Design

The interfaces we have specified in the SRS document are developed and detailed in Appendix B.

5. Requirements Traceability

As we planned earlier, we have prepared 60 percent of the requirements that we have detailed in the SRS document. We also implemented 11 of 18 use cases that we mentioned in the SRS document. These are like the following table.

Note: Numbers in the table is refers to requirements. Requirements number table is in the below.

Requirement/Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
CommunicationLibraryForm.cs											X					
DBOperations.cs	X		X	X	X	X	X	X	X	X		X	X	X	X	X
FineForm.cs										X						X
LibrarianOperationsForm.cs		X											X	X		
Login.cs	X															
LoginForm.cs	X		X													
ManipulateBooksForm.cs													X			
ManipulateMembersForm.cs														X		
MemberOperationsForm.cs		X		X	X	X	X	X	X	X	X				X	X
Reser_Selfche_SelfretForm.cs				X	X	X	X	X	X			X				
SignUpForm.cs			X													
UserInfo.cs	X		X	X			X									

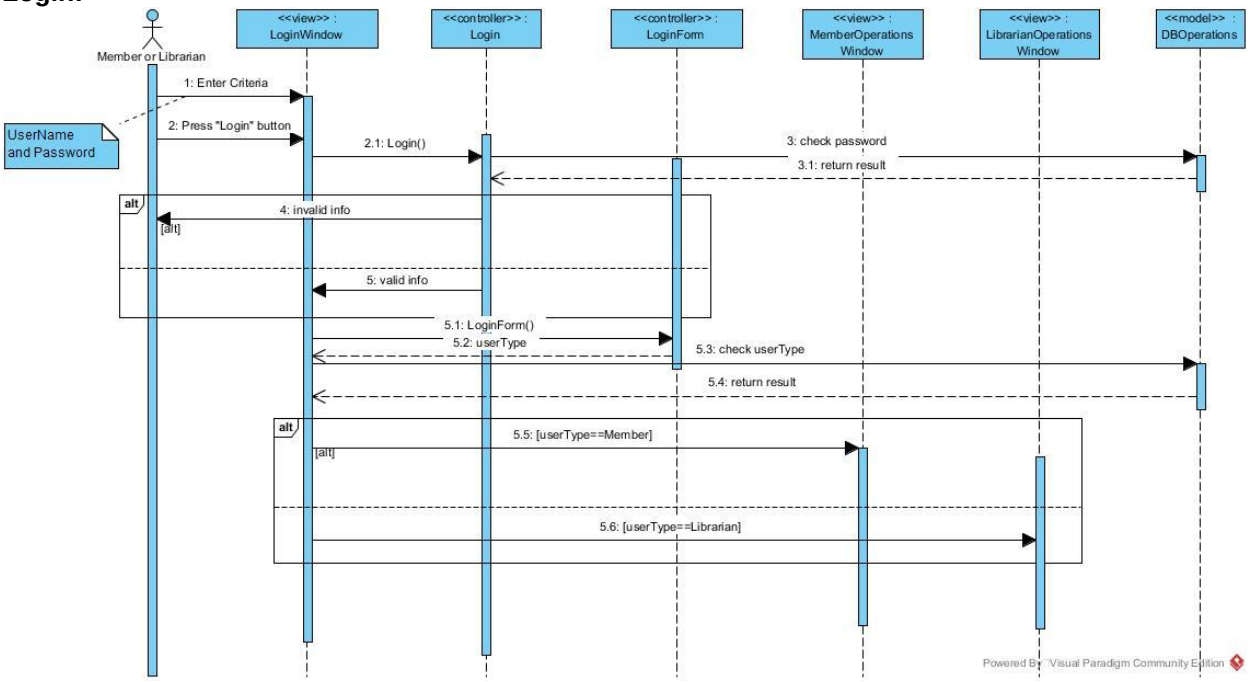
No	Requirement
1	Login
2	Logout
3	Sign up
4	Make Reservation
5	Cancel Reservation
6	Search Book
7	Self Check Out
8	Self Return
9	View My Books
10	Pay Fine
11	Communication with the Librarian
12	Waiting List
13	Manipulate Book
14	Manipulate Member
15	Notify of Book Being Available
16	Issue Late Fine

6. Annexes

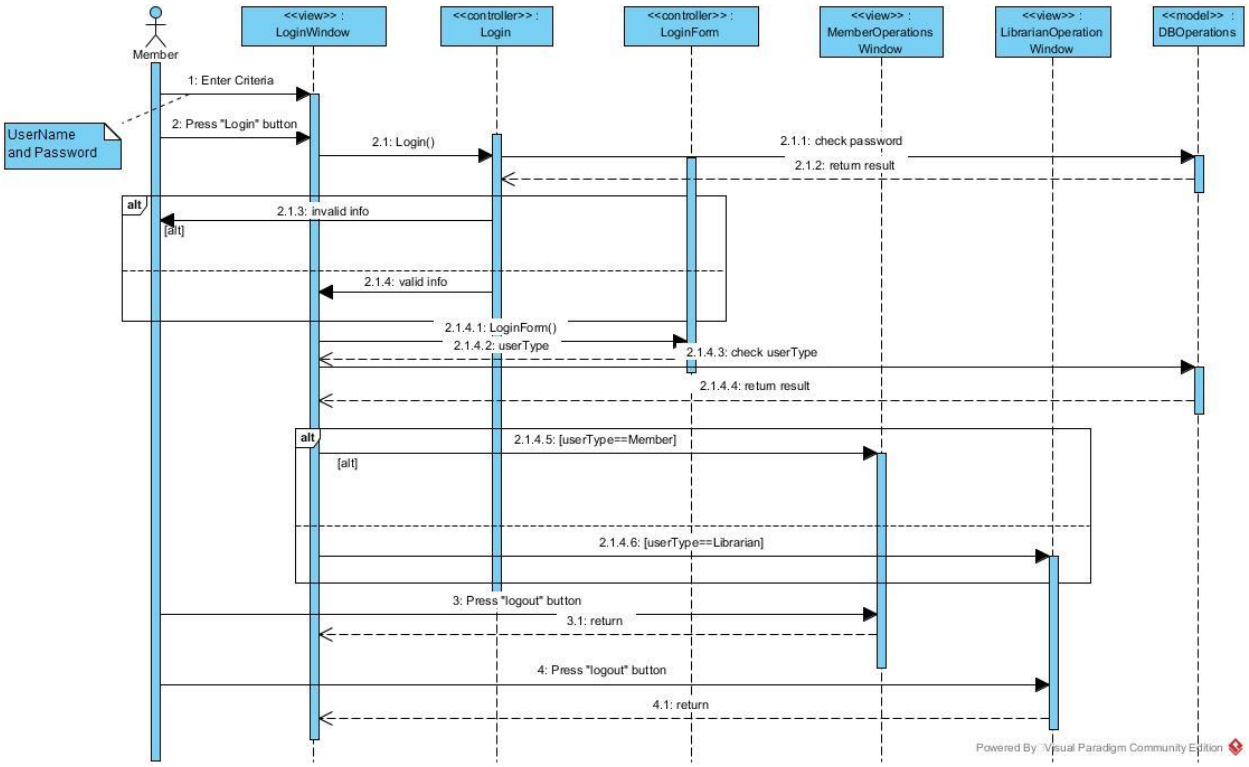
- Appendix A
- Appendix B

Appendix A:

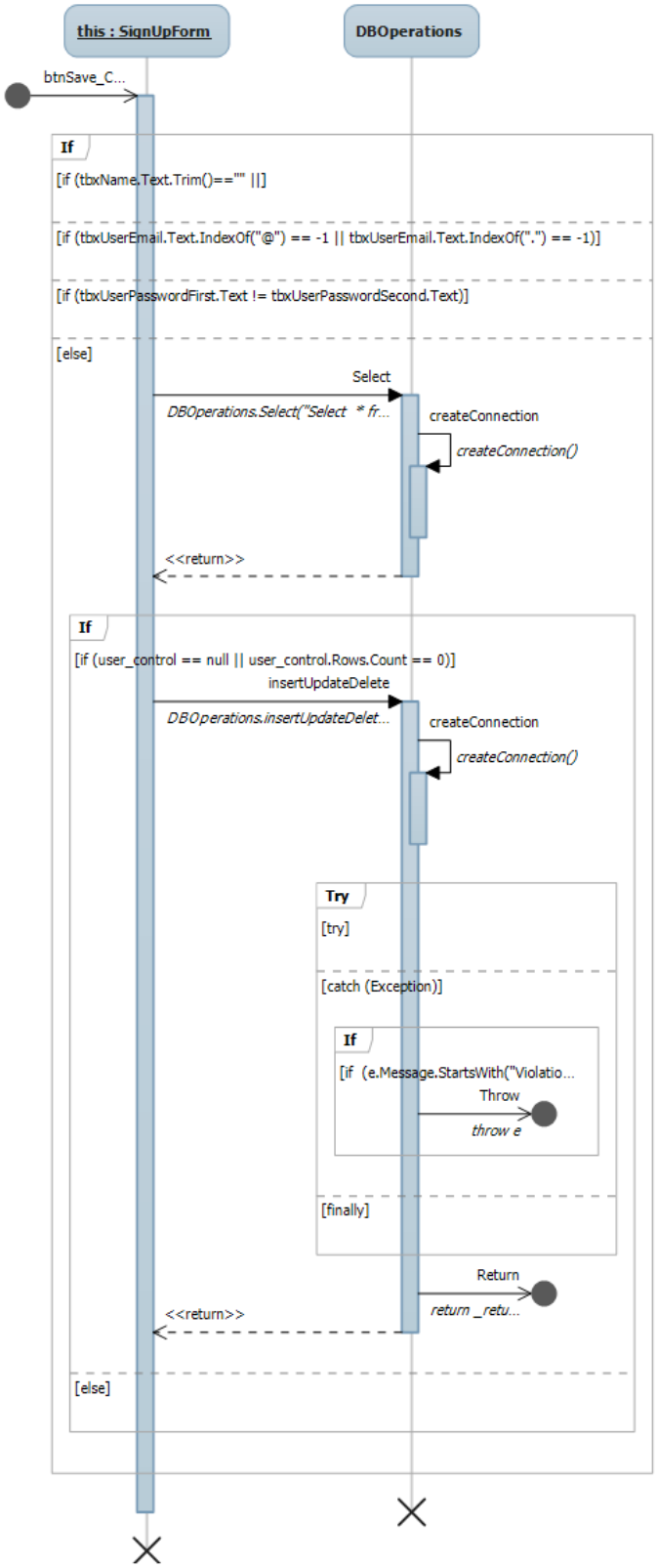
Login:



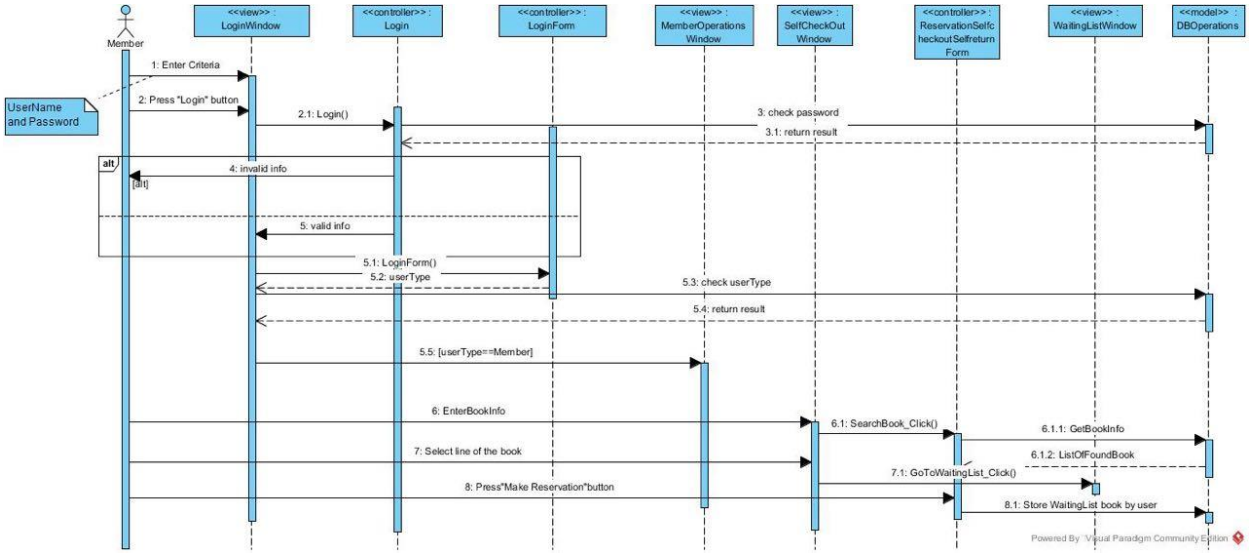
Logout:



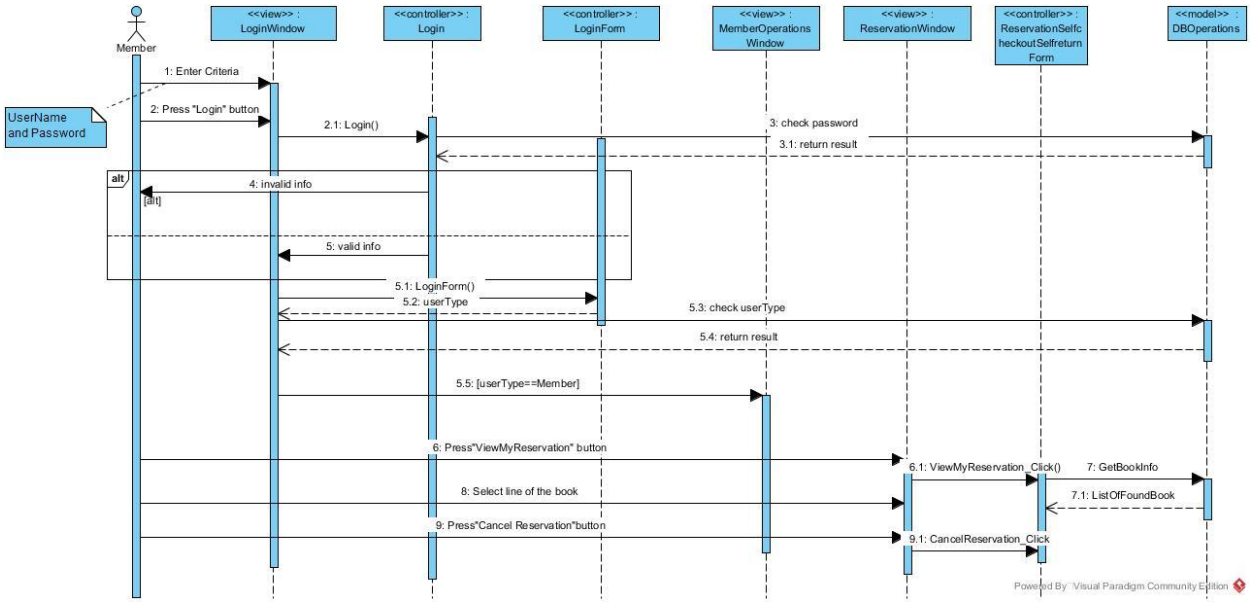
Sign up:


this : SignUpForm
DBOperations
btnSave_C...
If
[if (tbxName.Text.Trim()=="" ||]
[if (tbxUserEmail.Text.IndexOf("@") == -1 || tbxUserEmail.Text.IndexOf(".") == -1)]
[if (tbxUserPasswordFirst.Text != tbxUserPasswordSecond.Text)]
[else]
Select
DBOperations.Select("Select * fr...
createConnection
createConnection()
<<return>>
If
[if (user_control == null || user_control.Rows.Count == 0)]
insertUpdateDelete
DBOperations.insertUpdateDelet...
createConnection
createConnection()
Try
[try]
[catch (Exception)]
If
[if (e.Message.StartsWith("Violatio...
Throw
throw e
[finally]
Return
return _retu...
<<return>>
[else]

