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

Group Number:5

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

# <Library Book Loan System>
# System-Wide Requirements Specification

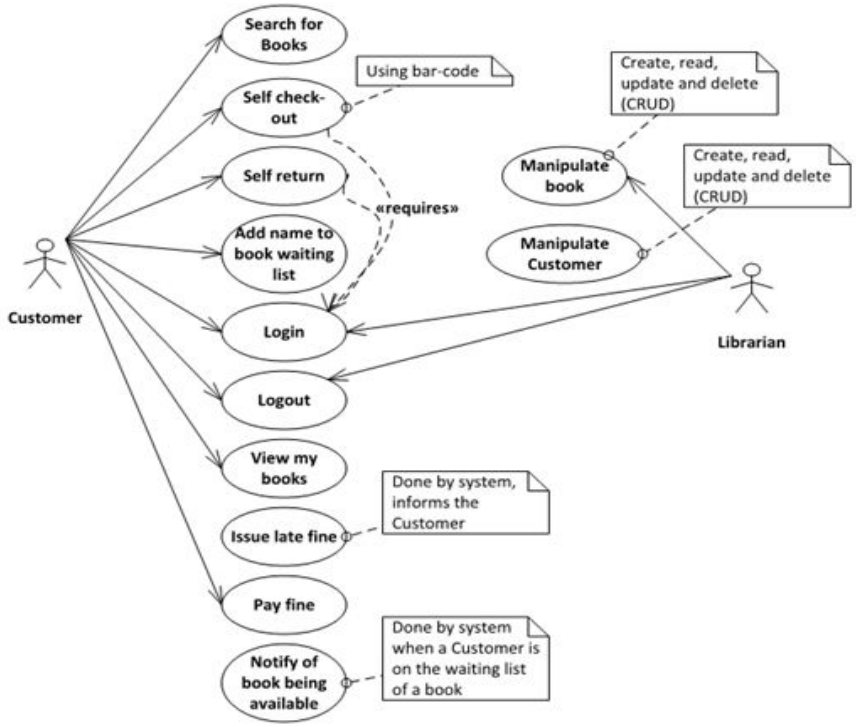
## 1. Introduction

The purpose of the this document prepared for the library book loan system, defined the functional and non-functional requirements of the system; design the user interfaces, show the main functions of the product with various diagrams, determine the system constraints, according to system quality attributes defined the system qualities and document them. Target group of the this project are: system administrators (librarians) , guests who can review the systems and books and users who can borrow books.
With this project, we aim to make an web application that requires less workload and time for system users and administrators.

## 2. System-Wide Functional Requirements

The functional requirements of the system are detailed with the use case diagram, tabular descriptions and ER diagram below.

**Use-case diagram:**



**Tabular descriptions:**

| **Use Case : Login** | |
|---|---|
| Actors | Customer,Librarian |
| Pre - condition | Customer / librarian not logged in |
| Post - condition | Customer / librarian logged in |
| Main (Happy) path | 1. User enter her/his username and password and presses on the Login button<br>2. System query to database find user and control password.<br>3. User is login and will be able to use features the system. |
| Alternative path | 1. If the username and password combination is invalid system give an message to user.<br>2. User will be able to enter another username and password |

| **Use Case : Logout** | |
|---|---|
| Actors | Customer,Librarian |
| Pre - condition | Customer / librarian must be logged in |
| Post - condition | Customer / librarian logged out |
| Main (Happy) path | 1. User presses on the logout button<br>2. User is logged out and views home page |

| **Use Case : Self - Checkout** | |
|---|---|
| Actors | Customer |
| Pre - condition | User must be logged in |
| Post - condition | User borrow book |
| Main (Happy) path | 1. User enter the barcode of the he wants to borrow into the barcode system.<br>2. User click the Self-Checkout button.<br>3. System checks whether the book is available.<br>4. System loan the book to user and show the information message |
| Alternative path | 1. If the book is not available system send user the message.<br>2. User be directed to Self-Checkout page. |

