HACETTEPE UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING
BBM 487:SOFTWARE ENGINEERING LABORATORY

Group Number:5

21228544 :Şehrinaz Koca
21328103 :Burcu İskender
21328499 :Sergen Topçu

<Project Name>
Software Design Description

1. Revision History

Version	*Date*	*Author*	*Change Description*
1.0	19.04.2017	Şehrinaz Koca	Some parts of the er diagram have changed.
1.1	22.04.2017	Burcu İskender	Some parts of the interfaces have changed.
1.2	24.04.2017	Sergen Topçu	Some parts of the design have changed.

2. INTRODUCTION

We are doing the library system in this project. In this part of the project we have made half of the use cases of the progeny. These are login, logout, view books, self-return, self-checkout, add waiting list and search books use cases.

2.1 Purpose and Scope

In this part of the project, half of the use cases were used. In this way, even if the project is incomplete, we could create the interface of the system that we will create. We designed our system using php and html. Since we design the system as web-based, it is necessary for the user connect internet in order to enter the system. We had to change some of the places we designed. You can see them in the bottom. When we wrote the code, we noticed some things and found them in the other file.

2.2 Document Overview

Design Constraints and Decisions: This section discusses design constraints and the circumstances in which the system can be used.

Design Details: This section provides information about the design.

Software Components: There is a class diagram for LBLS.

Software Behavior: There is a sequence diagram for each use case in this section.

Data Model (E-R Diagram): This section contains the updated version of the system er diagram.

User Interface Design: This section contains the interfaces of the use cases of the system.

Requirements Traceability: We have drawn the requirement / class matrix in this section.

2.3 System Overview

Our system is a system that supports the web base. Our main system is the library system. In addition, there is a payment system and a bar code system. The barcode system is considered as a side system and it is only shown as a design and there is no coding. You can access this system from any computer connected to the internet via the browser or from the phone.

2.4 Definitions, Acronyms, and Abbreviations

Term/Acronym	***Definition***
MVC	Model View Controller
LBLS	Library Book Loan System

2.5 References

1- https://www.w3schools.com/php/php_mysql_connect.asp
2- https://www.ibm.com/developerworks/rational/library/3101.html
3- https://www.tutorialspoint.com/uml/uml_class_diagram.htm

3. Design Constraints and Decisions

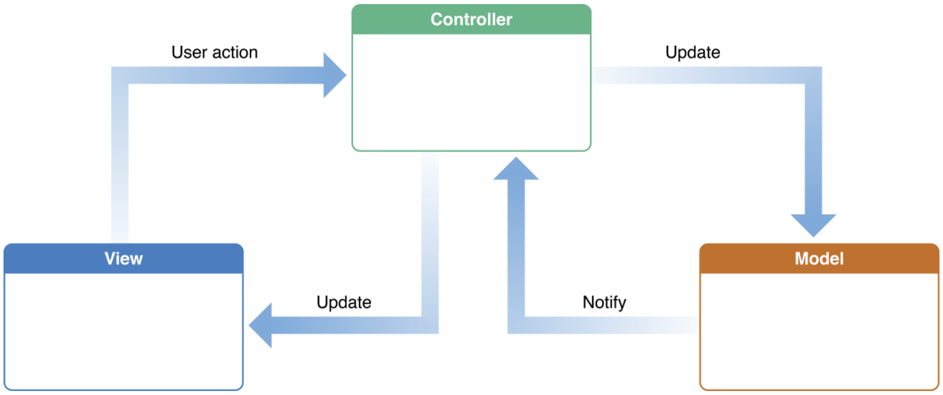
First, this system can be used on any platform where the web browser is. Both user and admin can log in to the system. Admin is the person who manages the system. In all parts except for searching books, the user needs to login to the system. When entered into the system, there is a part where we can search for books and a part where we can login. We used the mvc model for this system. In addition, we used php and html languages. We think our system is safe. Admin is the person who controls the system. We think that the infrastructure of the system is robust and is open to future development.

- A user must return a book within 30 days.
- The user must log in for transactions outside search.
- Admin can update user and book information.
- All of people can search book.
- User can reserved books in system.
- User can use barcode system for borrowing operations.
- User and admin can logged in system.
- User can pay money for late fine days.
- Anyone can use this system with web browser.
- It is a system where the user is an admin and books.

4. Design Details

We designed our system according to the mvc model. There are 3 parts according to mvc model: 1-Model,2- view, 3-controller.

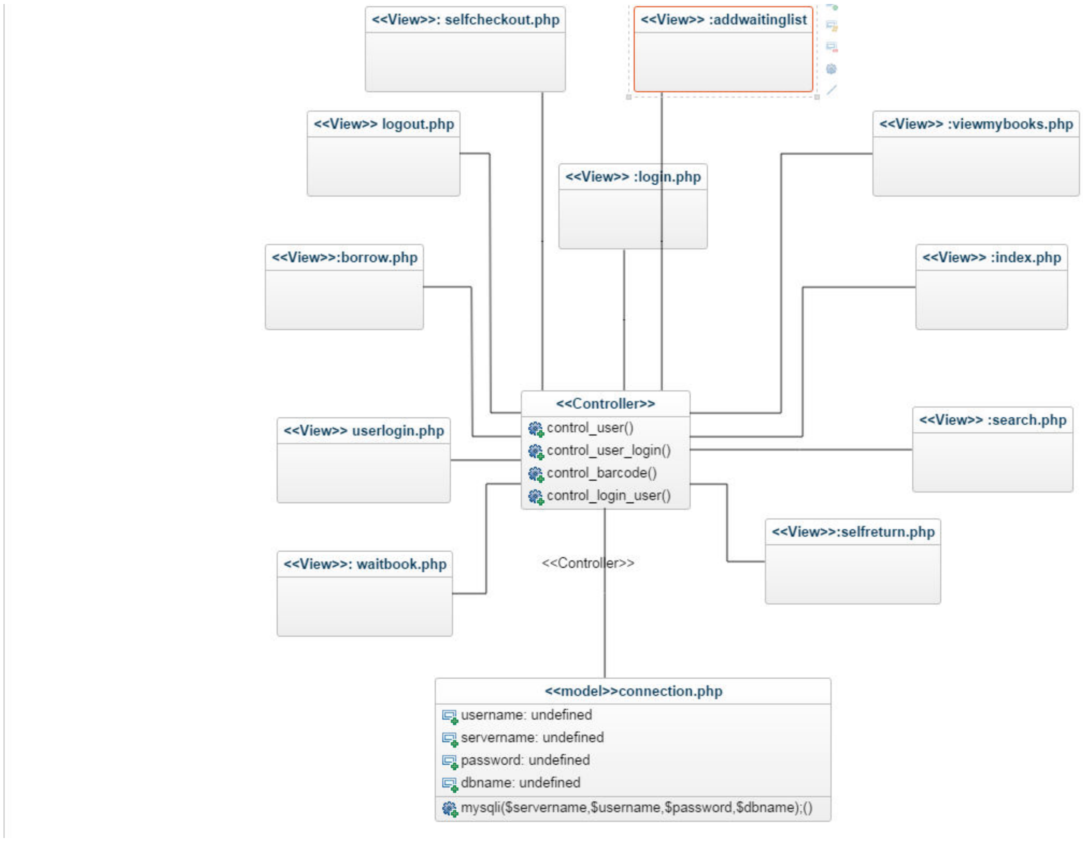
The Model-View-Controller (MVC) design pattern assigns objects in an application one of three roles: model, view, or controller. The pattern defines not only the roles objects play in the application, it defines the way objects communicate with each other. Each of the three types of objects is separated from the others by abstract boundaries and communicates with objects of the other types across those boundaries.



4.1 Software Components

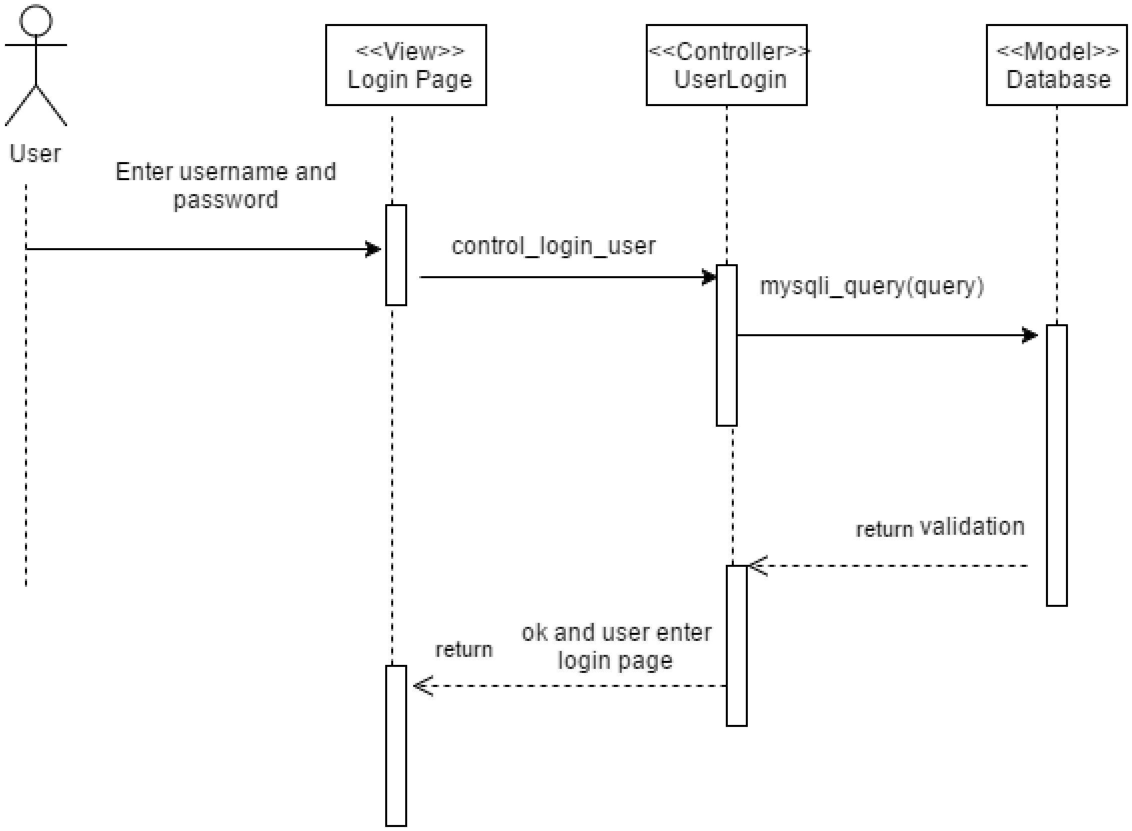
Since we use php, we did not use class to do this project.
So we tried to make this part according to the design of the mvc model. We saw the php extensions we used as class.