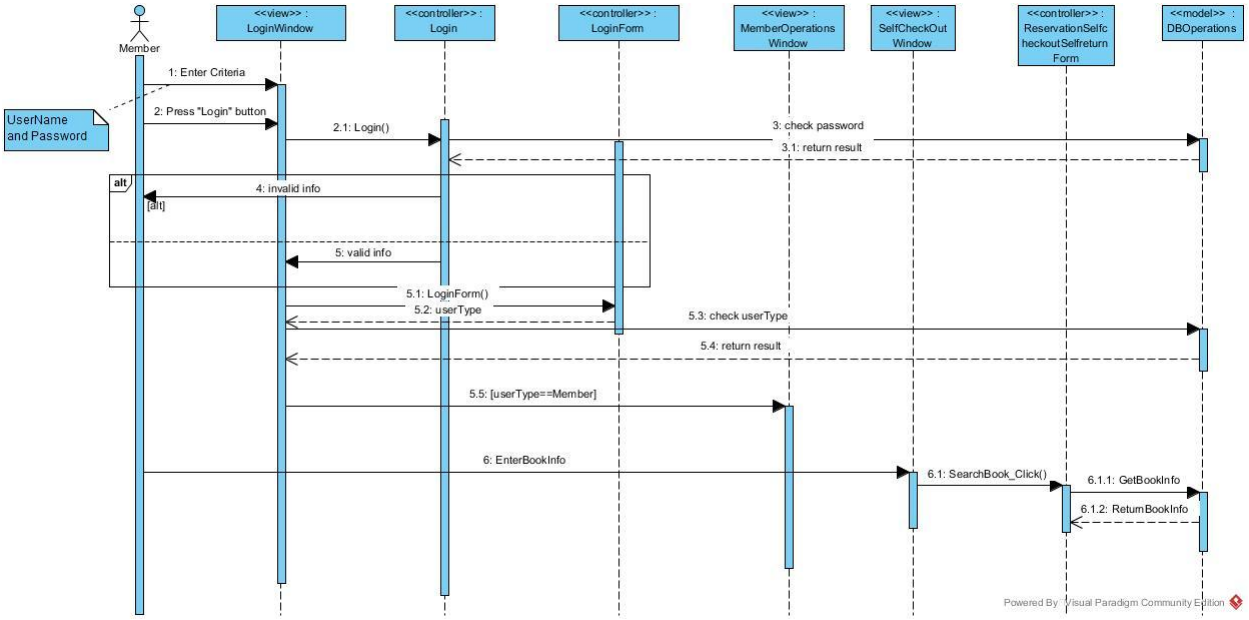
Make Reservation:



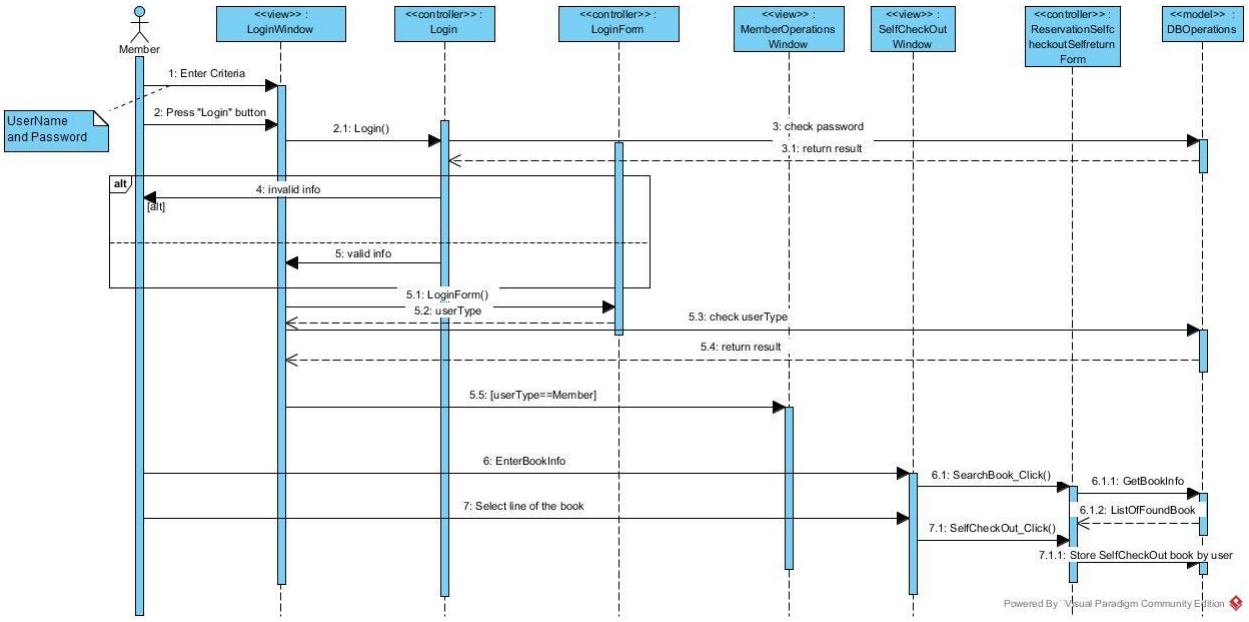
Cancel Reservation:



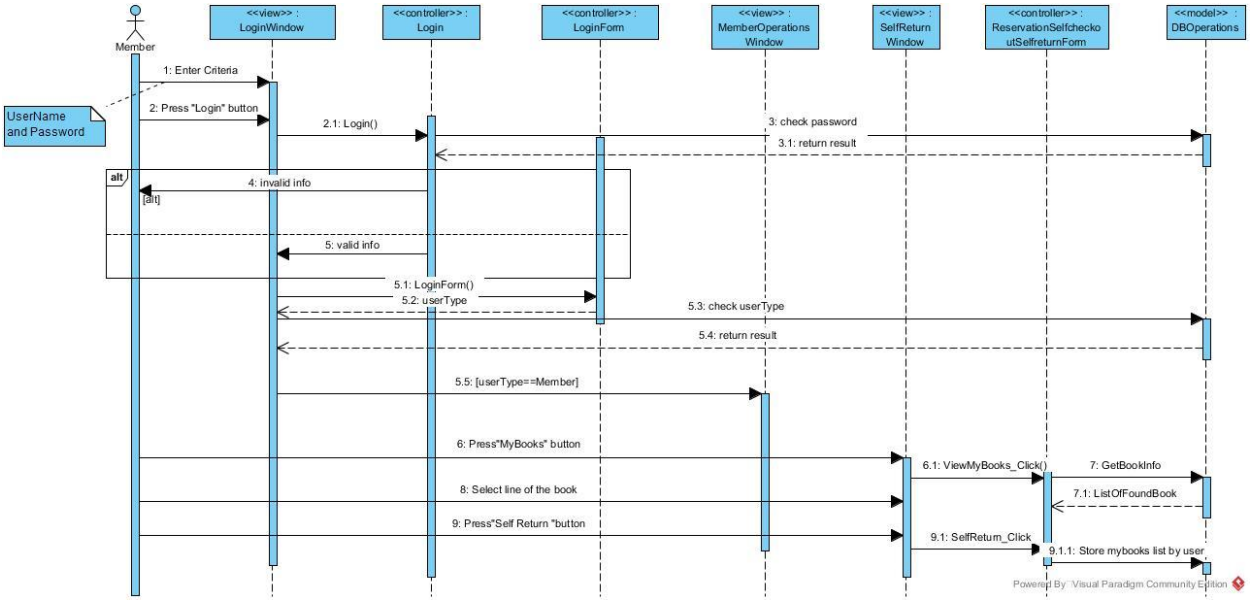
Search Book:



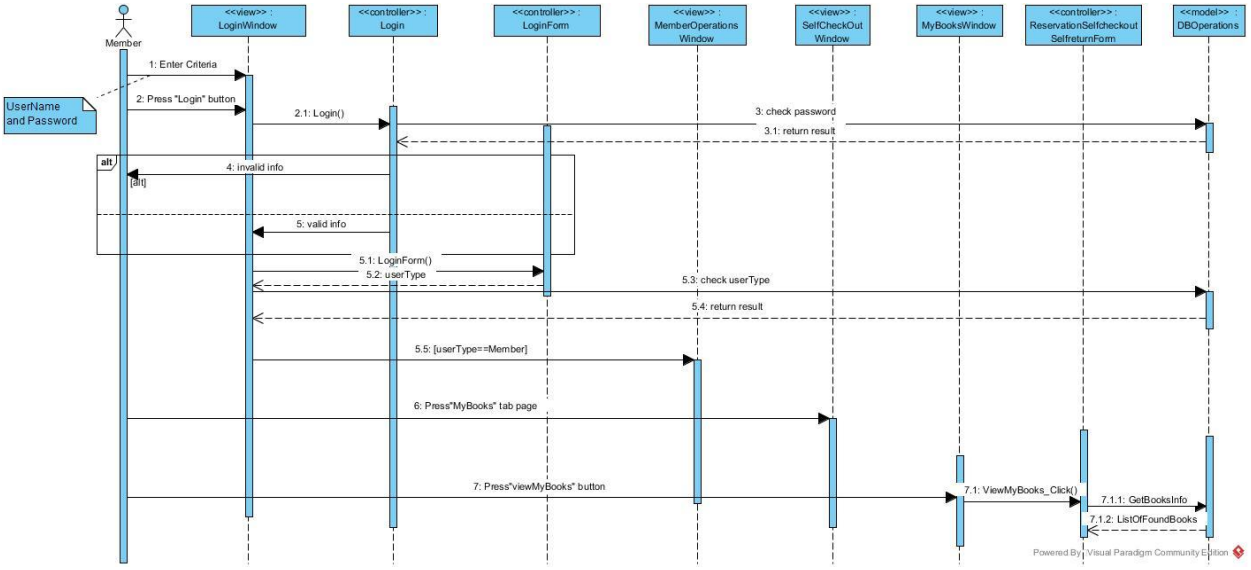
Self Check Out:



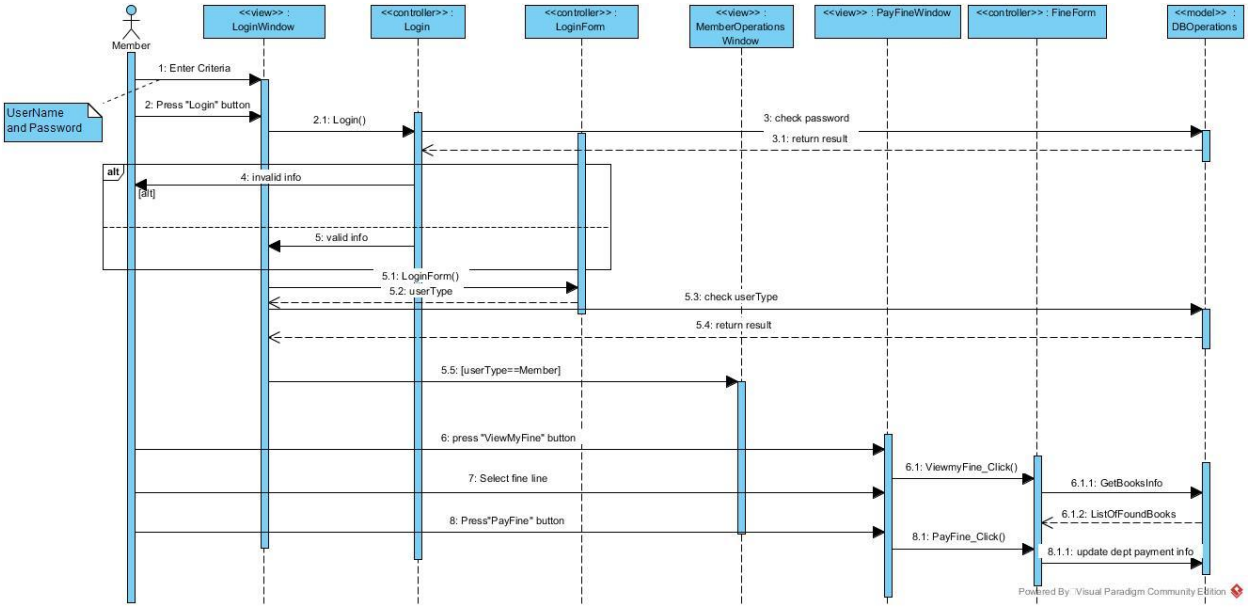
Self Return:



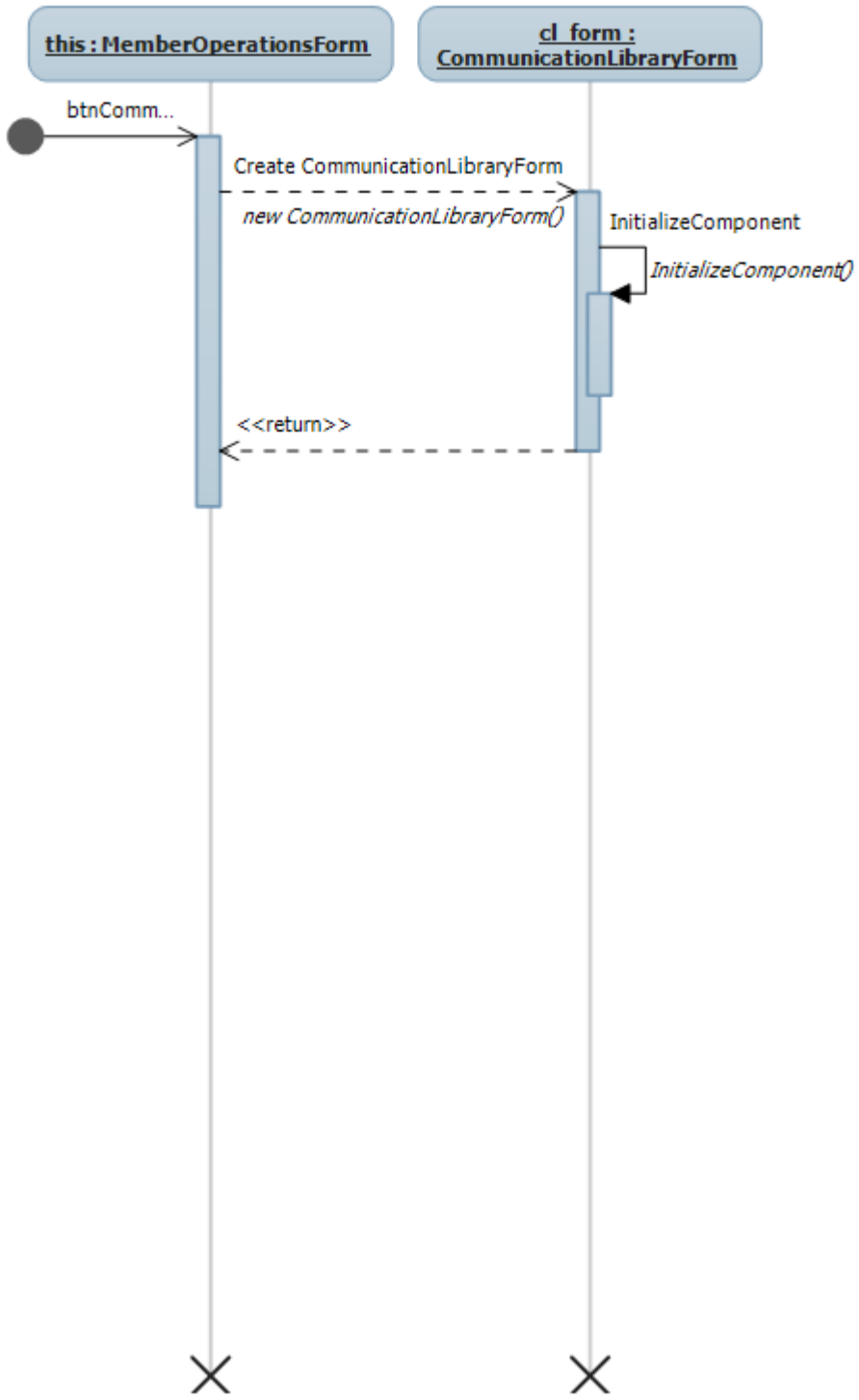
View My Books:



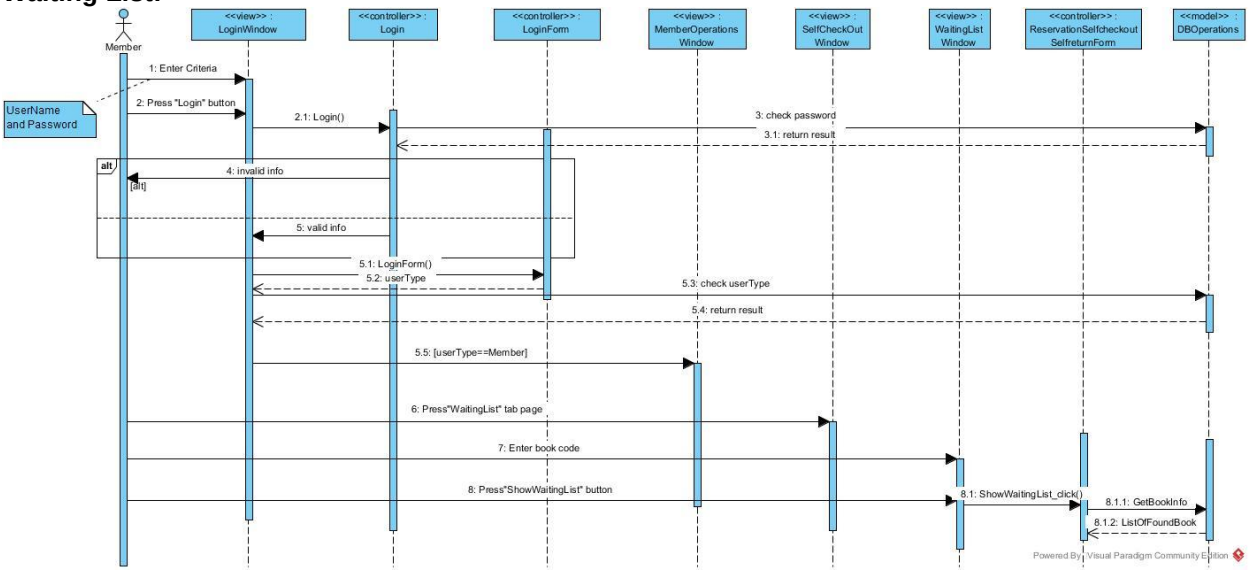
Pay Fine:



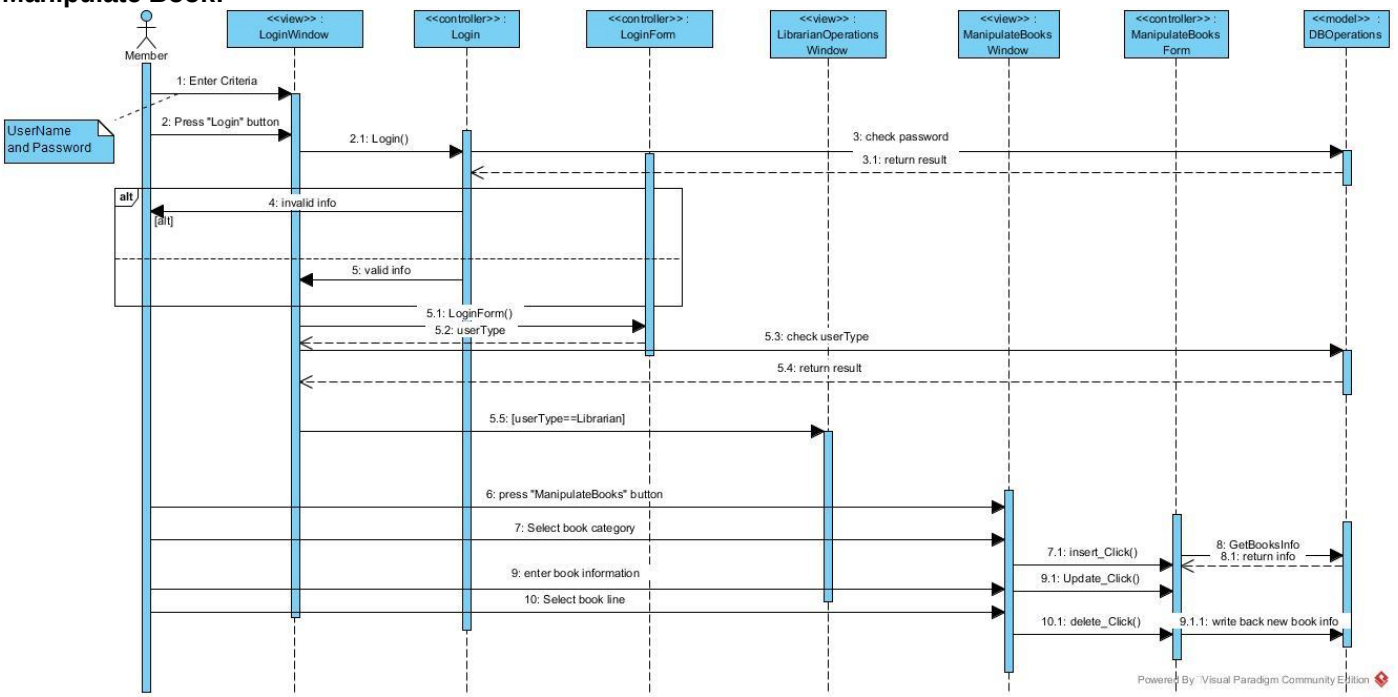
Communication with the Librarian:


this : MemberOperationsForm
cl_form : CommunicationLibraryForm
btnComm...
Create CommunicationLibraryForm
new CommunicationLibraryForm()
InitializeComponent
InitializeComponent()
<<return>>

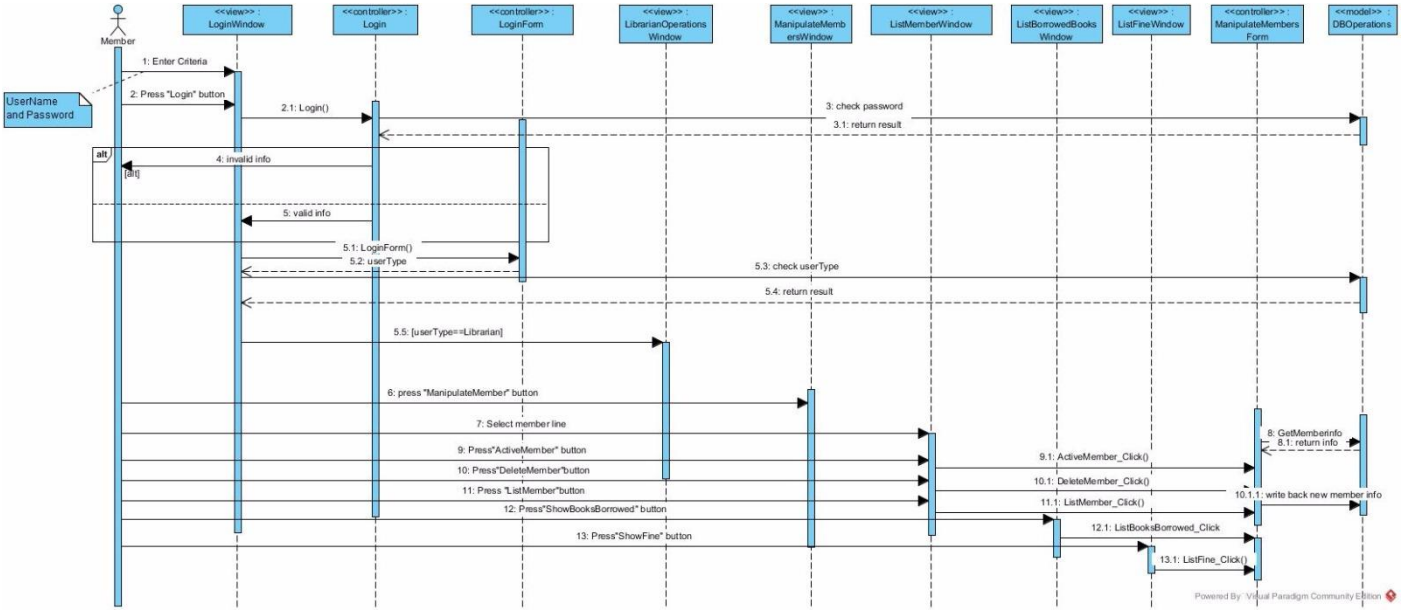
Waiting List:



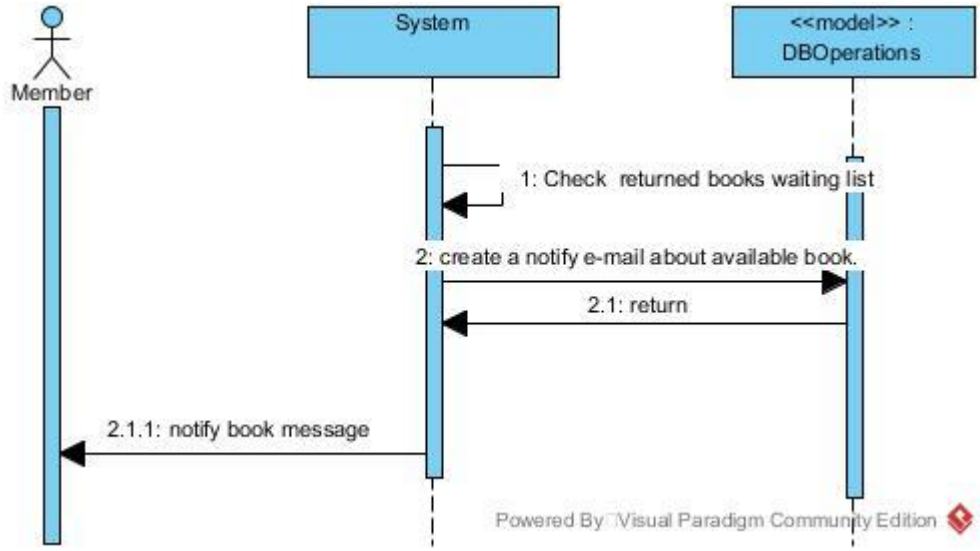
Manipulate Book:



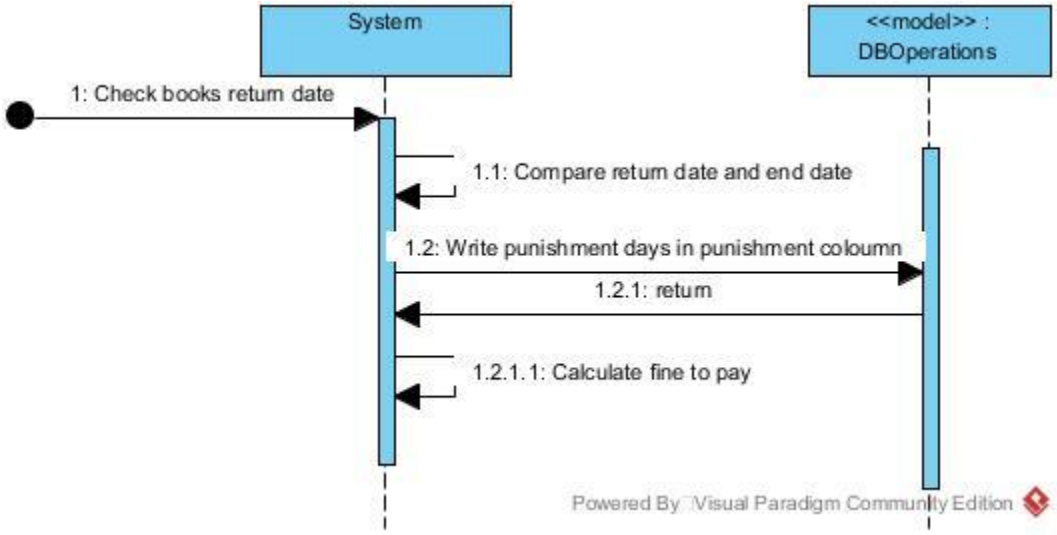
Manipulate Member:



Notify of Book Being Available:



Issue Late Fine:



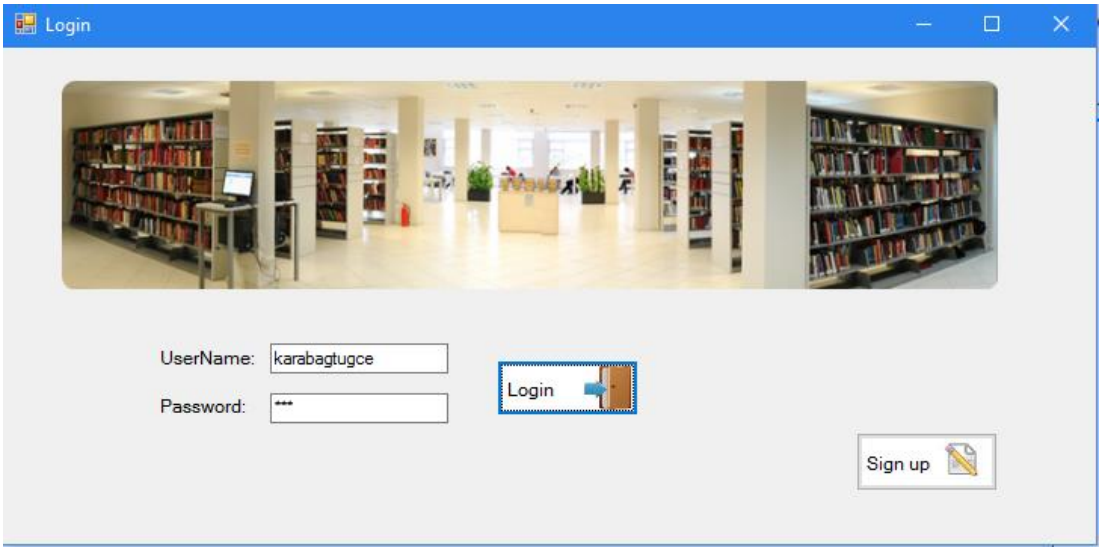
Appendix B:

In general, at all interfaces, we made changes in the buttons. We added buttons icons to remind users what that button does. Such as:

- Login button has a open door icon.
- Logout button has a closed door icon.
- Back button has an arrow icon that point back.
- Search button has a magnifying glass icon.
- Sign up button has a form icon.
- Go to Waiting List button has an arrow icon too. For this button arrow points right where waiting list screen is.