| **Use Case : Self - return** | |
|---|---|
| Actors | Customer |
| Pre - condition | User must be logged in |
| Post - condition | User returns the book |
| Main (Happy) path | 1. User chooses the book which he/she wants to return from the view my books page<br>2. User click the Self-return button.<br>3. System take back the book and marks the book as available |

| **Use Case : Search for books** | |
|---|---|
| Actors | Customer |
| Pre - condition | Customer in the search book page |
| Post - condition | Customer view book |
| Main (Happy) path | 1. User enters book name<br>2. System check the book in database<br>3. System bring to book and user view book informations |
| Alternative path | 1. If the book is not in the system give message to user<br>2. User will be able to search and view another books. |

| **Use Case : Add name to waiting list** | |
|---|---|
| Actors | Customer |
| Pre - condition | Customer must be logged in |
| Post - condition | Customer name added in book waiting list |
| Main (Happy) path | 1. User enter “wait book” button in book page<br>2. System add the user to book waiting list<br>3. System give the book is reserved message to user |
| Alternative path | 1. If the waiting list is free system give a message that book can be borrow now to user |

| **Use Case : View my books** | |
|---|---|
| Actors | Customer |
| Pre - condition | Customer must be logged in |
| Post - condition | Customer view own books history |
| Main (Happy) path | 1. User enter “view my books” button<br>2. System query books from database<br>3. User view all books which he/she borrowed |
| Alternative path | 1. If the user has not borrowed any books, system gives nothing to display message |

| **Use Case : Notify of book being available** | |
|---|---|
| Actors | System Agent |
| Pre - condition | A book has been returned |
| Post - condition | User receive notification |
| Main (Happy) path | 1. System control book waiting list<br>2. System send notification to waiting user the book is available |

| **Use Case : Issue late fine** | |
|---|---|
| Actors | System Agent |
| Pre - condition | A book has not been returned at the appointed time |
| Post - condition | User get fine |
| Main (Happy) path | 1. System checks who borrowed the book<br>2. System send notification about get late fine to user<br>3. System add users fine to database |

| **Use Case : Pay fine** | |
|---|---|
| Actors | Customer |
| Pre - condition | User logged in |
| Post - condition | User pay fine |
| Main (Happy) path | 1. User press the my fines button<br>2. System query user's fines information to database.<br>3. System brings up the payment system screen<br>4. User enter payment information.<br>5. User click 'pay fine' button. |
| Alternative path | 1. If payment system give an information error user directed to payment screen again.. |

| **Use Case : Manipulate book** | |
|---|---|
| Actors | Librarian |
| Pre - condition | Librarian must be logged in |
| Post - condition | Librarian manipulate(create,read,update,delete) books |
| Main (Happy) path | 1. Librarian press (create/read/update/delete) book button<br>2. System queries the database for the requested book operation<br>3. Librarian complete the selected action |
| Alternative path | 1. If there are problems with the operation to be performed(ex:read book / there is no such book in the system etc) , system send the information message to librarian.<br>2. Librarian control book information and can be able to choosed manipulate operation |

| **Use Case : Manipulate customer** | |
|---|---|
| Actors | Librarian |
| Pre - condition | Librarian must be logged in |
| Post - condition | Librarian manipulate(create,read,update,delete) customer |
| Main (Happy) path | 1. Librarian press (create/read/update/delete) customer button<br>2. System queries the database for the requested user operation<br>3. Librarian complete the selected action |
| Alternative path | 1. If there are problems with the operation to be performed(ex:delete user / there is no user in the system etc) , system send the information message to librarian.<br>2. Librarian control user information and can be able to choose other manipulate operation |

**Data Model :**

The library book loan project needs some knowledges . These knowledges can be grouped 4 main headings.