The class diagram design is as follows:

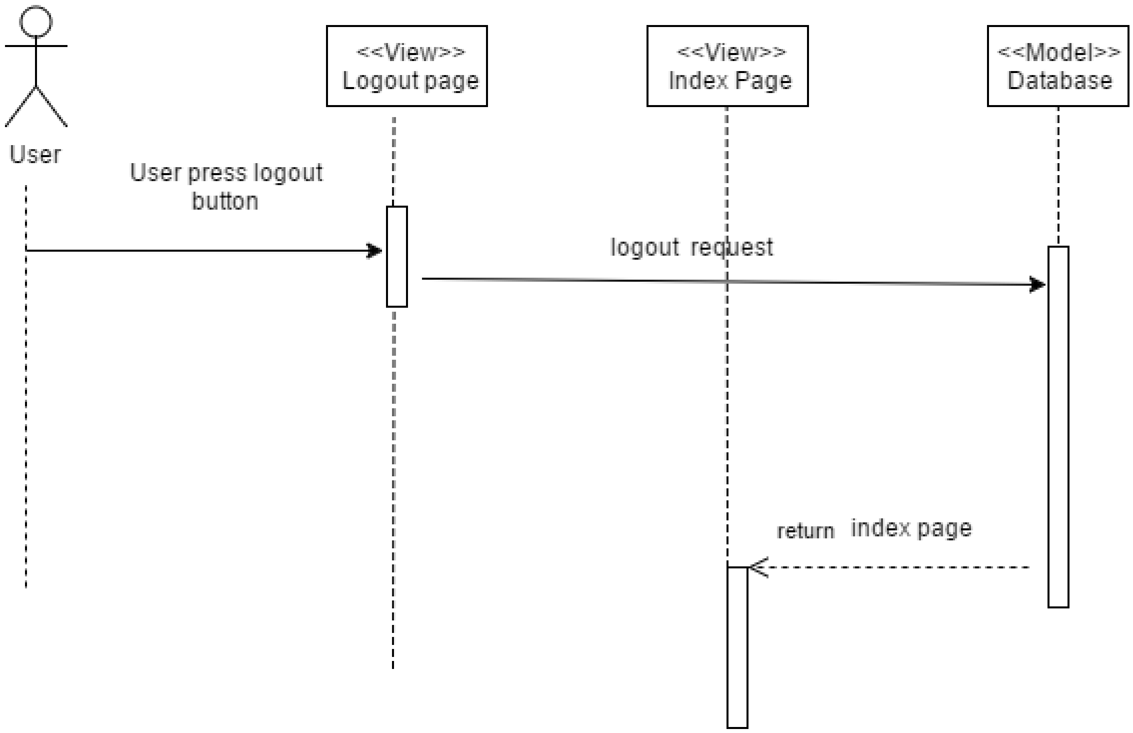


4.2 Software Behavior

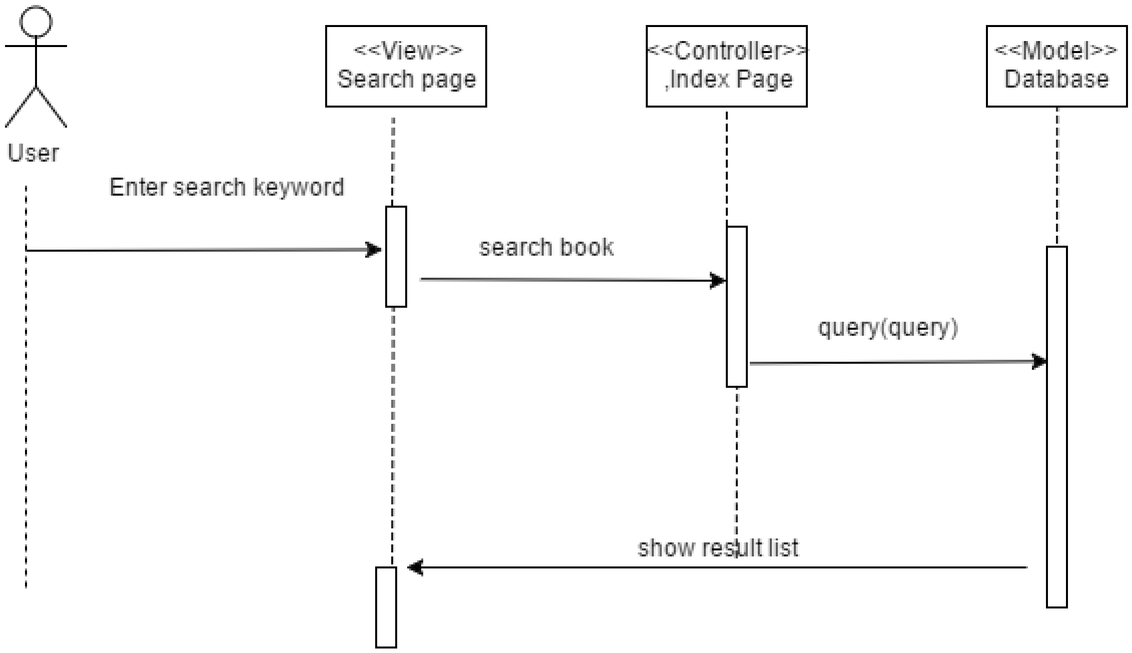
Login Sequential Diagram

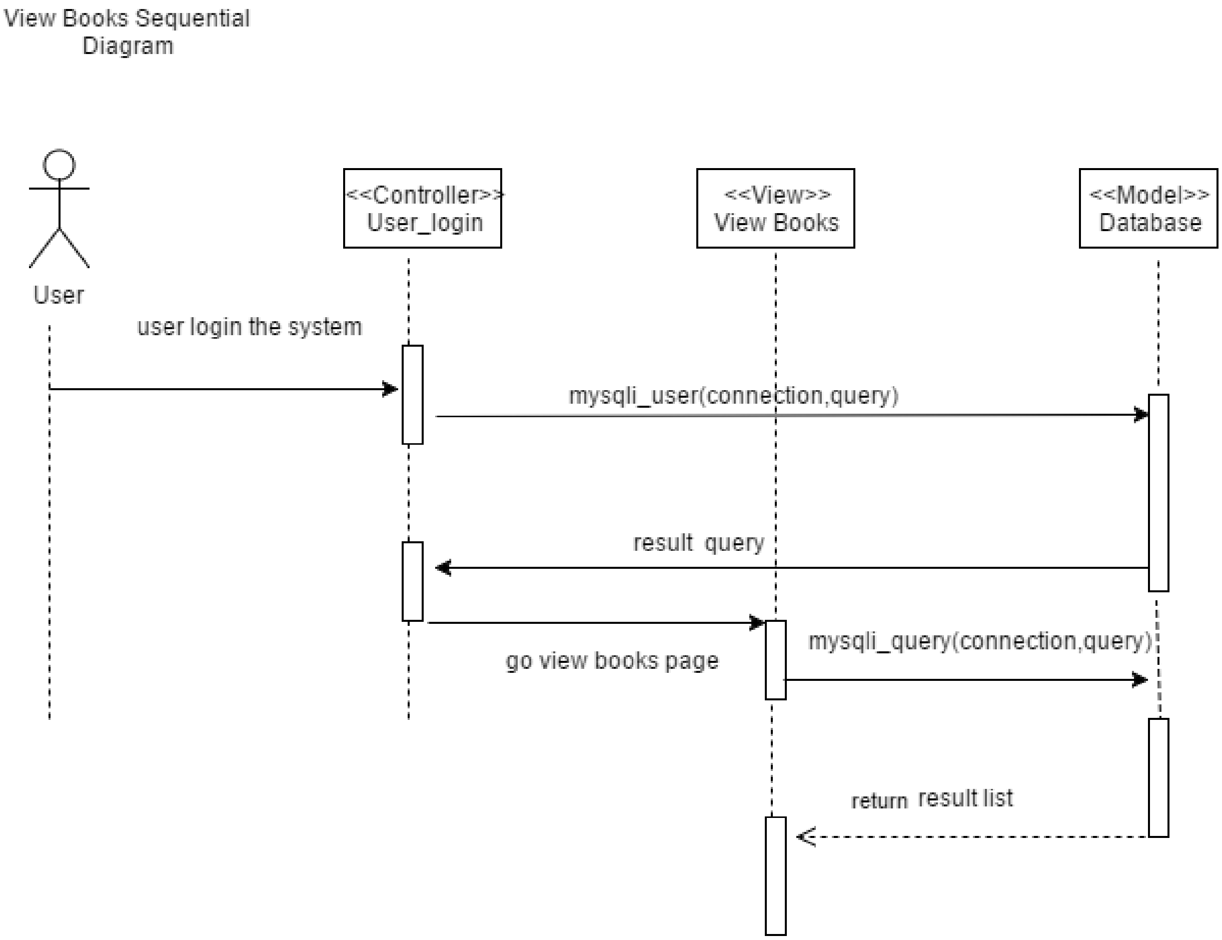


Logout Sequential Diagram

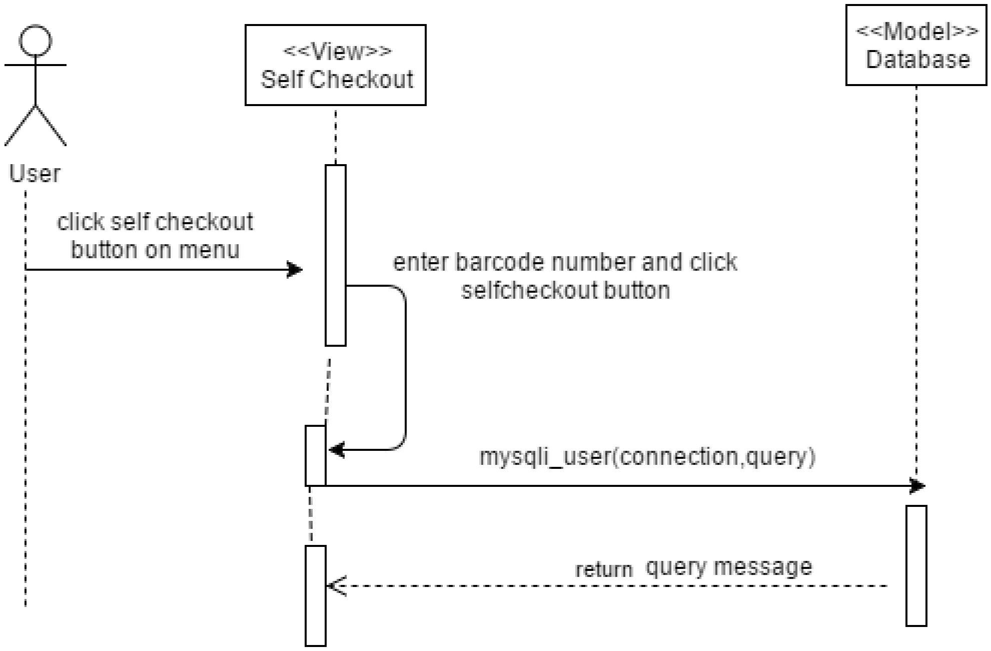