In addition to these, we have increased the font size for the user to have trouble reading.

Login:



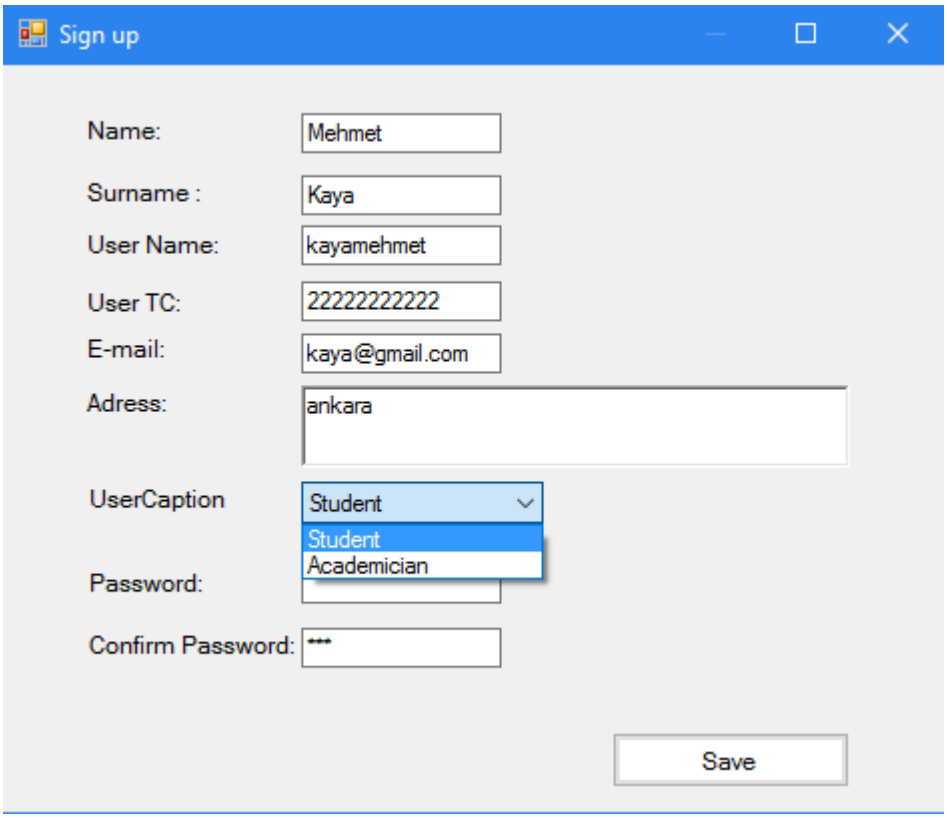
This is opening interface of our system.

Logout:

This is logout interface for member.

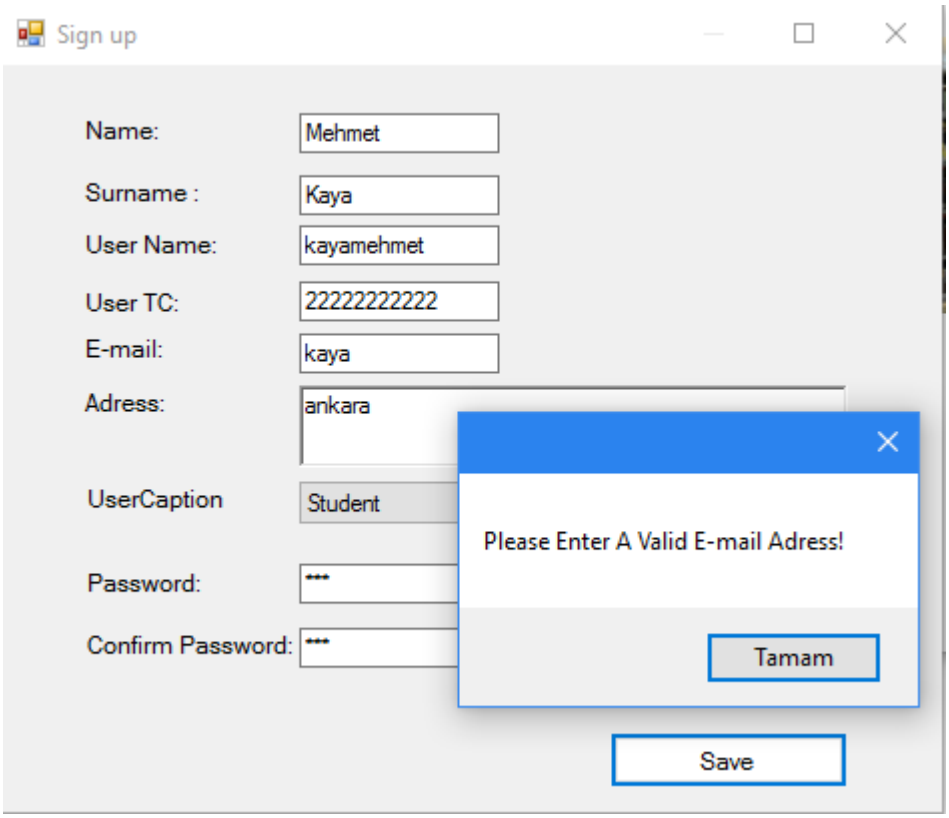
This is logout button for librarian.

Sign up:

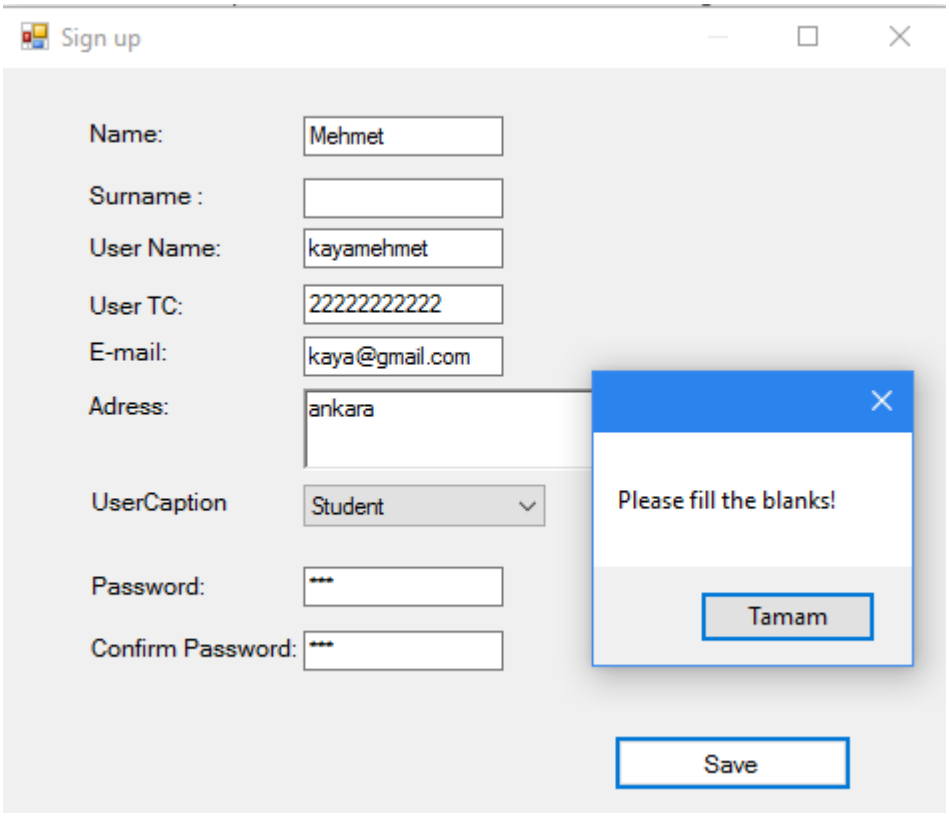


This use case has 5 warning window. 4 of them is error message, 1 of them is say user process is completed.

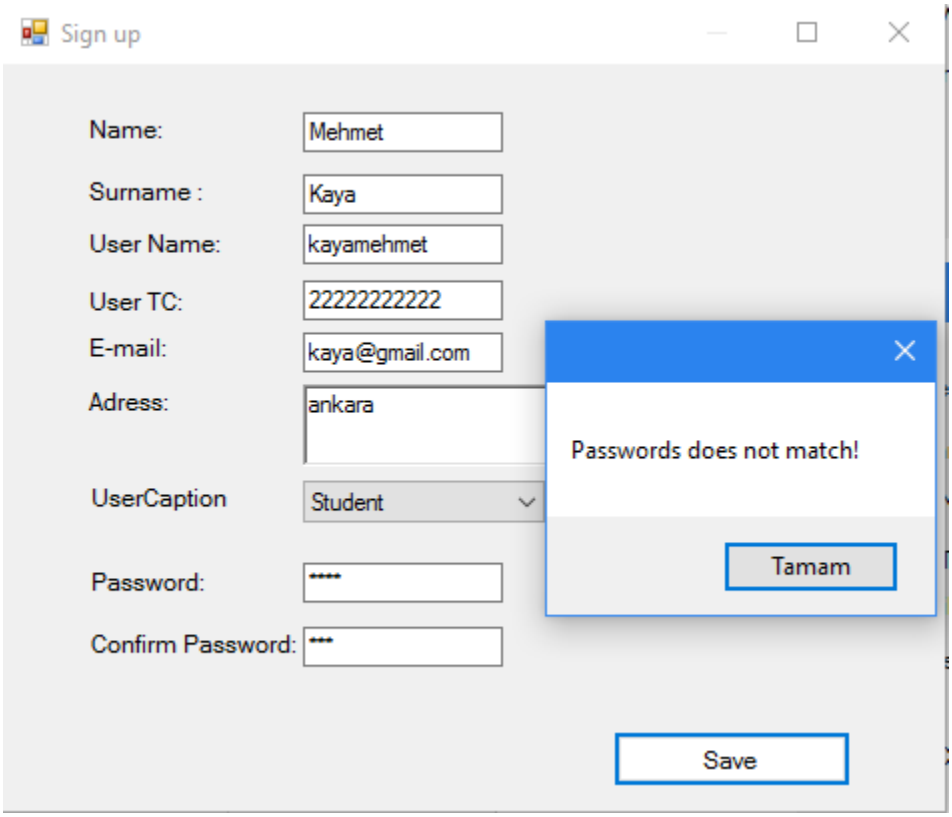
- If user enter an invalid e-mail adress:



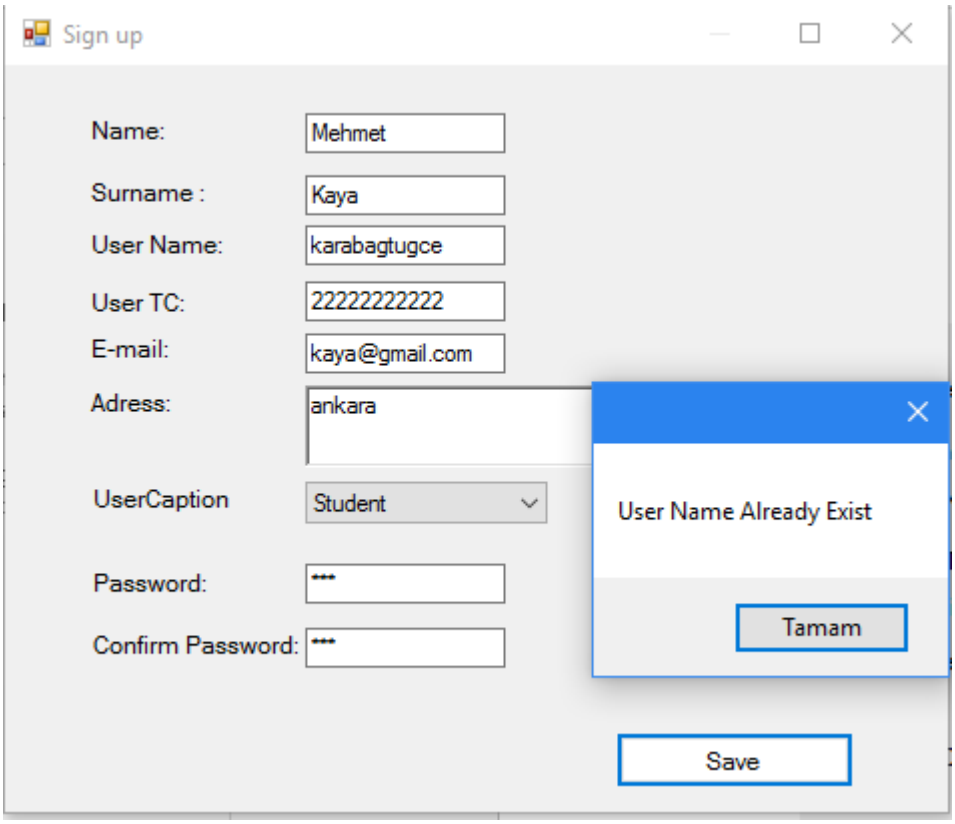
- If user leave unfilled textbox:



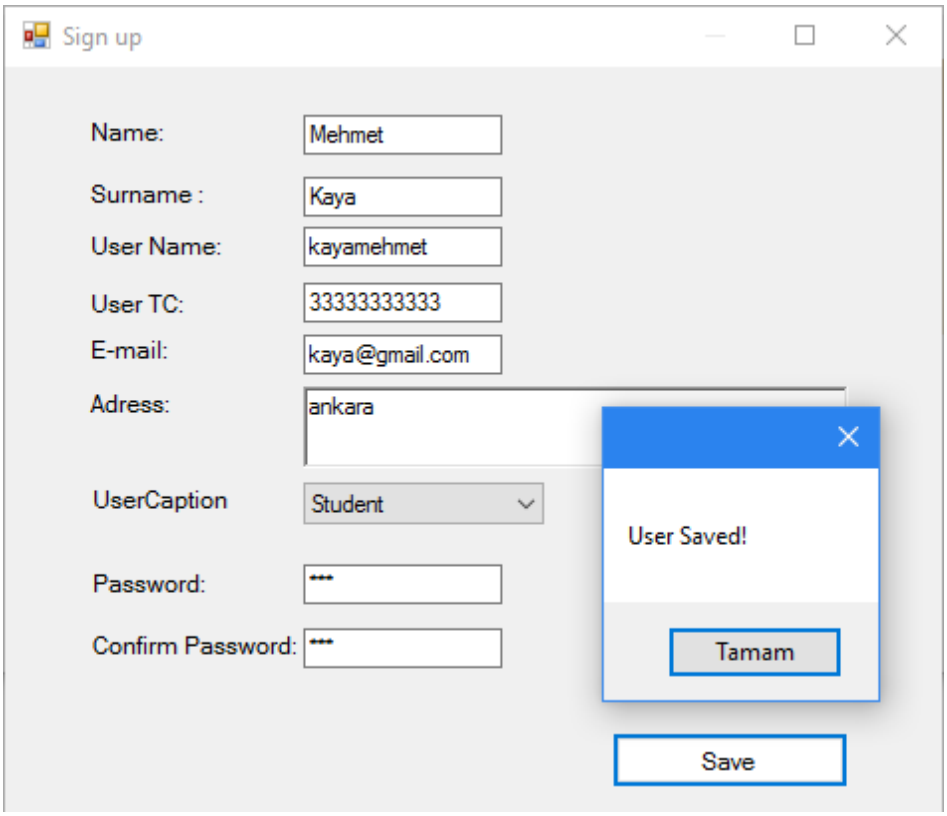
- If passwords doesn't match each other:



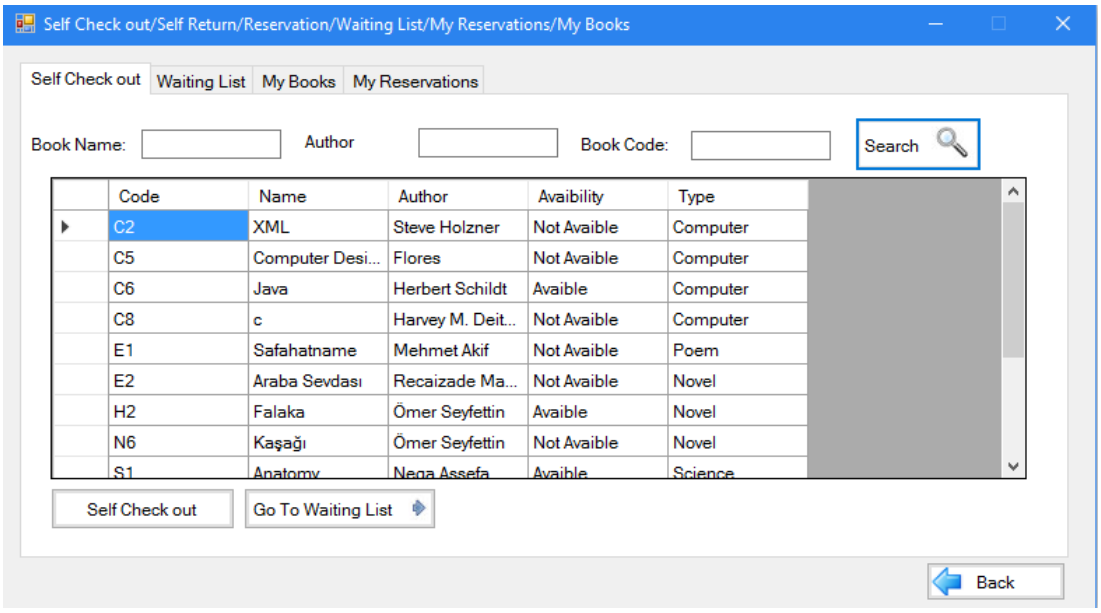
- If user enter an username already exist:



- If user sign up successfully:

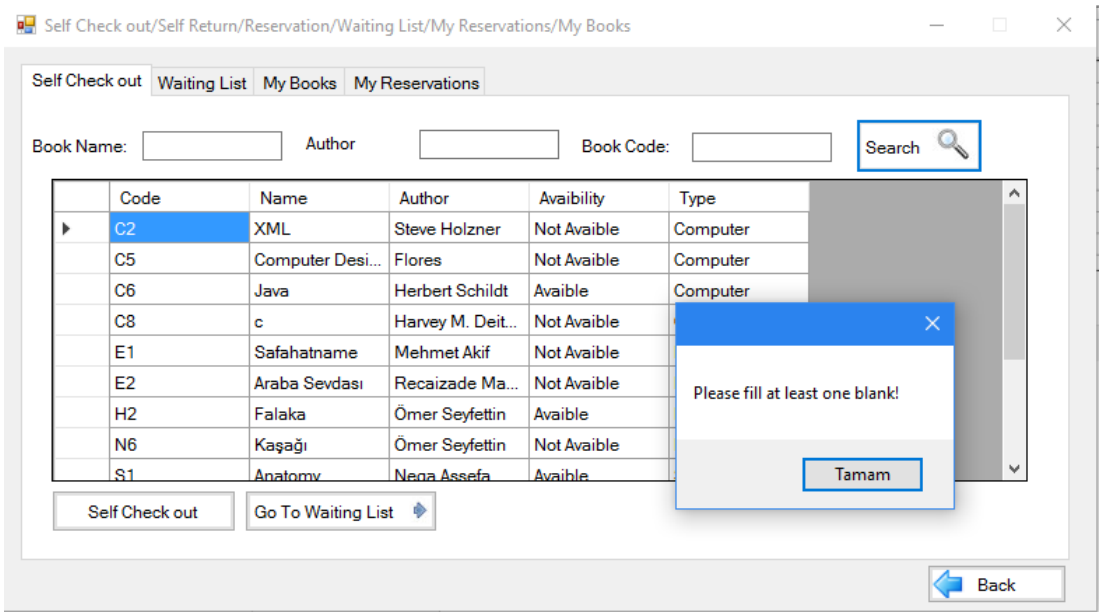


Search:

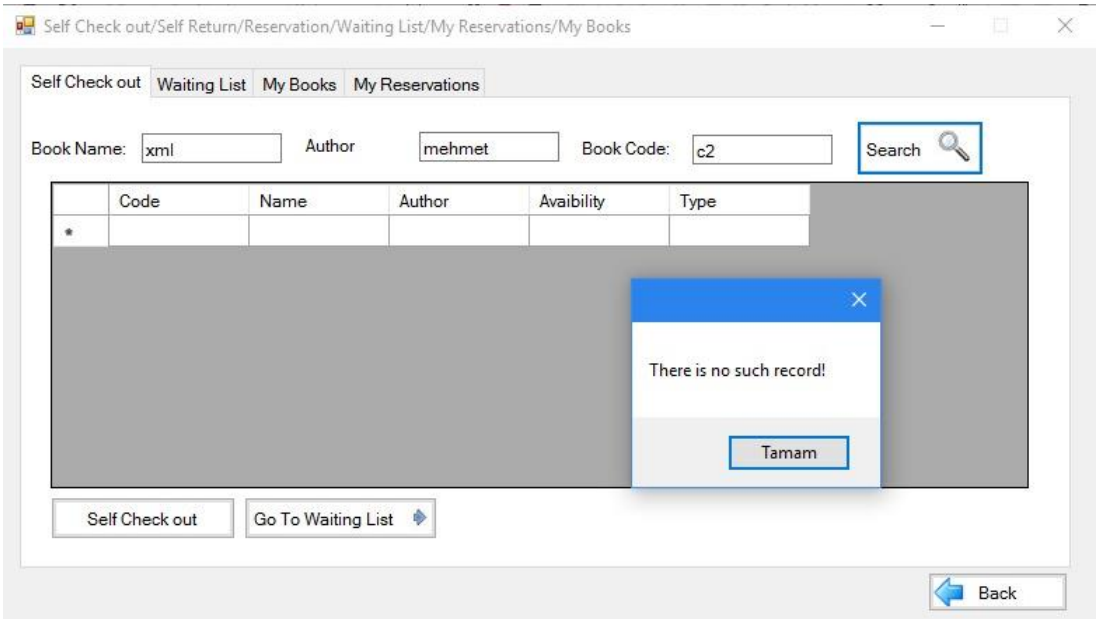


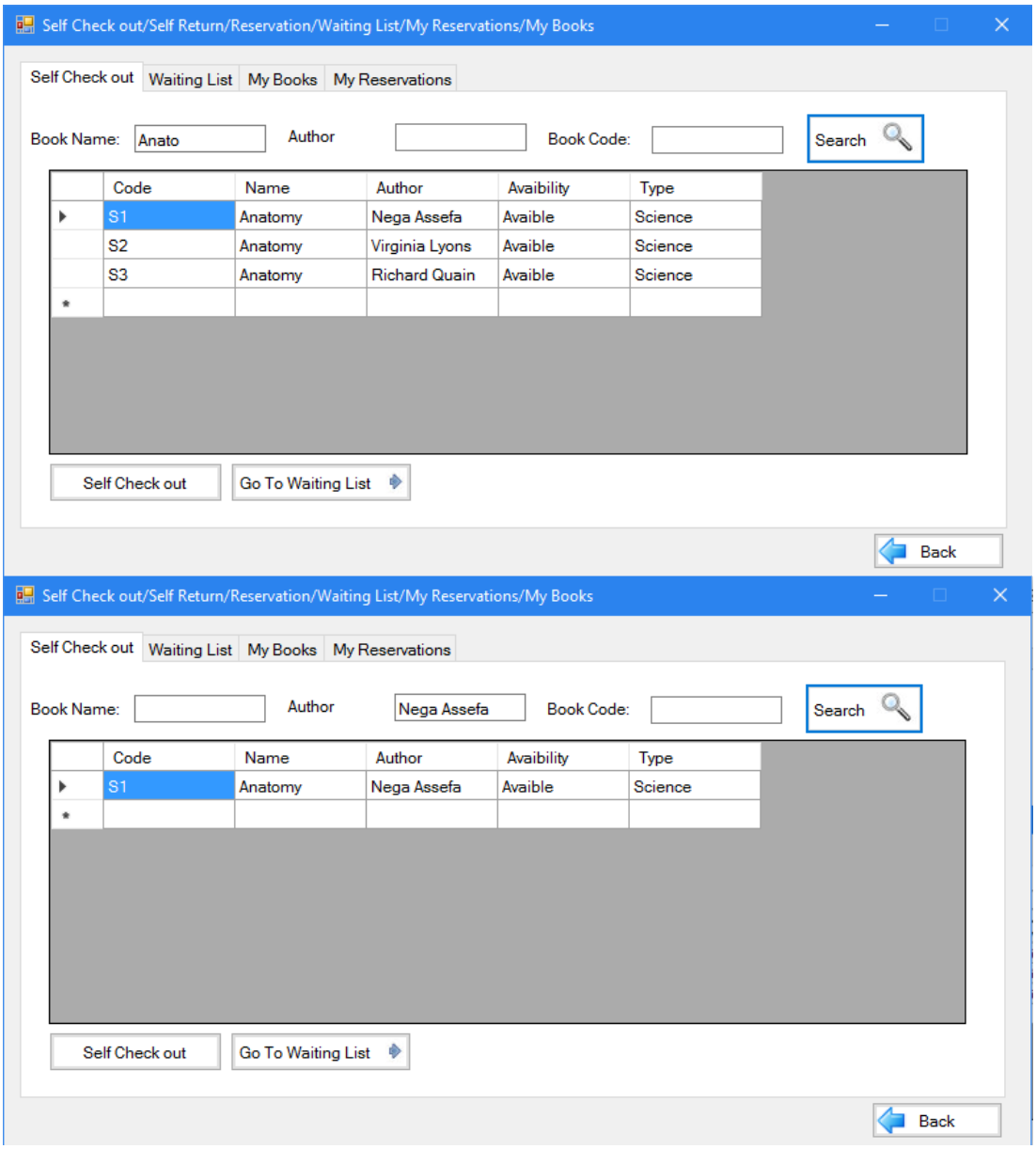
This is interface of searching book. We added icons search and back buttons. There are 2 warning windows for this use case.

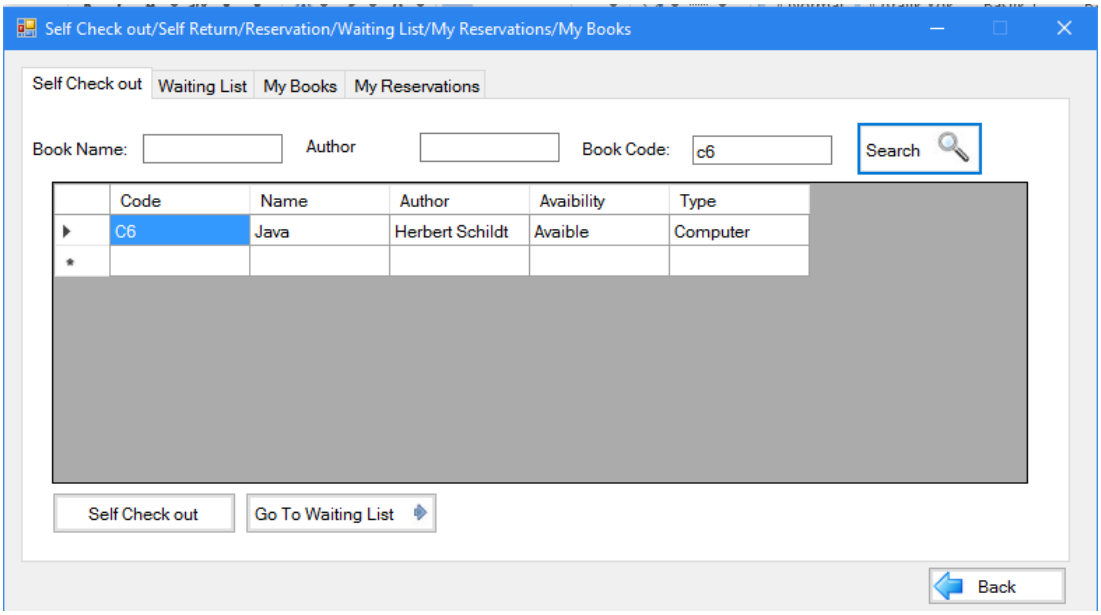
- If user does not enter any information to the textboxes:

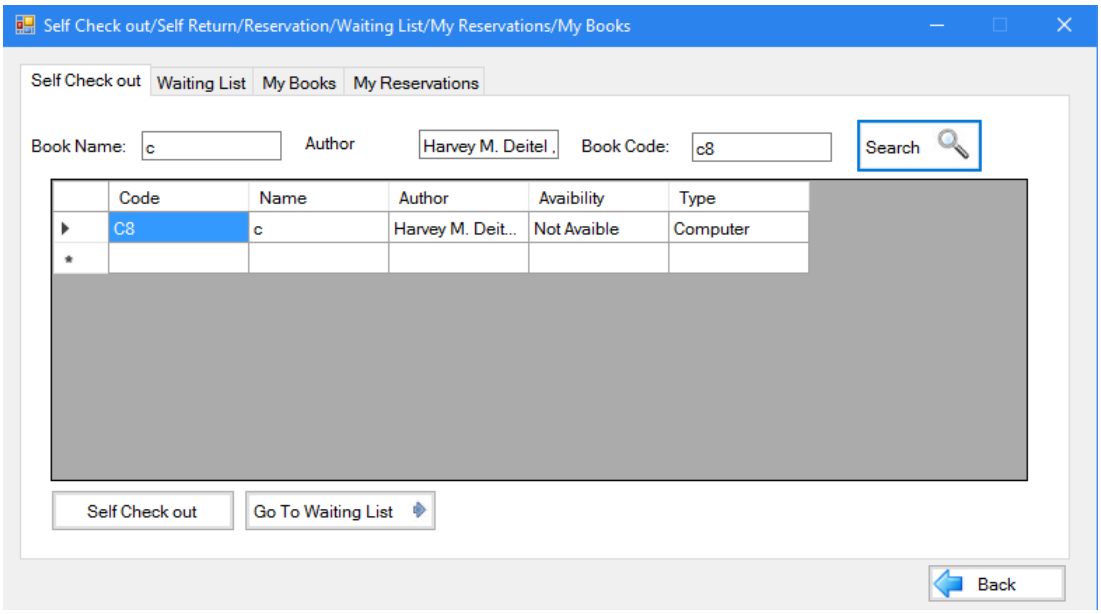


- If book name, author and book code do not match each other:

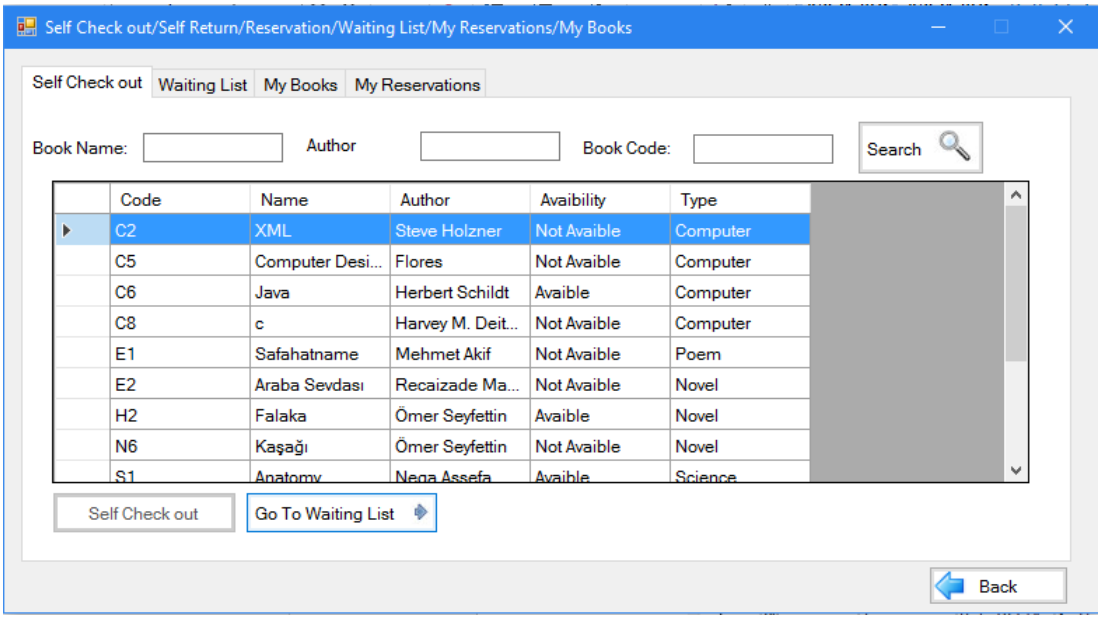



Self Check out/Self Return/Reservation/Waiting List/My Reservations/My Books
Self Check out
Waiting List
My Books
My Reservations
Book Name:
Anato
Author
Book Code:
Search
Code
Name
Author
Avaibility
Type
S1
Anatomy
Nega Assefa
Avaible
Science
S2
Anatomy
Virginia Lyons
Avaible
Science
S3
Anatomy
Richard Quain
Avaible
Science
Self Check out
Go To Waiting List
Back
Self Check out/Self Return/Reservation/Waiting List/My Reservations/My Books
Self Check out
Waiting List
My Books
My Reservations
Book Name:
Author
Nega Assefa
Book Code:
Search
Code
Name
Author
Avaibility
Type
S1
Anatomy
Nega Assefa
Avaible
Science
Self Check out
Go To Waiting List
Back





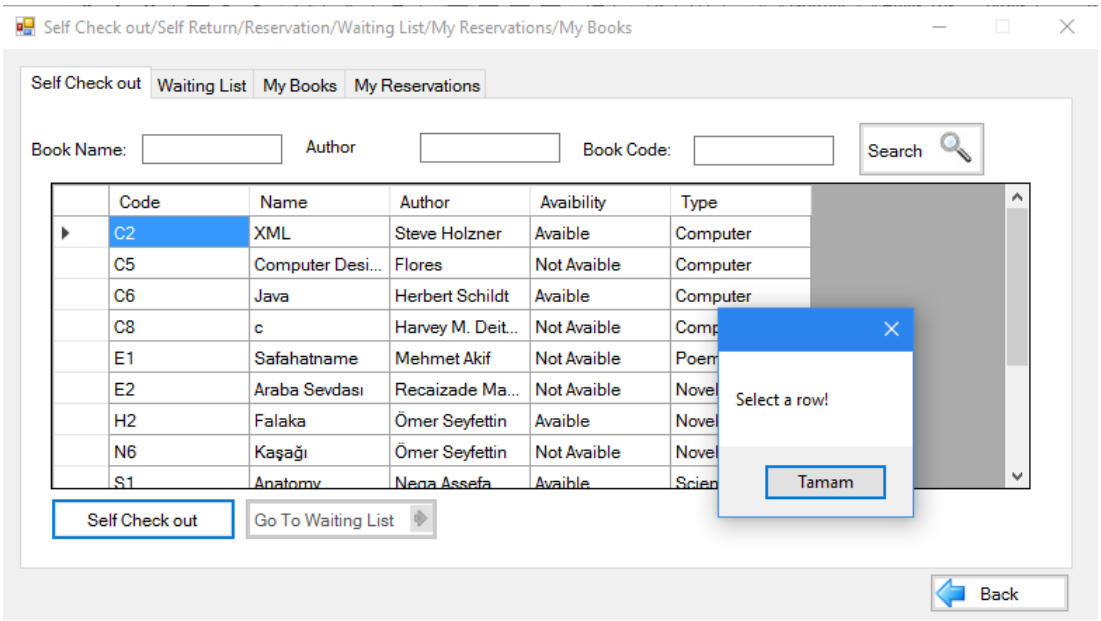
Self Check Out:



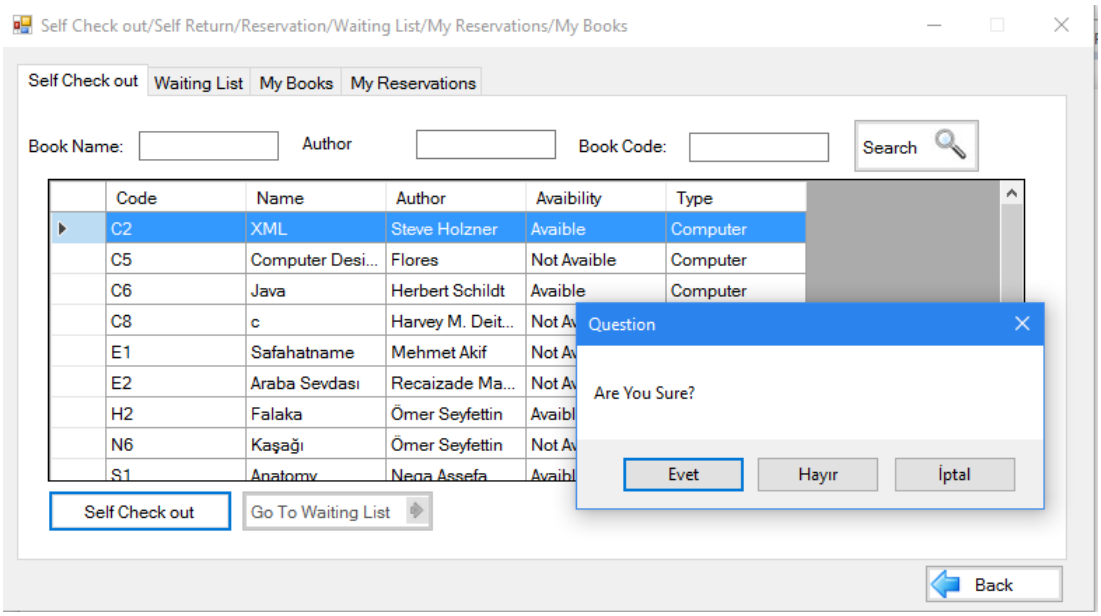
This interface is for self check out use case. You have to select the row of listed books. There are two options.

If book is aviable:

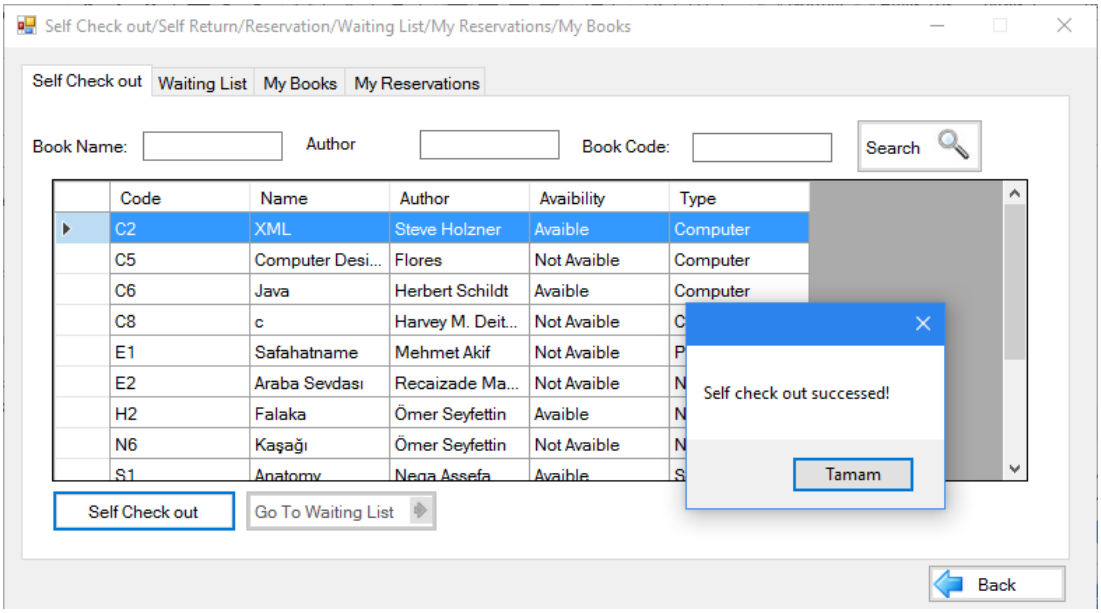
- If user does not select a row:



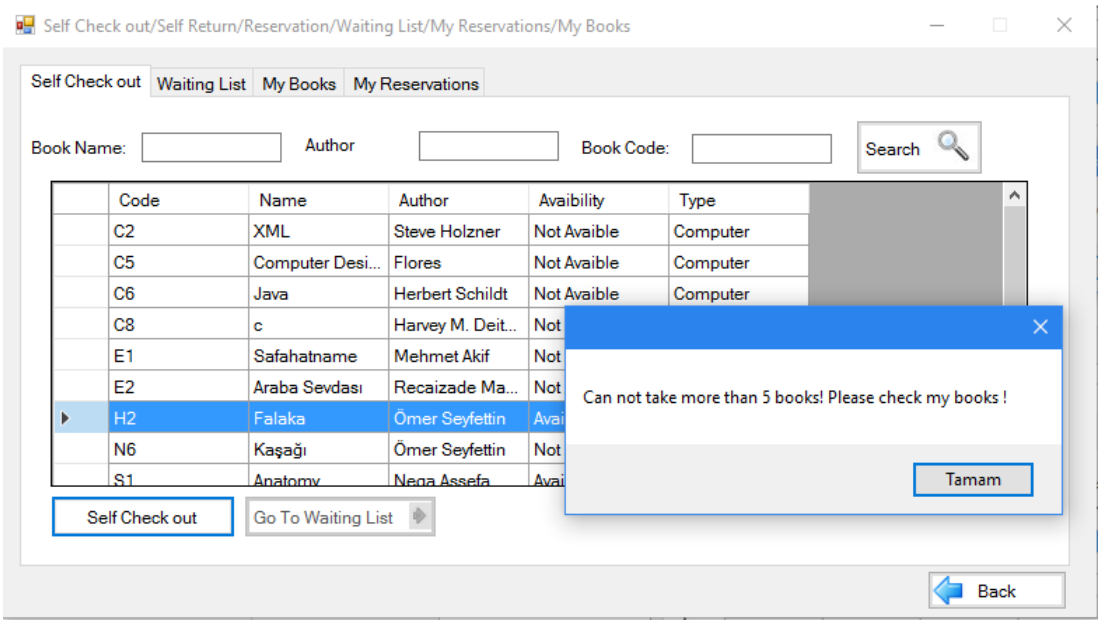
- The system create a question window.



- If self check out is successfully completed:

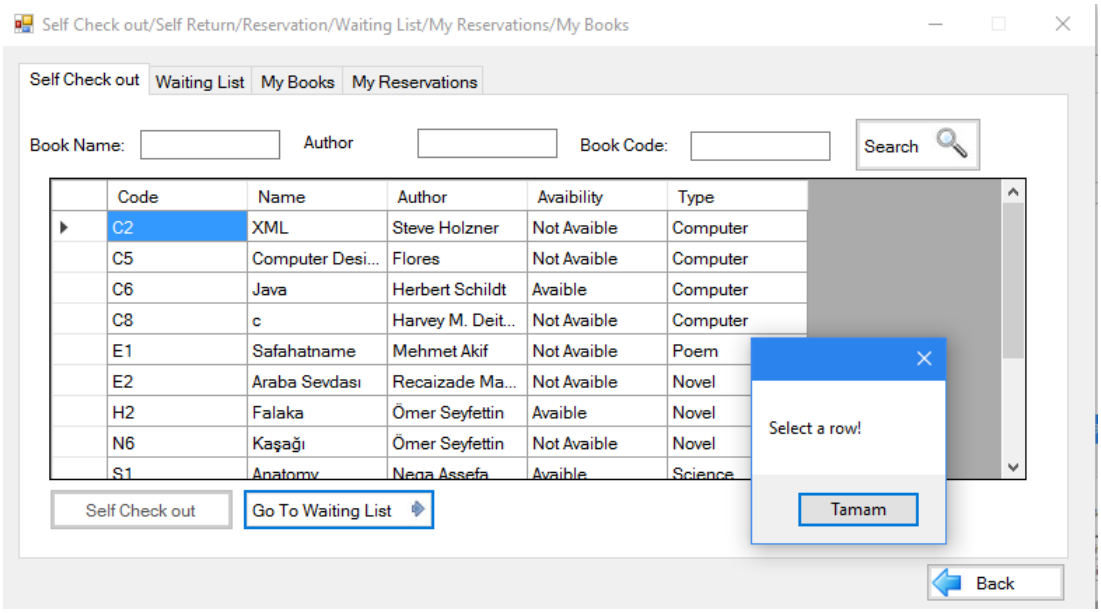


- If user already has 5 books and still try to self check out:

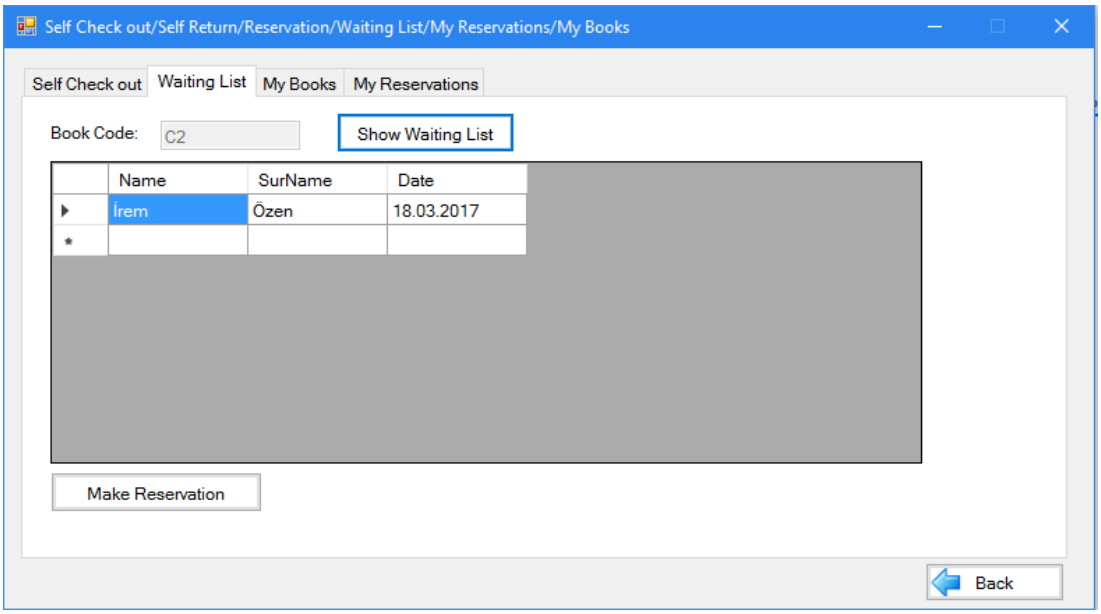


If book is aviable:

The system directs user to make reservation.