One of them is for **books**, in this project needs :

- bar-code number
- books' name,
- author of books,

- publisher ,
- date edition ,
- available to keep book is borrowed or not,
- type.

Second heading is for **user**, in this project needs :

- username for create account and login to project,
- password for create account and login to project,
- e-mail address to receive notification from the system,
- birthdate to use for payment fine system,
- job for the different users may has different roles in system,
- authorization for the show user has any authorization,
- fine to keep the amount of fine every user.

Third heading is for **borrow**, in this project's needs,

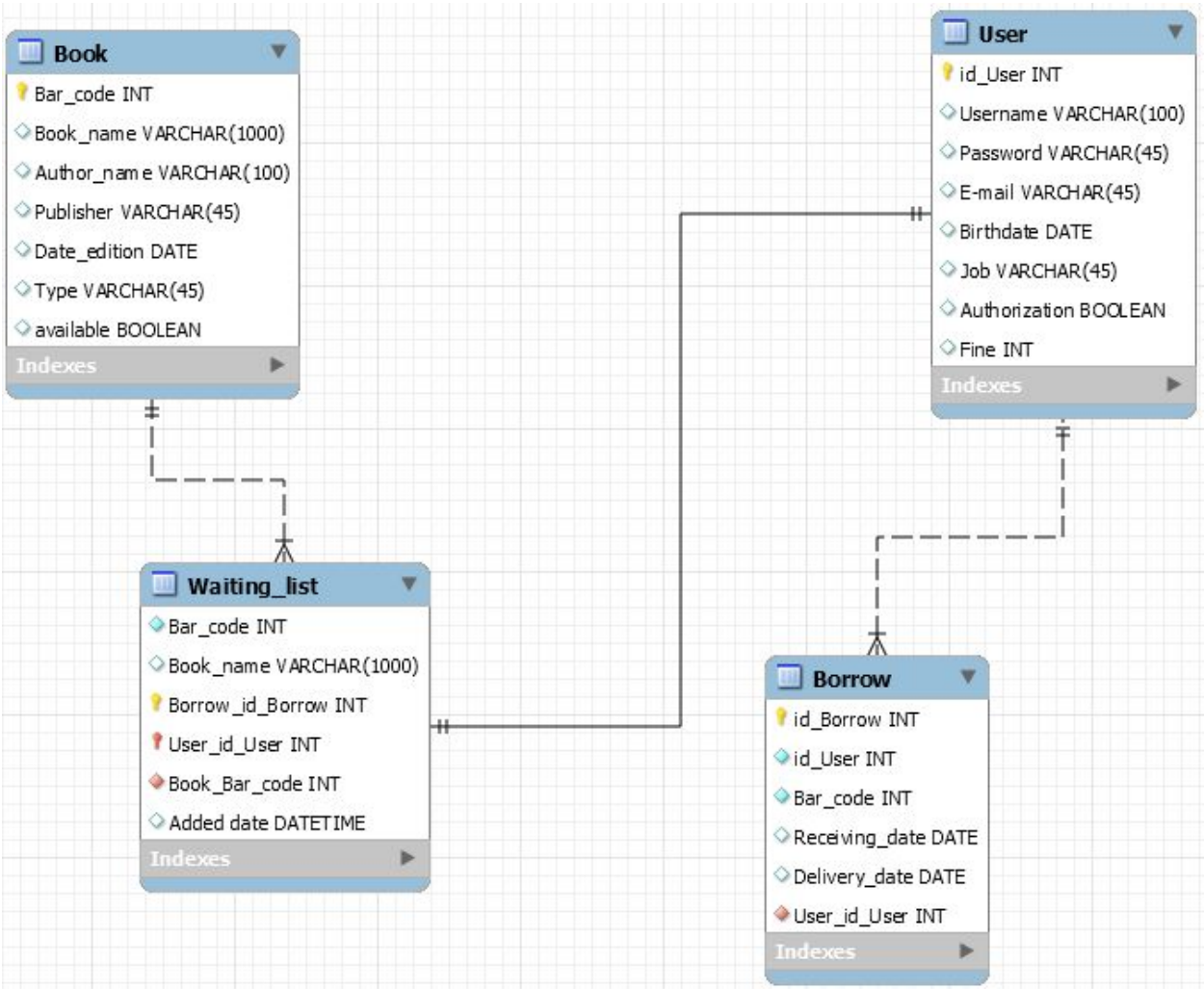
- user id to keep who borrow the book,
- bar-code number to keep which book is borrowed,
- receiving date to control the borrow process,
- delivery date to control the borrow process.