Search Books Sequential Diagram

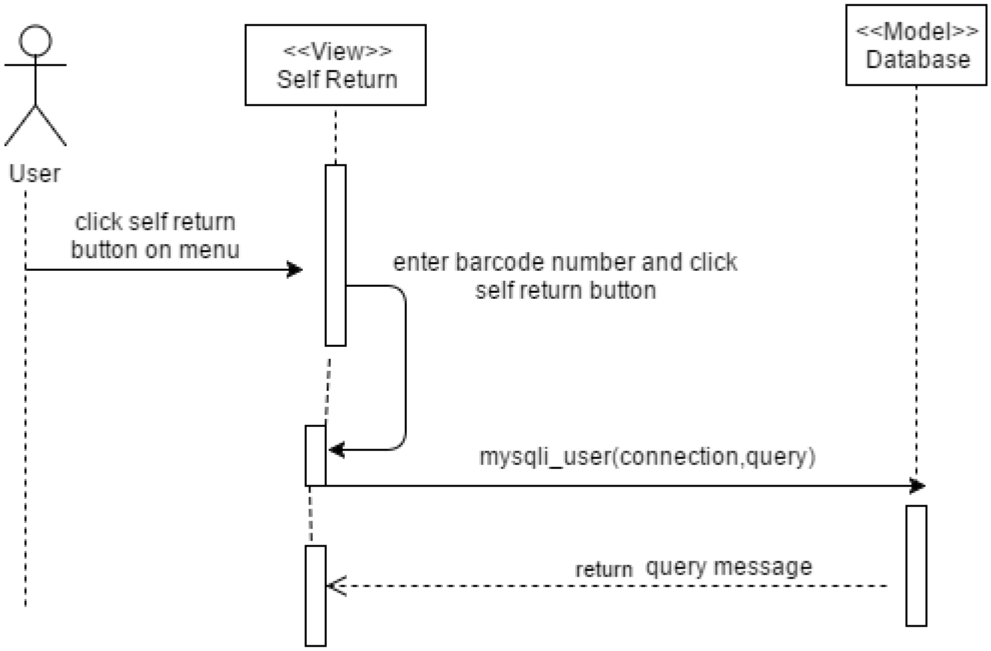



View Books Sequential Diagram
User
<<Controller>> User_login
<<View>> View Books
<<Model>> Database
user login the system
mysqli_user(connection,query)
result query
go view books page
mysqli_query(connection,query)
return result list