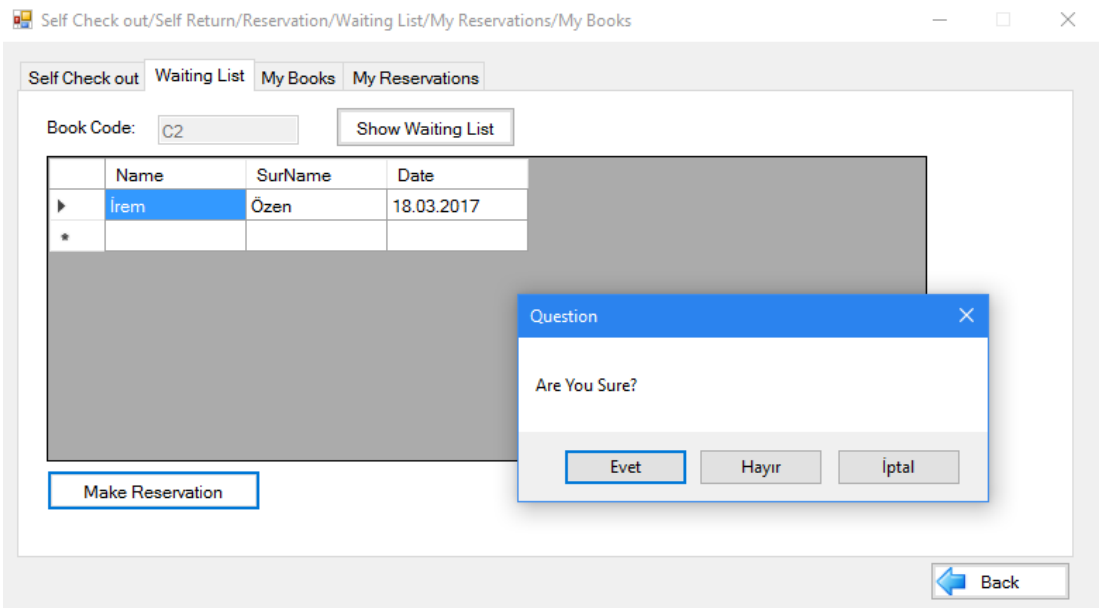
Waiting List:



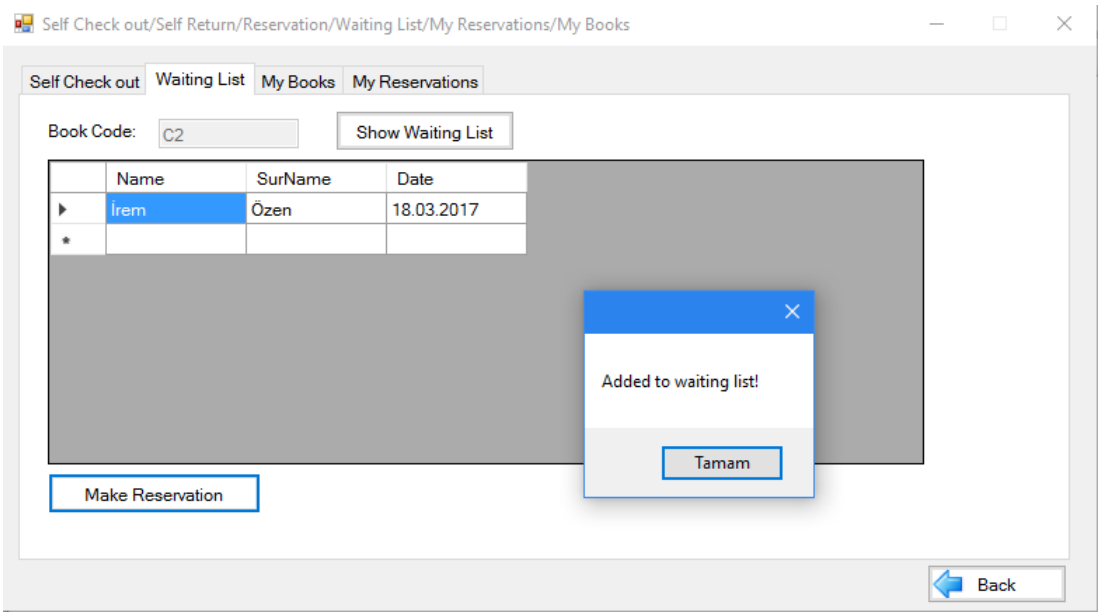
This is interface for waiting list use case. User can see waiting list of book according to the book code.

Make Reservation:

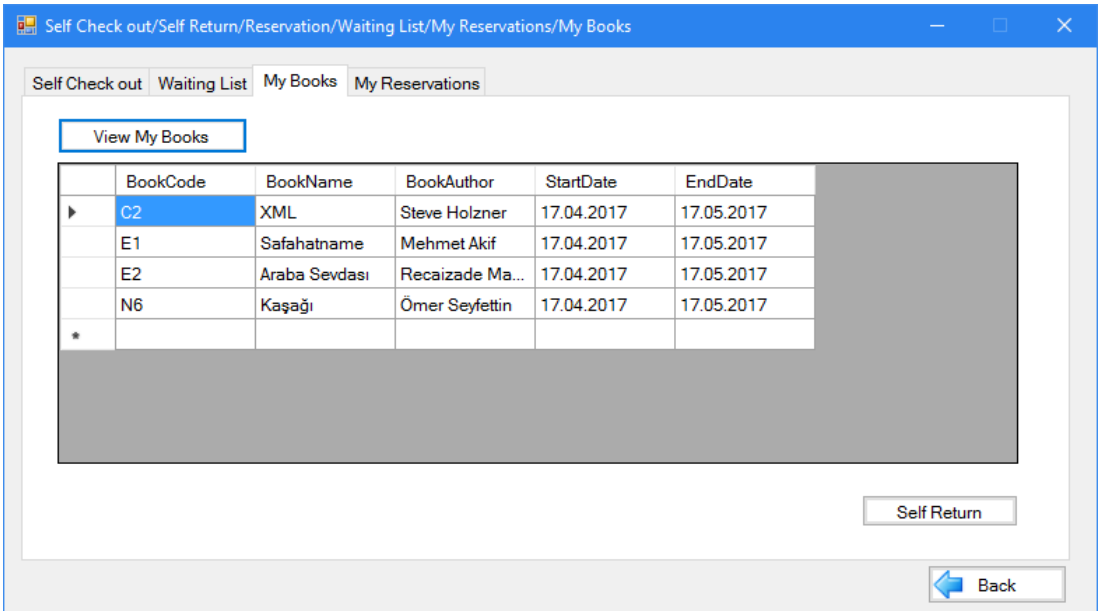
This is interface of Make Reservation use case. The system asks user if user is sure about process.



- If user make reservation successfully, user added to the waiting list of book:

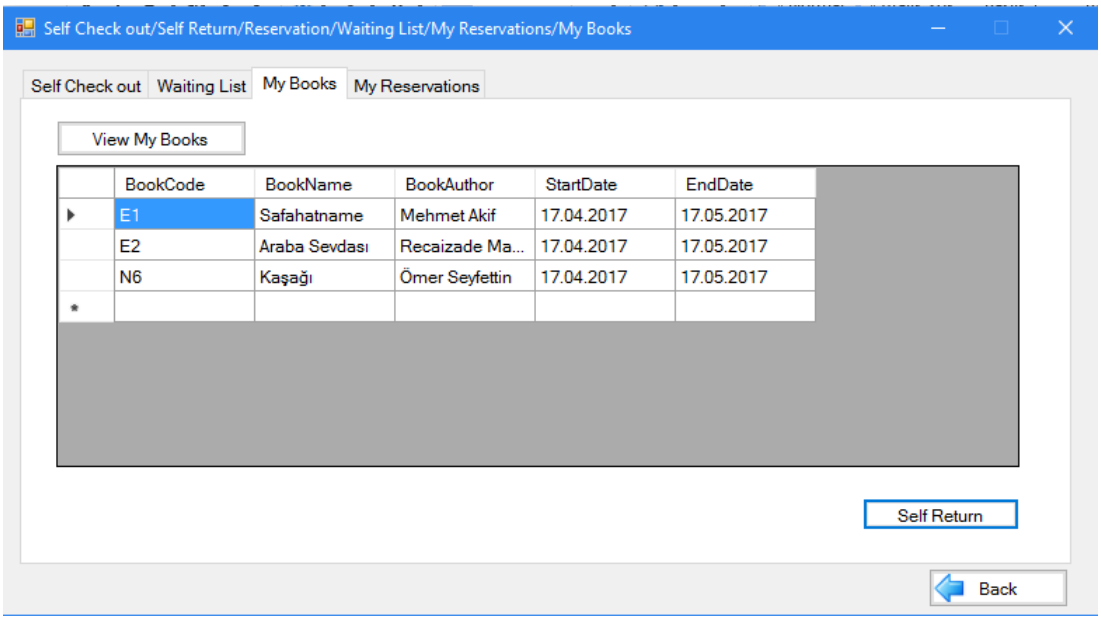


View My Books:

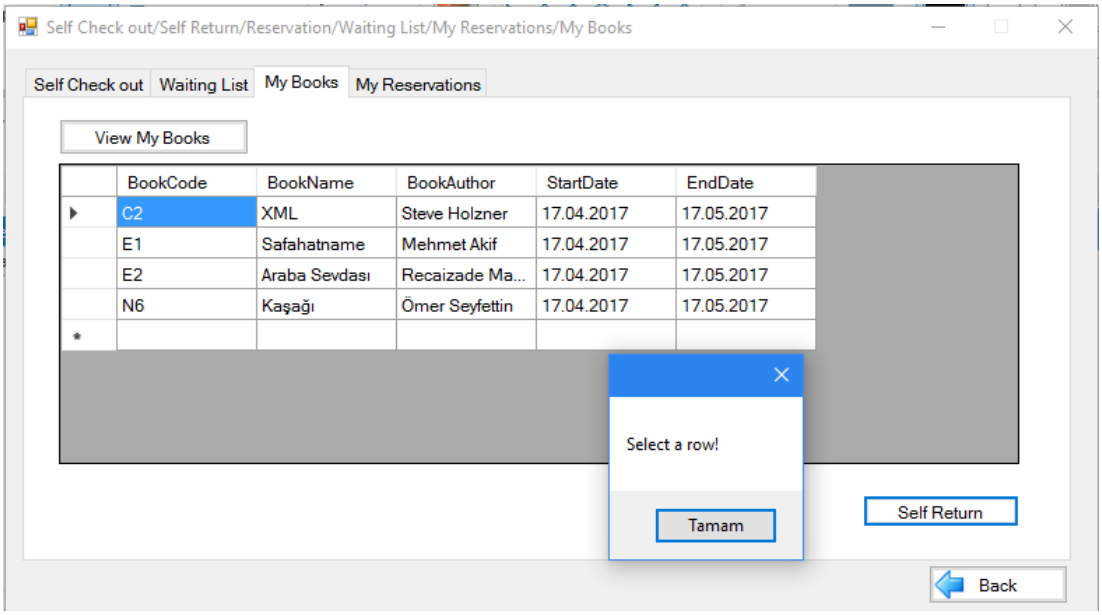


This is interface for view my books use case. User can see books by click to the View My Books button.