The fourth heading is for **waiting lists** of users, this project needs:

- user id to keep who has the list,
- bar-code number to keep which book is in the list,
- book name to keep which book is in the list,
- added time to keep when the book is added list.

According to these knowledges the project is used a database which has 4 tables. In this project, listed knowledges and tables are showed with "Entity Relationship Diagram(E/R diagram).Tables in database is called entity, listed knowledges are attributes of entities. Also these tables has relationships each other.

**ER Diagram:**




- A user has a waiting list. So waiting list and user tables has 1:1 relationship.

- A user may has many borrowed books. So user and borrow table has one to many relationship.

- A waiting list may contain many books. So waiting list and book tables has one to many relationship.

# 3. System Qualities

According to system quality attributes(URPS) describe system quality below.

### 3.1 Usability

User interfaces are designed in a simple and understandable so that the system is easy to understand and easy to use. For this reason, users will not have any complexity when using the system.

### 3.2 Reliability

This web-based application will not have any reliability problems.
All book information and user information will be stored in the database in case of any problems in case the responsible person can be accessed immediately and the problem will be removed.As a result system will be reliable.

### 3.3 Performance

Database tables were designed to speed up the system. Database queries are planned for the speed up the system.

### 3.4 Supportability

The library system needed a lot of resources for the above functional requirements. We also php and mysql, which are advanced technologies.We also preferred html css and js technologies to provide a simple interface to the user.

We tried to keep the database and system design as flexible as possible, considering the features that could be added in the future.

As a result ,we have planned a system that is easy to maintain and open to new features.

# 4. System Interfaces

Our system is web based library system.You can see the detailed interface properties at the bottom of the interface section.

## 4.1 User Interfaces

### 1- User Interface: Login

This division is entered with the username and password.Each user has only one account.

After this information is entered, the system is entered.

### 2- User Interface: Logout

The user can logout by clicking the logout option in the upper right corner of the system.

### 3- User Interface: Searching Books

The user can search for the desired book by making the book name, author name, or Isbn number in the search area.

### 4- User Interface: Viewing Books

The user can enter my books view page. He can see books that bought previous or current books.

### 5- User Interface: Pay Fine

When the customer enters the penalty payment page, they can see the list of penalties that they have already paid and if there are unpaid penalties, they are listed as in the table. If there are penalties that you want to pay, you can do so by marking the payment boxes and pressing the pay button.

### 6-User Interface: Book Reservation

When the customer enters the book reservation page, the system will notify the user when he / she wants to borrow the book by entering the name of the book and clicking on the list on the waiting list.

**7-User Interface: Manipulate Book**

The librarian can add a new book by pressing Add new book button.

**8-User Interface: Manipulate Customer**

The librarian can add a new customer by pressing Add new customer book button.

### 4.1.1 Look & Feel

Our interface is simple and understandable.System is designed with the convenience of users.It has been decided that the interface will be in the shades of blue.For this reason the light blue tone is less of a human eye.We can change this idea later.

### 4.1.2 Layout and Navigation Requirements

First of all, if the book is searched in the system, there is no need login by user.In other cases the system needs to be logged in. All of these transactions are done through the user account.