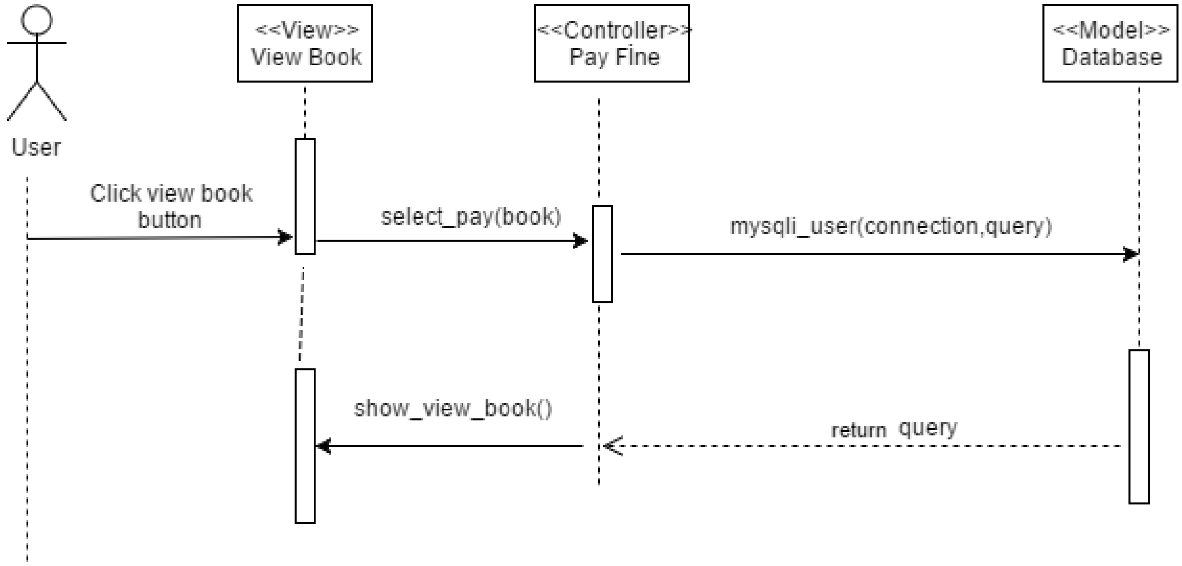
Self Checkout Sequential Diagram



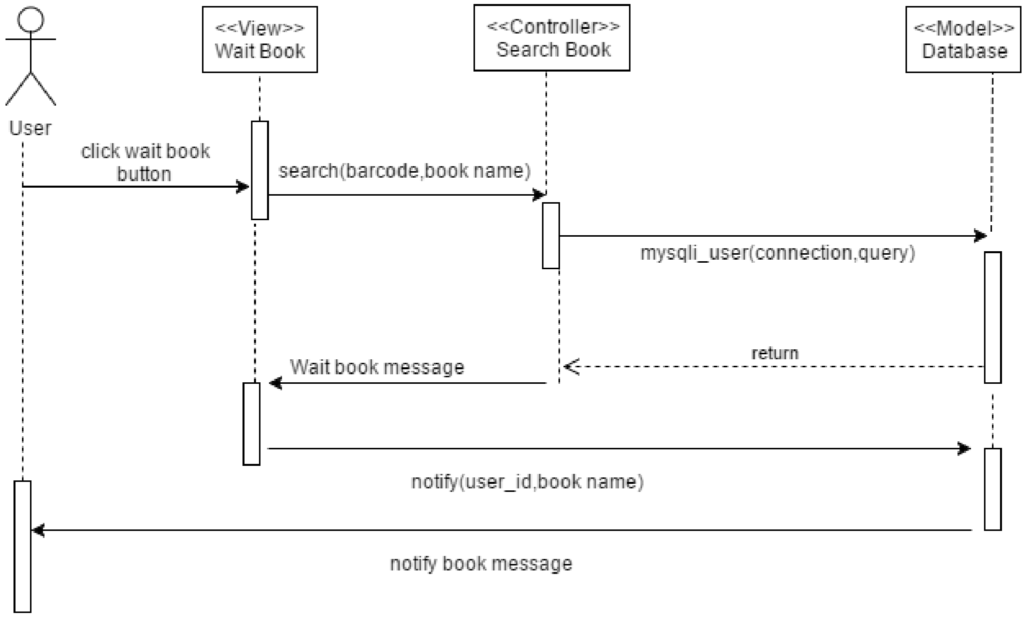
Self Return Sequential Diagram



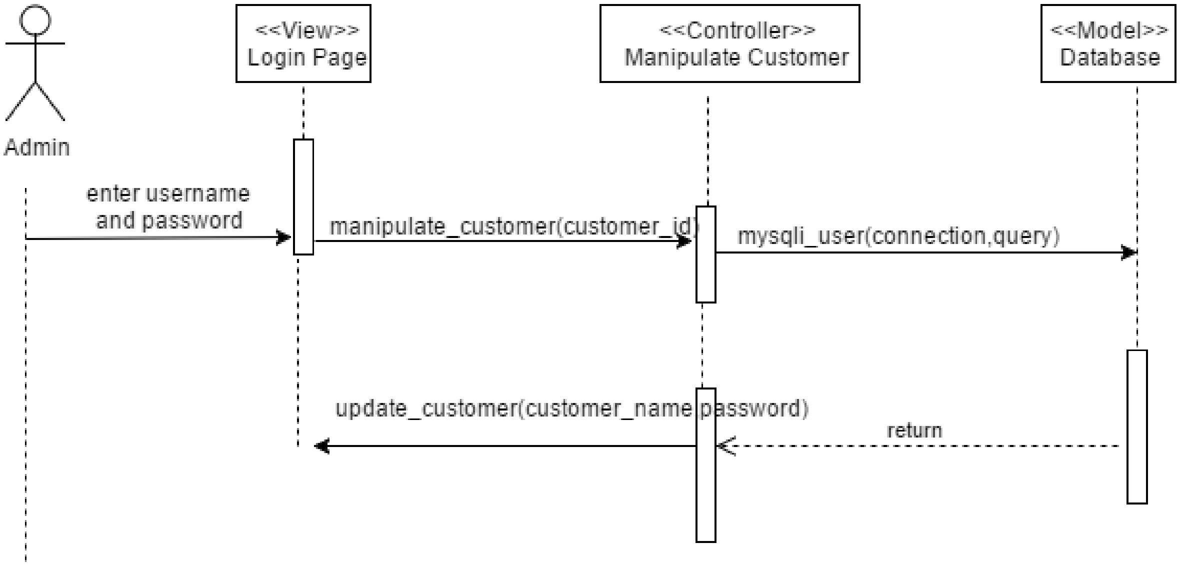
Pay Fine Sequential Diagram



Notify of Book being avaliable
Sequential Diagram

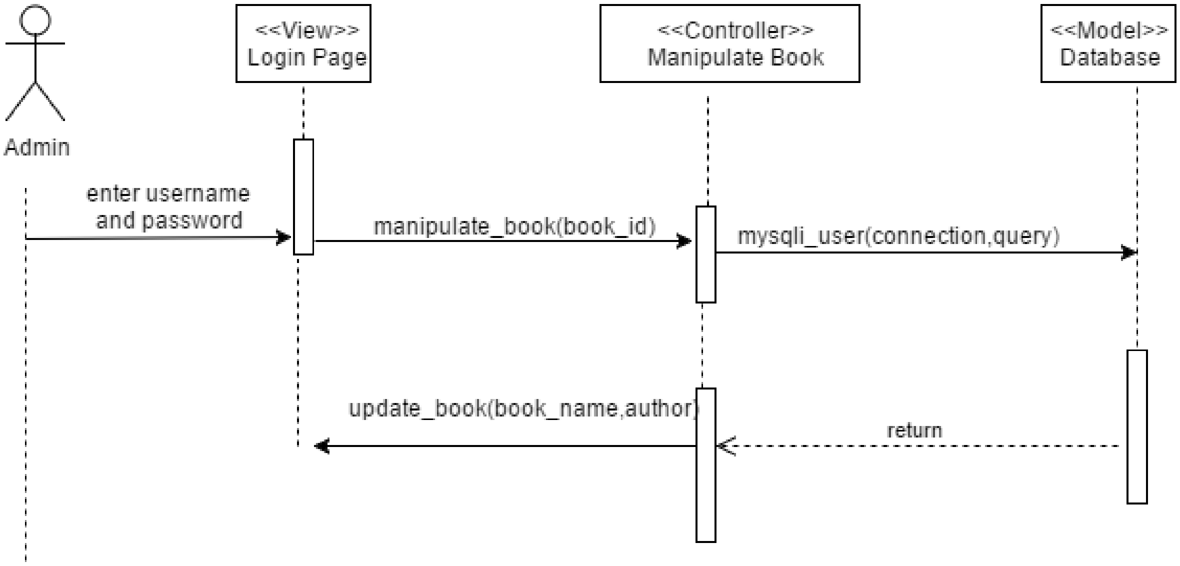


Manipulate Customer Sequential Diagram

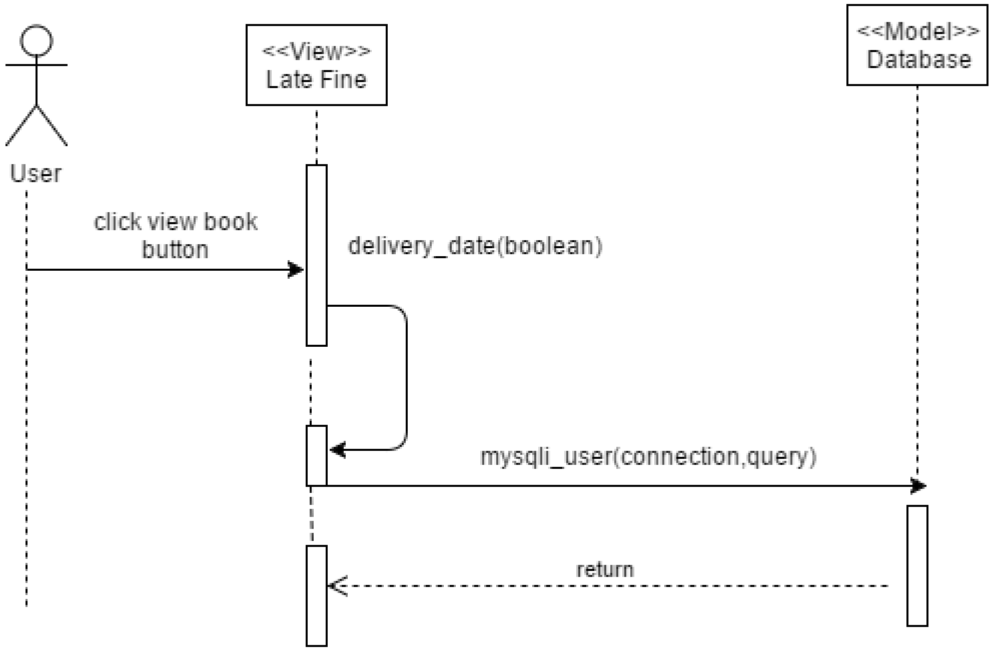


Admin
<<View>>
Login Page
<<Controller>>
Manipulate Customer
<<Model>>
Database
enter username and password
manipulate_customer(customer_id)
mysqli_user(connection,query)
update_customer(customer_name,password)
return