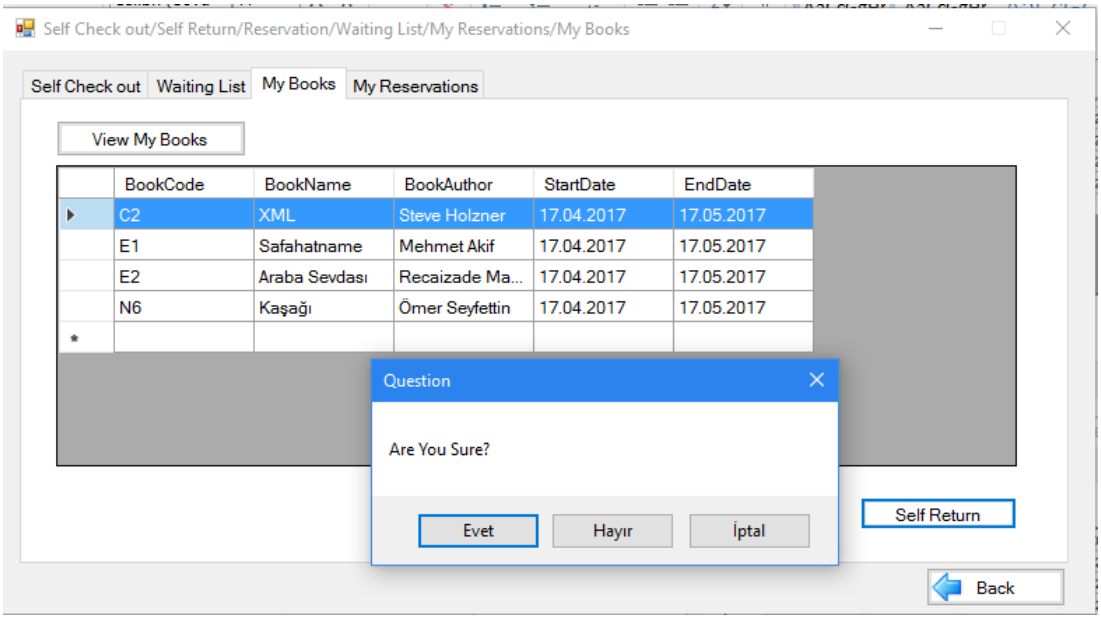
Self Return:



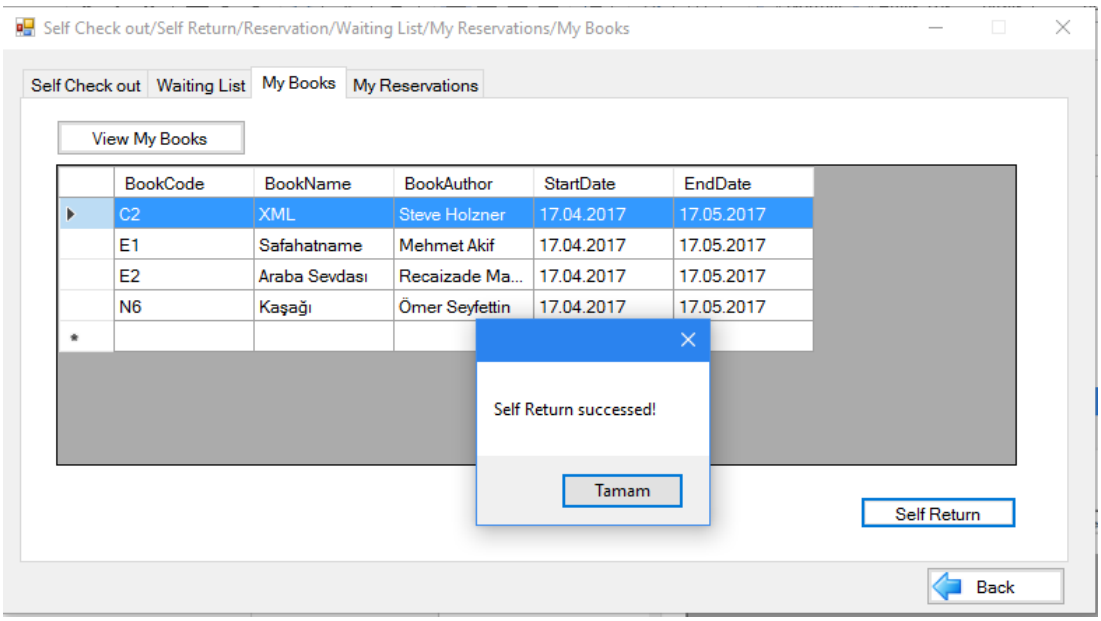
This is interface for self return use case. User can return book by click self return button after select the row of book.



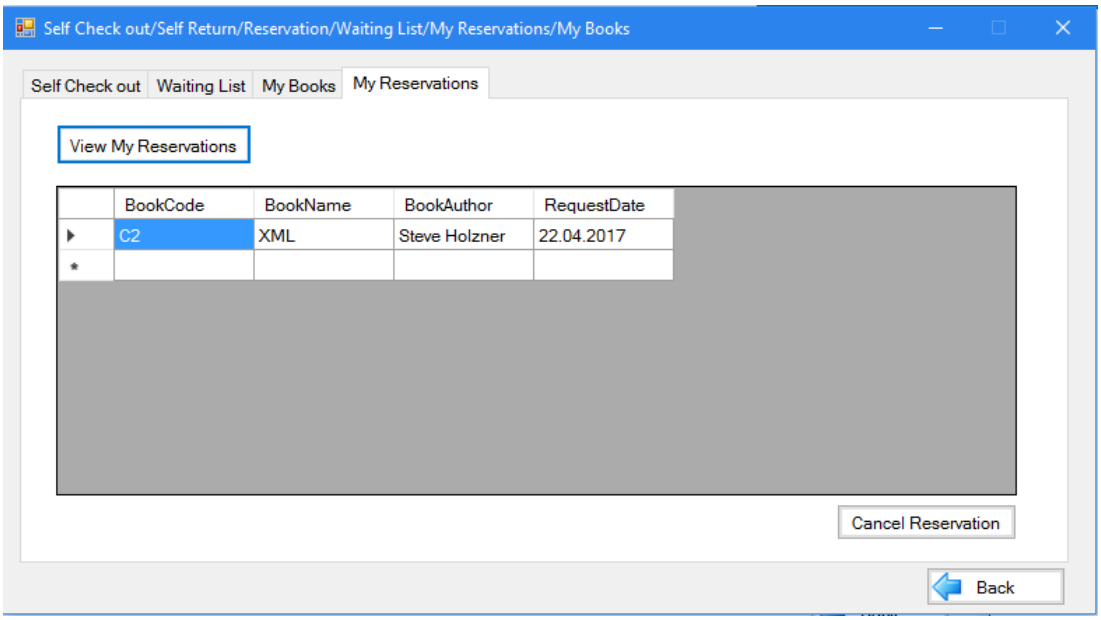
- The system create a question window with “Are You Sure?” message.



- If self return is successfully done:

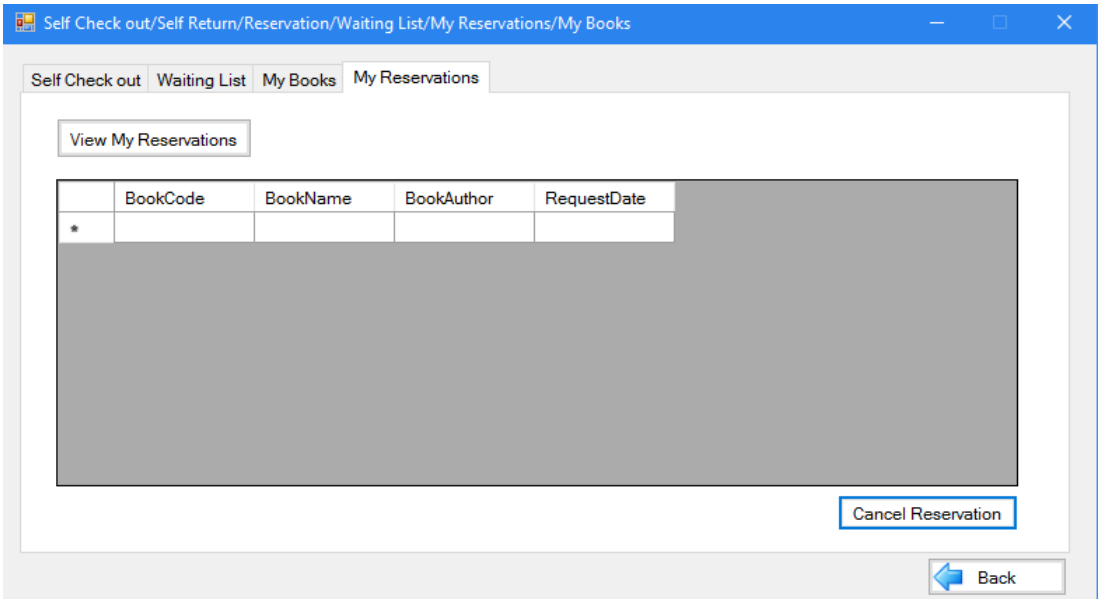


View My Reservation:

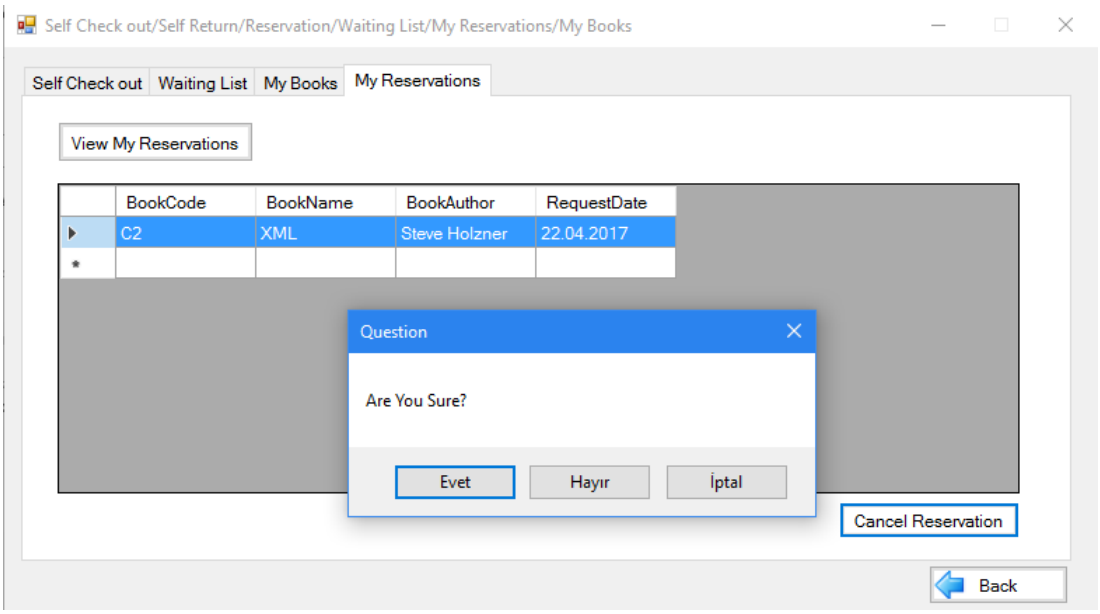


This interface is for view reservation use case. User see her/his reservation by cliking View My Reservation button.

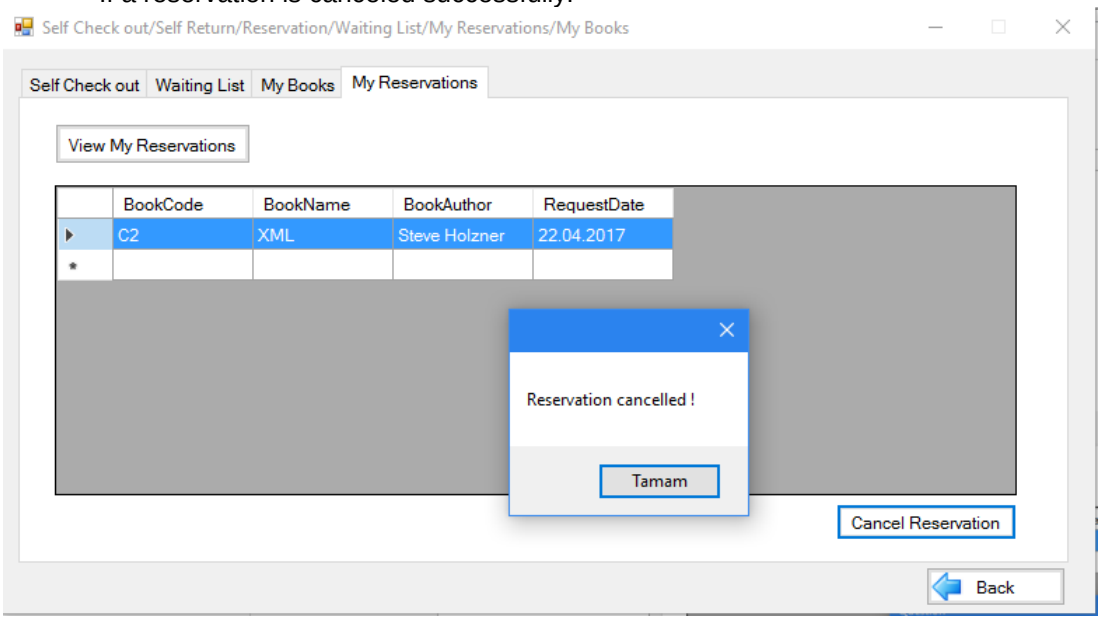
Cancel Reservation:



This interface is for cancel reservation use case. User select a reservation and click Cancel Reservation. For this interface, the system create a question window with “Are You Sure?” message.

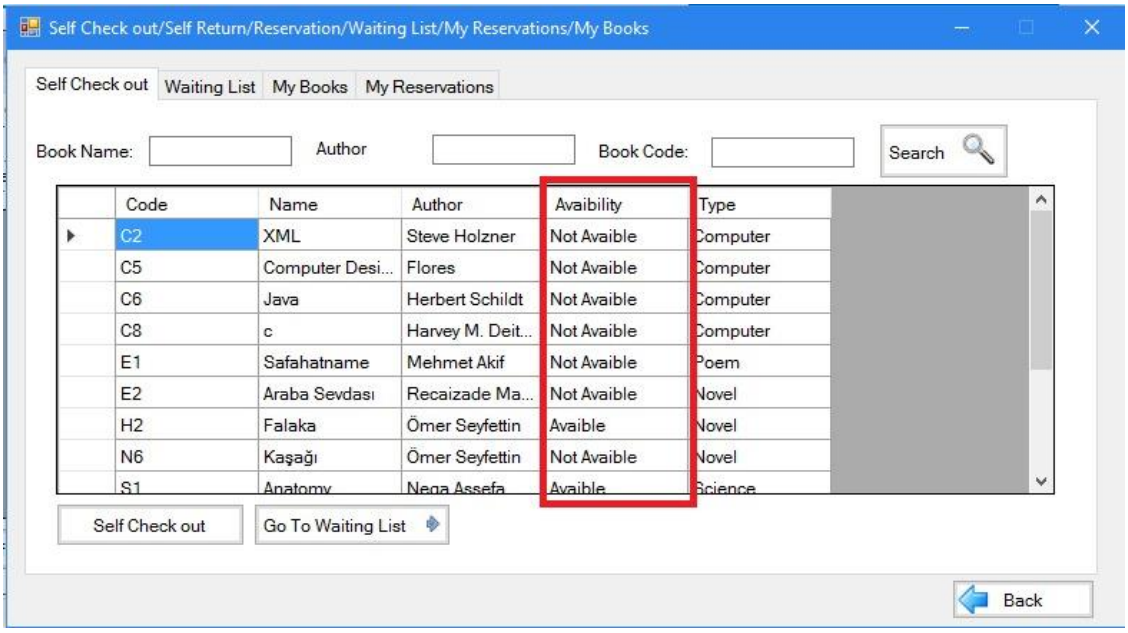


- If a reservation is canceled successfully:



Notify of Book Being Avaliable:

For this use case, there is not a specific interface. This use case done by the system, so user need to see book's status when user taking a book. Therefore, this use case is seen in Self Check Out interface.


Self Check out/Self Return/Reservation/Waiting List/My Reservations/My Books
Self Check out
Waiting List
My Books
My Reservations
Book Name:
Author
Book Code:
Search
Code
Name
Author
Avaibility
Type
C2
XML
Steve Holzner
Not Avaible
Computer
C5
Computer Desi...
Flores
Not Avaible
Computer
C6
Java
Herbert Schildt
Not Avaible
Computer
C8
c
Harvey M. Deit...
Not Avaible
Computer
E1
Safahatname
Mehmet Akif
Not Avaible
Poem
E2
Araba Sevdası
Recaizade Ma...
Not Avaible
Novel
H2
Falaka
Ömer Seyfettin
Avaible
Novel
N6
Kaşağı
Ömer Seyfettin
Not Avaible
Novel
S1
Anatomy
Nega Assefa
Avaible
Science
Self Check out
Go To Waiting List
Back