### 4.1.3 Consistency

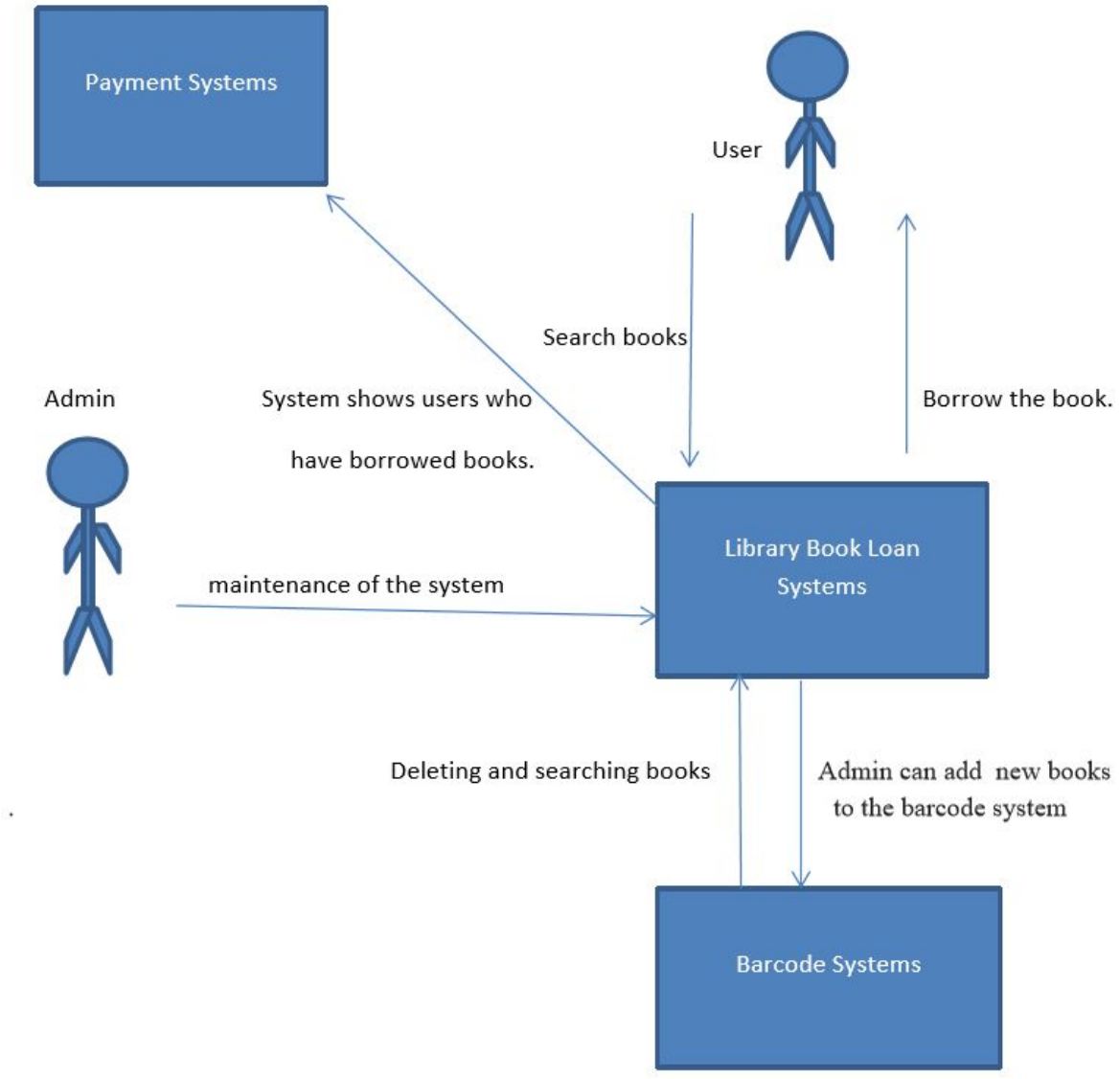
In this system a user can buy books, make reservations, view book penalties.
These parts are very important for a library system. For example, if the borrowing date of a user's book has passed, the user can see it in a short period of time and user borrower by paying less than the penalties amount.