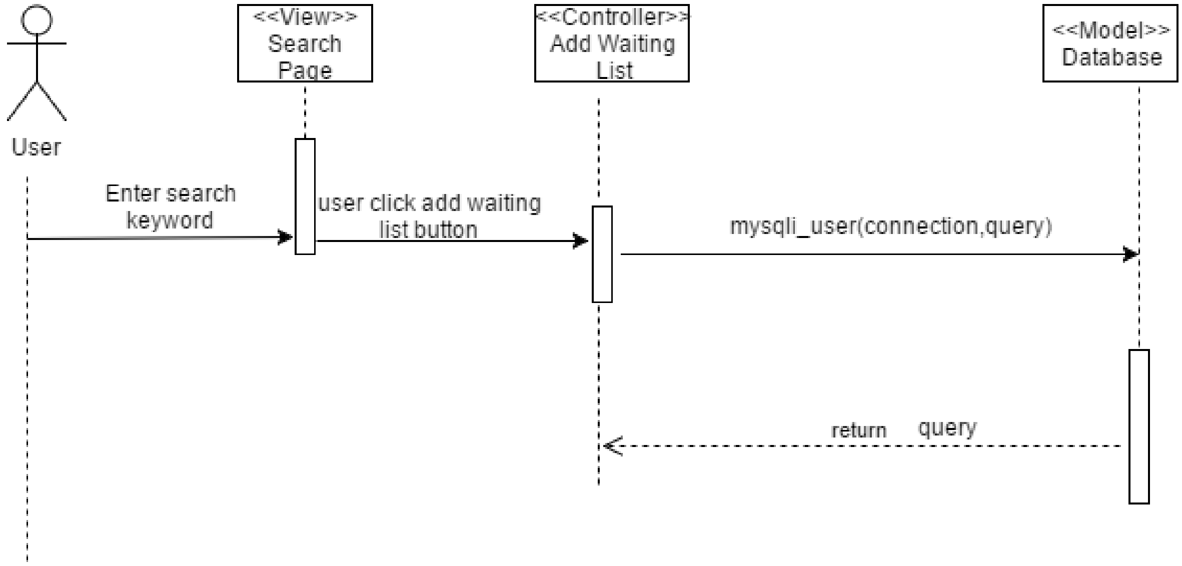
Manipulate Book Sequential Diagram



Issue late fine Sequential Diagram



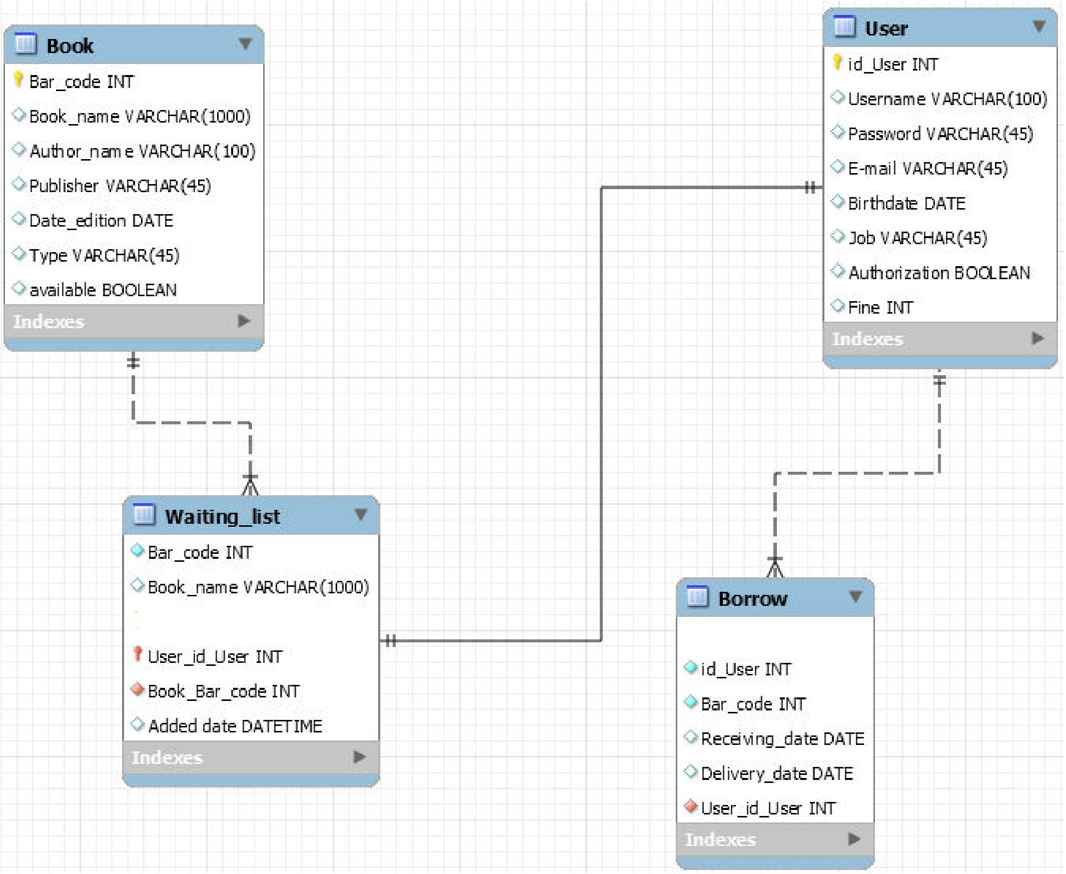
Add Book Waiting List Sequential Diagram



4.3 Data Model (E-R Diagram)

There were some changes in the er diagram.
These changes did not change much in design. We removed the borrow_id property on the waiting list and borrow entities.
Borrow id do not need to keep it so we removed the diagram.

The new er diagram is as follows:



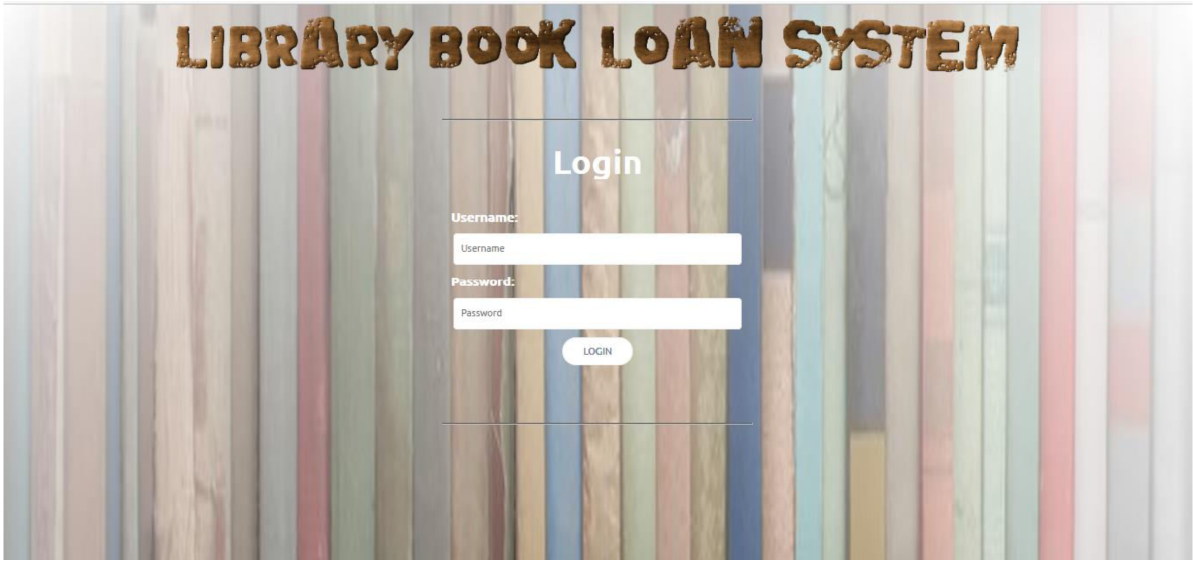
4.4 User Interface Design

We made some changes in this section. First we changed all interface parts.Changing it to make it look more beautiful. We think that more users will visit the system after the changes. We wanted to use a more colorful and interesting structure for the system background.Of course, the changes we make are not the final version of the system.We can make changes to our system again.

The system now has interfaces for login, logout, view book, search book, self return, self checkout, homepage and wait book use cases.

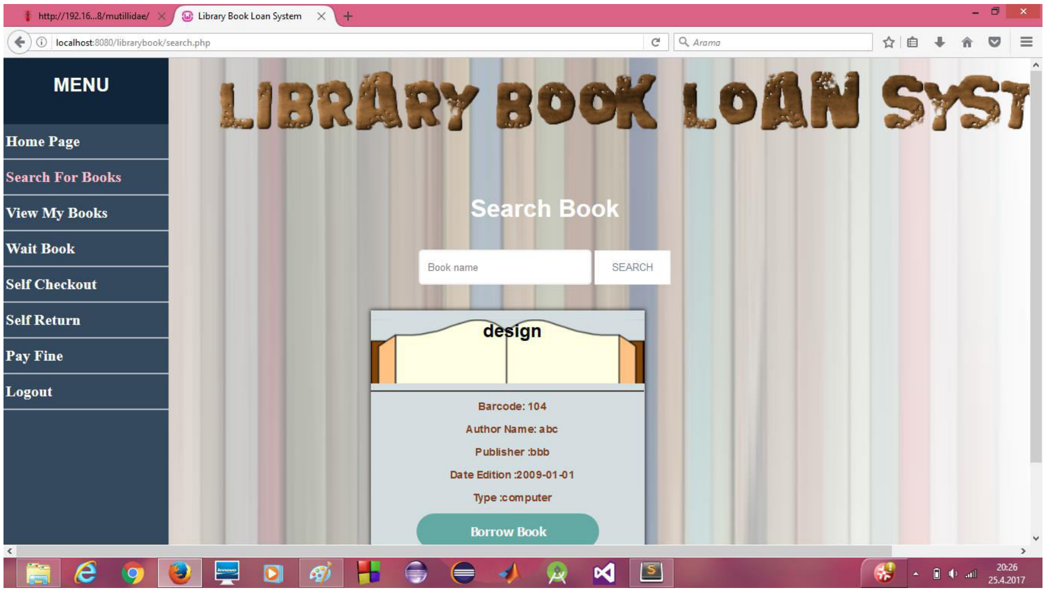
You can refer to the updated interface designs below:

Login interface:



-If the user enters the user name and password incorrectly, they will get an error message and cannot log in to the system.

User homepage:



Barcode: 104

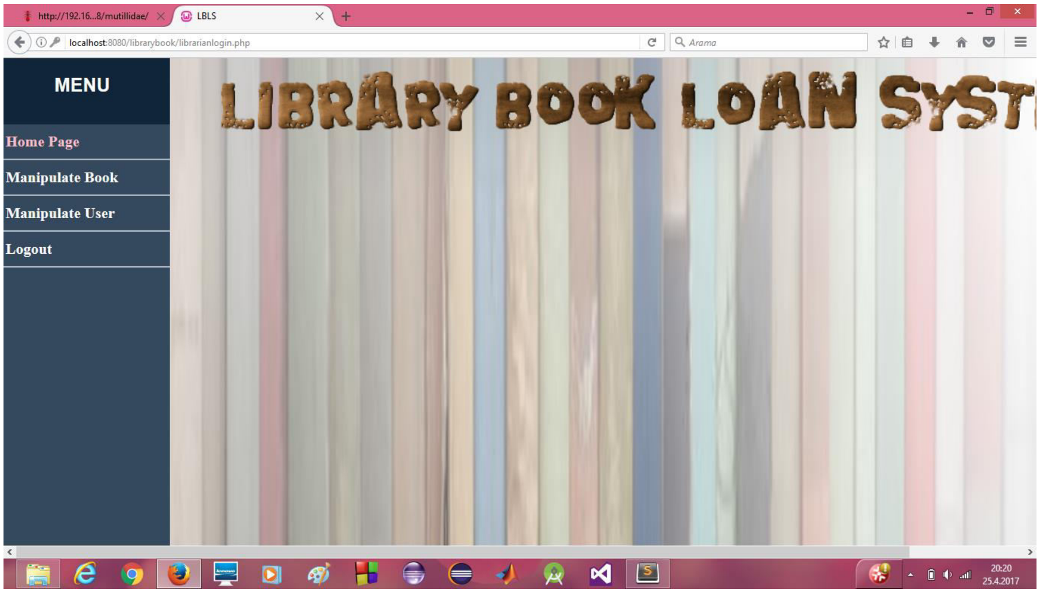
Author Name: abc

Publisher :bbb

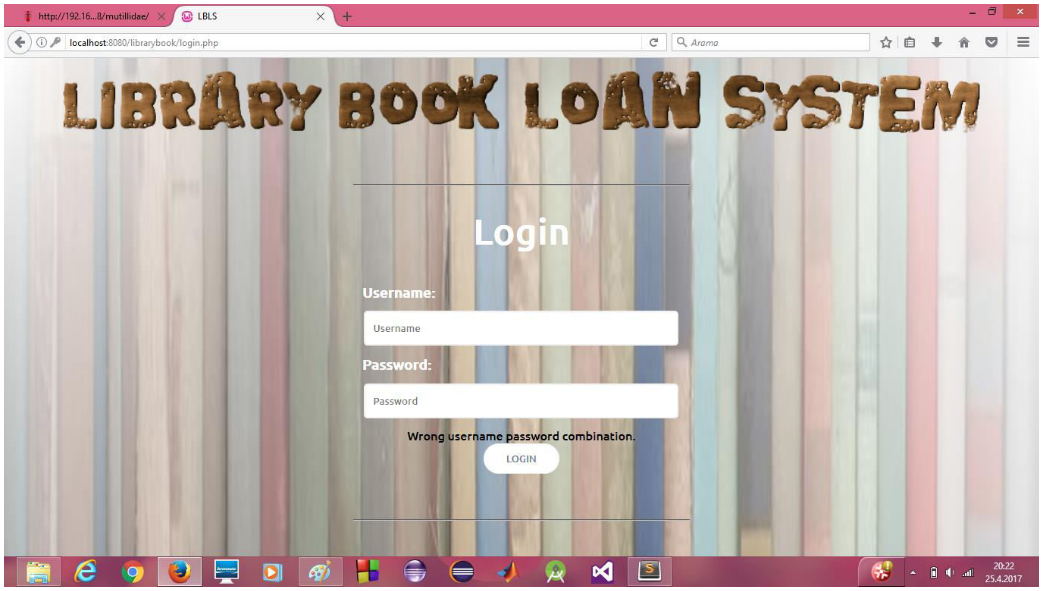
Date Edition :2009-01-01

Type :computer

Admin homepage:

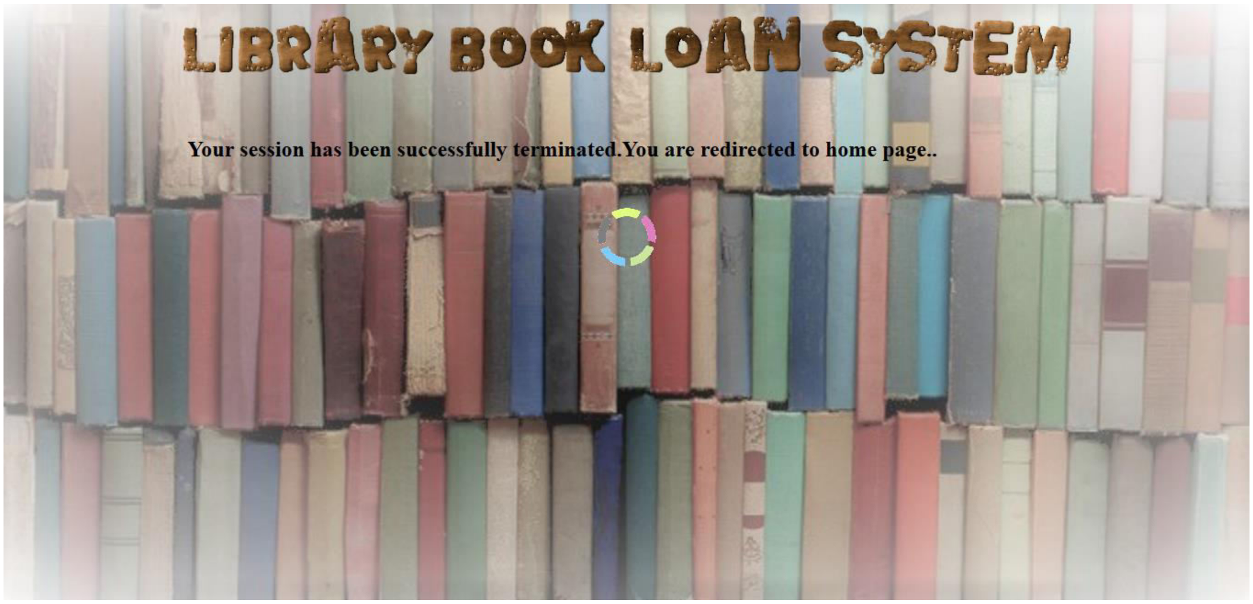


Wrong login:



Wrong username password combination.

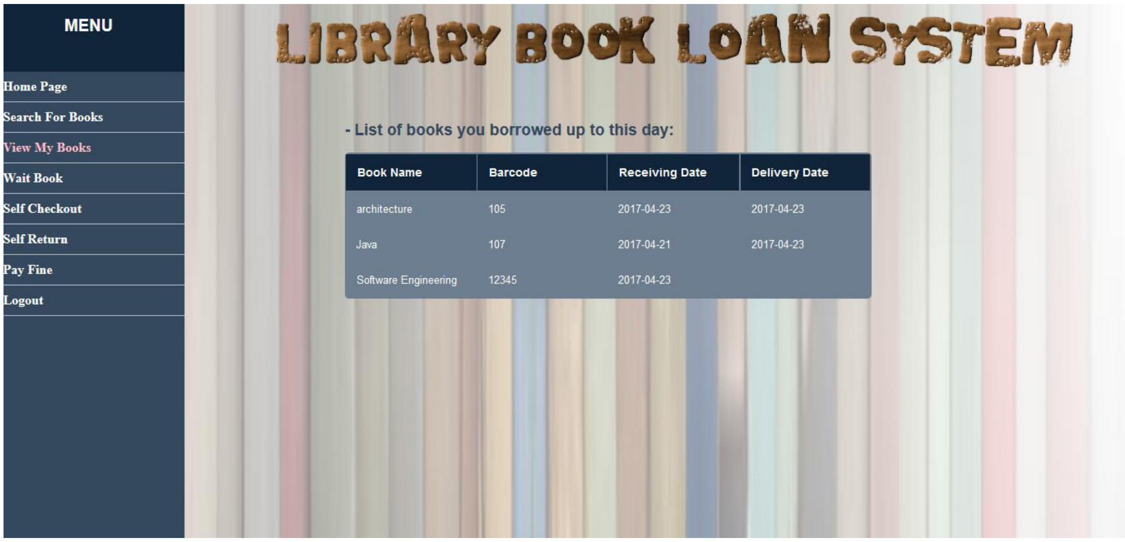
Logout interface:



Your session has been successfully terminated.You are redirected to home page..

-The user must log in before logging out. Pressing the button for logout is sufficient.

View Books interface:

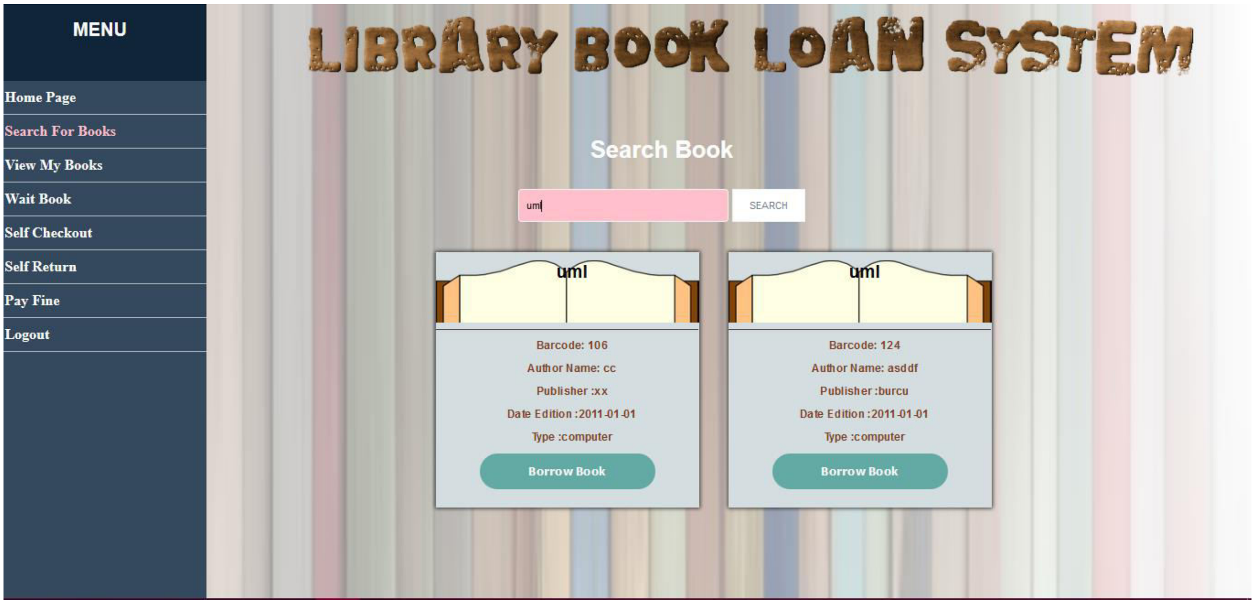


- List of books you borrowed up to this day:

Book Name	Barcode	Receiving Date	Delivery Date
architecture	105	2017-04-23	2017-04-23
Java	107	2017-04-21	2017-04-23
Software Engineering	12345	2017-04-23	

-If the user has already taken a book, they can look at these options. You need to log in before you can go to this option.