### 4.1.4 User Personalization & Customization Requirements

The user interfaces in the system will be fixed for all users.Users can only change their personal information.They can not make any changes about the interface.

## 4.2 Interfaces to External Systems or Devices

The system we will build will be valid on the web platform. Android based systems can be accessed using a web browser.



### 4.2.1 Software Interfaces

We will use a web based system for library system.We are considering using the php language  but we can change it later.

### 4.2.2 Hardware Interfaces

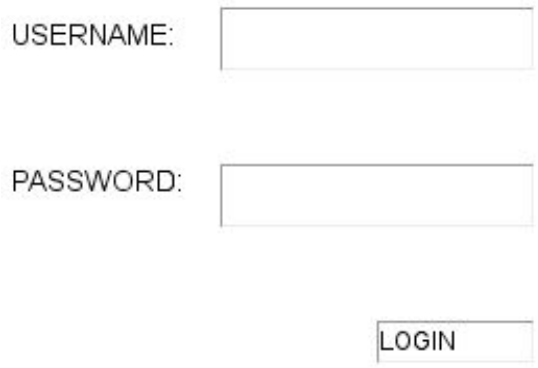
The barcode system will also be used in this application.Borrowing or give back to books will be completed sooner.There will be a library system besides this, and the system will managed by admins

### 4.2.3 Communications Interfaces

The application applies to all systems using the web base platform.Internet is required for this application.