Search Books interface:



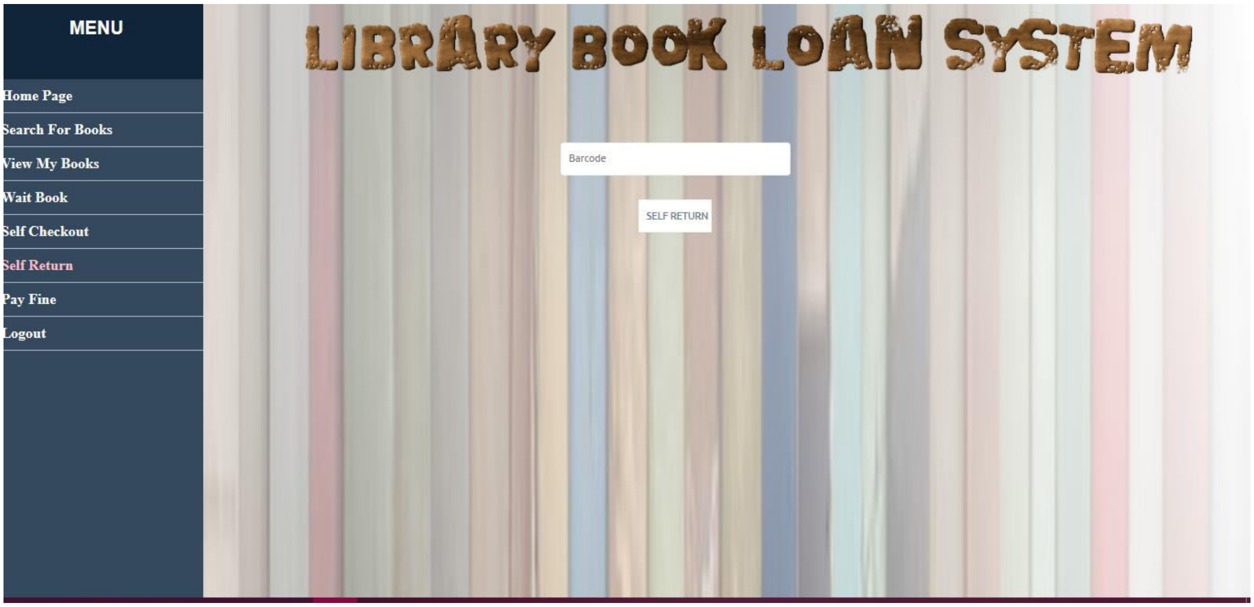
uml

Barcode: 106
Author Name: cc
Publisher :xx
Date Edition :2011-01-01
Type :computer

Barcode: 124
Author Name: asddf
Publisher :burcu
Date Edition :2011-01-01
Type :computer

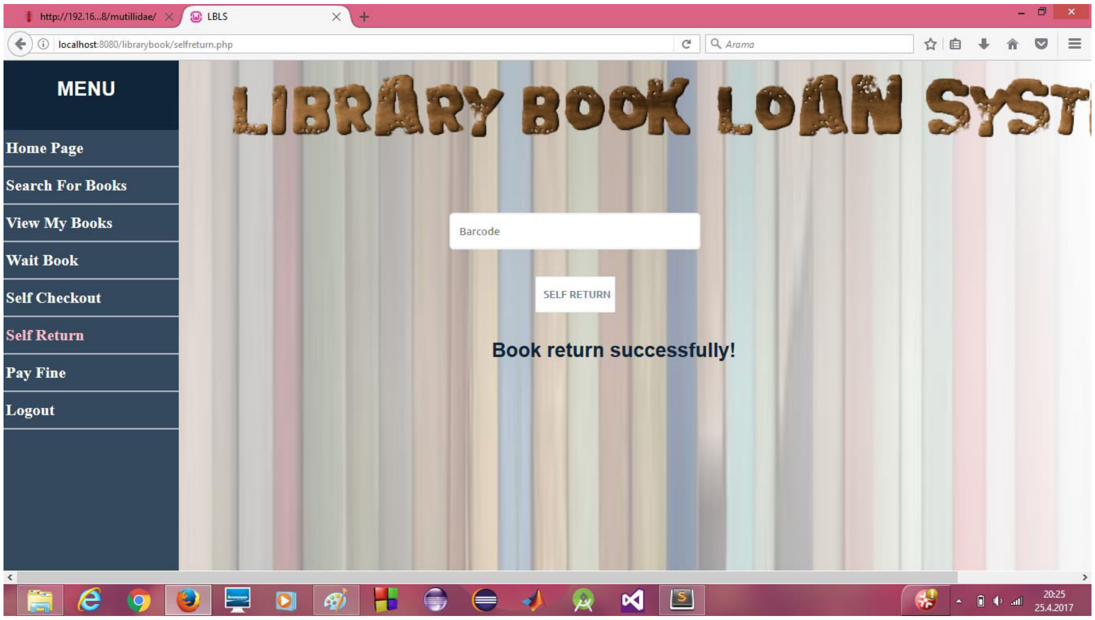
-Anyone can search the book without having to login. You can type the name of the book you want in search.

Self-return interface:



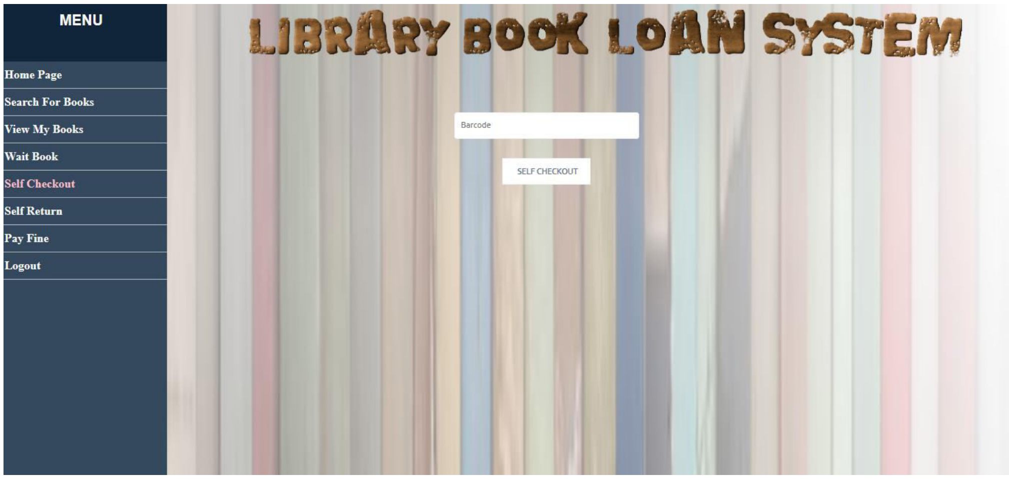
-If the user clicks on this button after logging in, the book can be borrowed.

Self-return successful message:



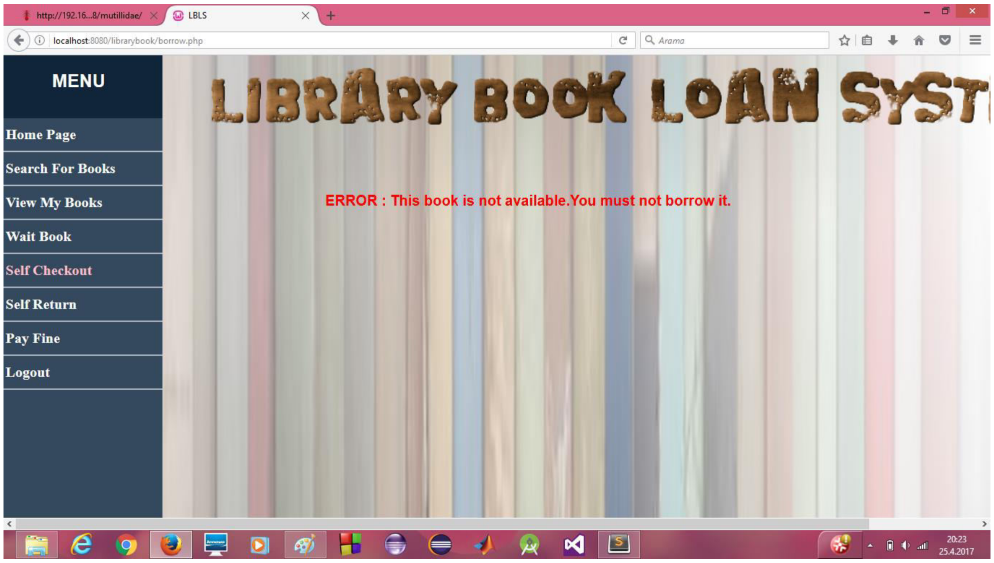
Book return successfully!

Self-checkout interface:



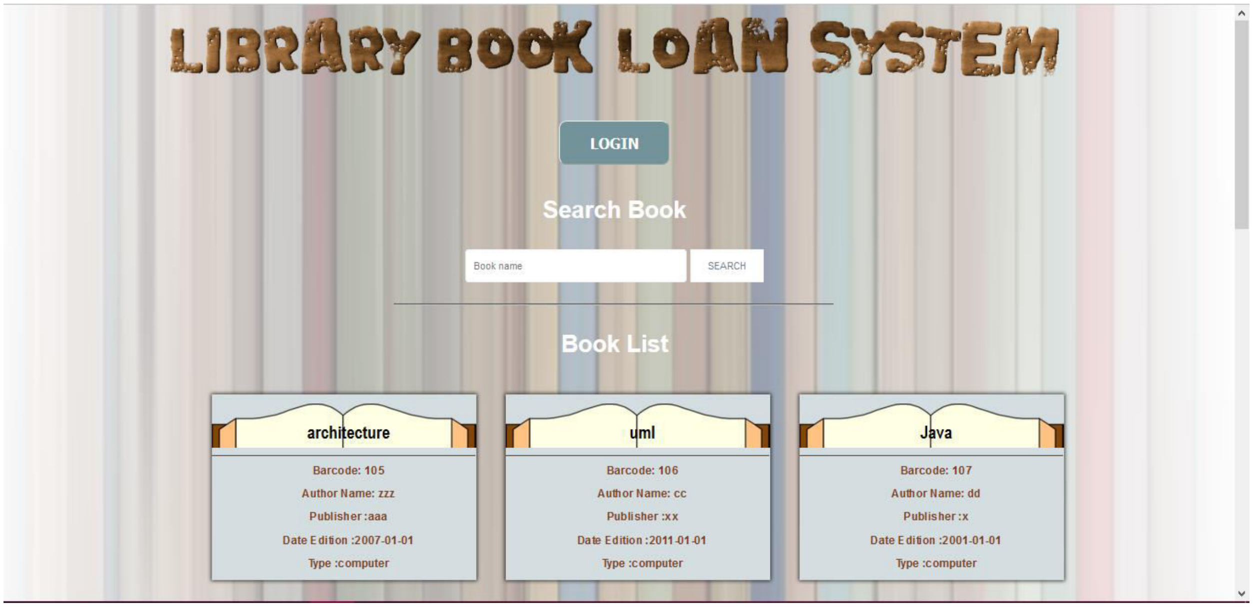
-The user can return some of the books he or she has taken after clicking the button. There is no refund if there is no book.

Not available book for self-checkout:



ERROR : This book is not available.You must not borrow it.

Homepage interface:



architecture

Barcode: 105
Author Name: zzz
Publisher :aaa
Date Edition :2007-01-01
Type :computer

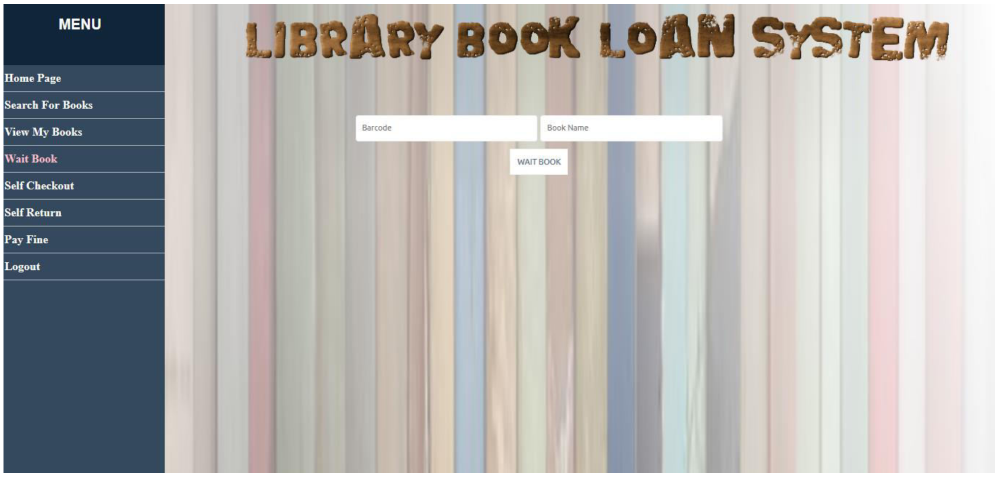
uml

Barcode: 106
Author Name: cc
Publisher :xx
Date Edition :2011-01-01
Type :computer

Java

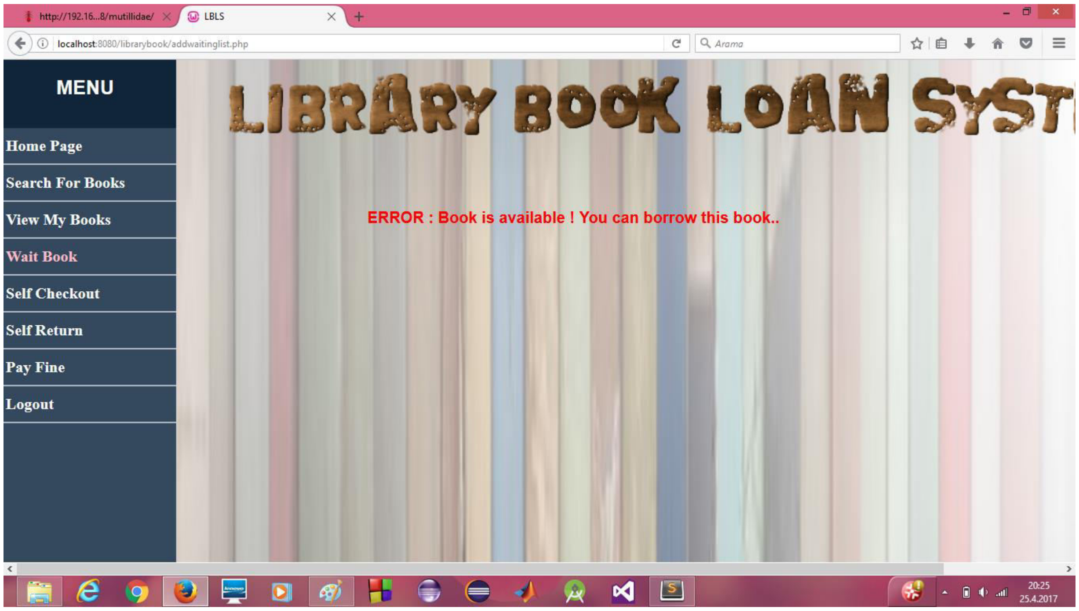
Barcode: 107
Author Name: dd
Publisher :x
Date Edition :2001-01-01
Type :computer

Wait book interface:



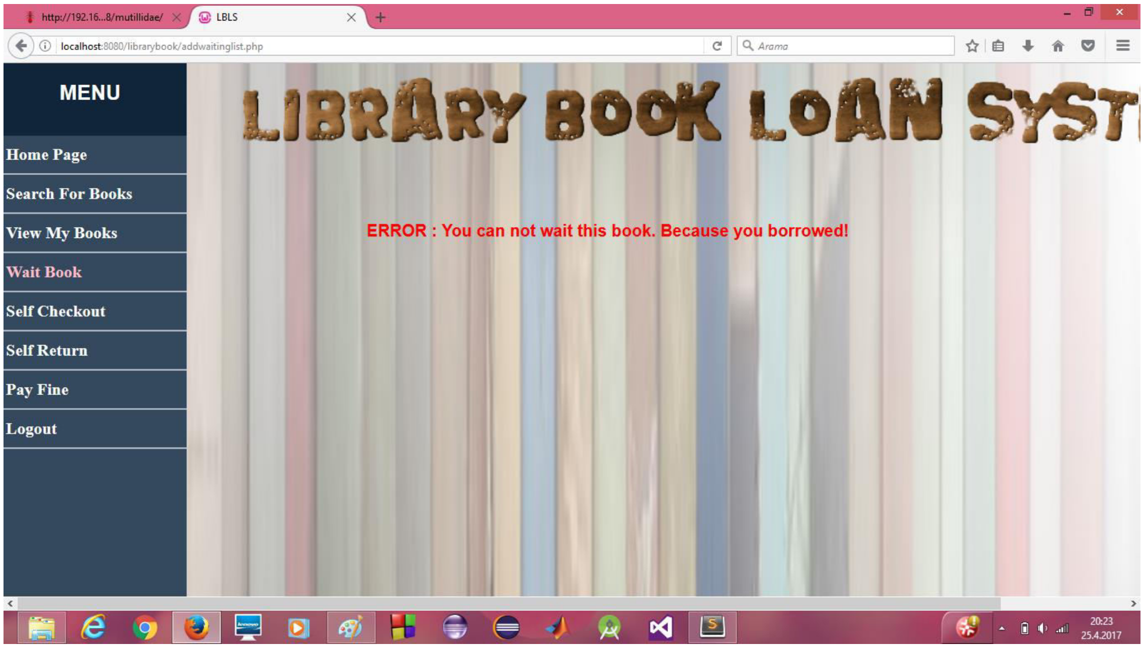
-The user may choose to book the book. But the user can not separate the same 2 books.

Warning message is given when you want to add a book that is available to the wait book list:



ERROR : Book is available ! You can borrow this book..

If the borrowed book is tried to be added to the wait list:



ERROR : You can not wait this book. Because you borrowed!

5. Requirements Traceability

------	index.php	login.php	logout.php	search.php	Selfreturn.php	Selfcheckout.php	Manipulate_book.php	Waitbook.php	Manipulate_customer.php	Viewbooks.php
Login	x	x								
Logout	x	x	x							
Search Book	x			x						
View Book	x	x								x
Manipulate Customer	x	x					x			
Manipulate Book	x	x							x	
Self-return	x	x			x					
Self-Checkout	x	x				x				
Late Fine	x	x								x
Pay Fine	x	x								x
Notify of book being avaliable	x	x						x		
Add name to book waiting list	x	x								