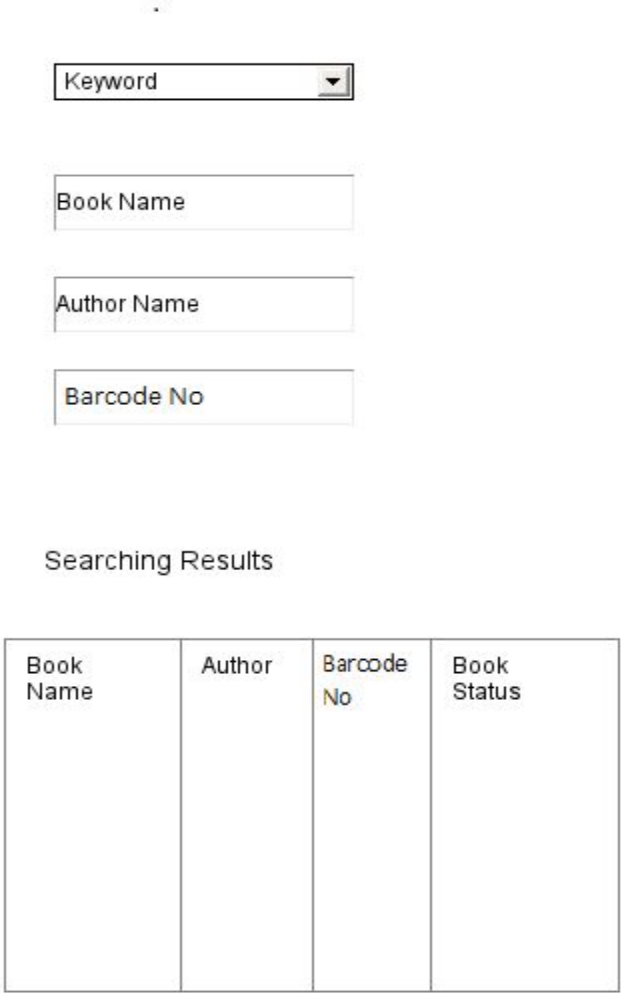
**USER INTERFACE DESIGN TEMPLATE**

**1-Login Page:**

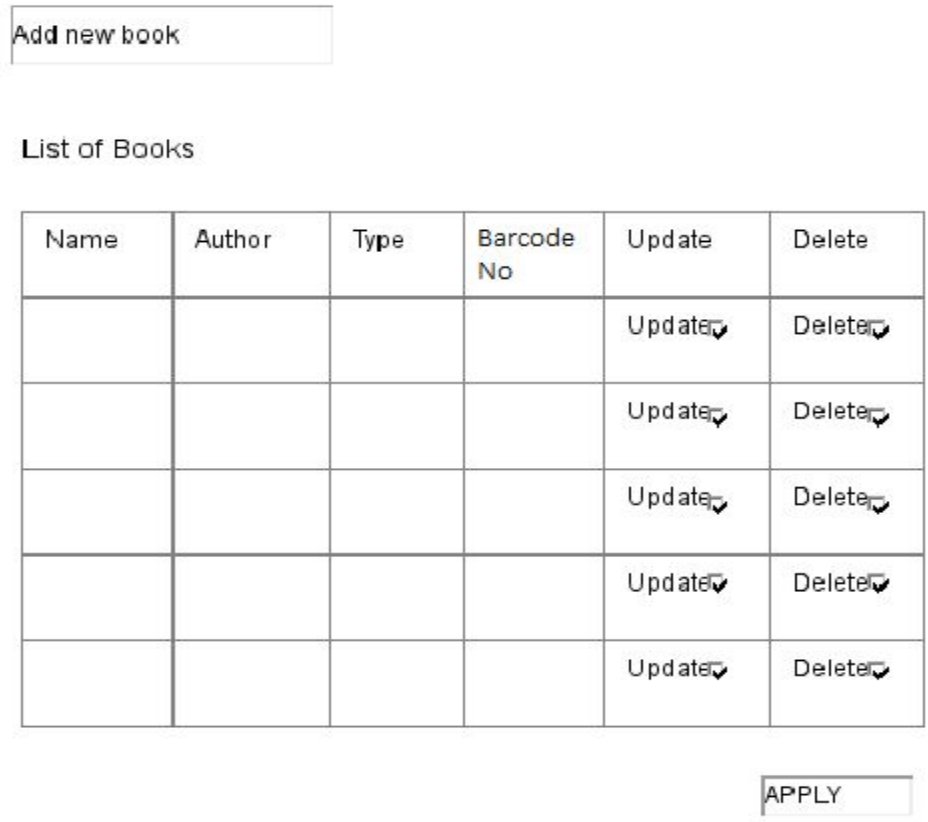


**2-Logout Page:**

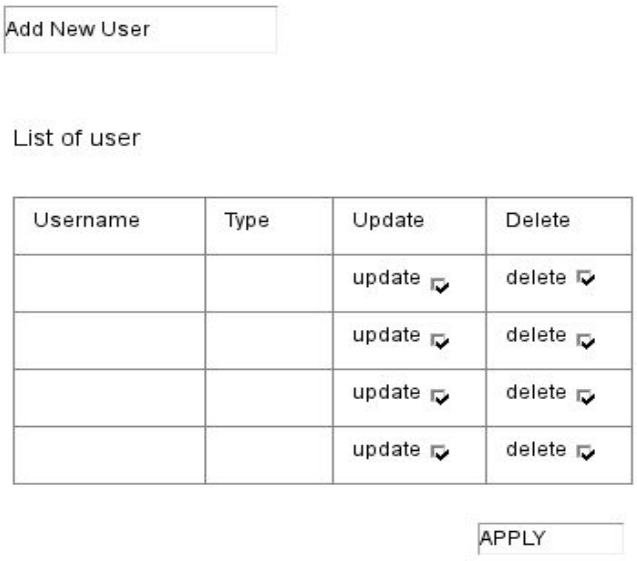
## 3-Searching Book Page:



## 4-Manipulate Book Page:



## 5-Manipulate Customer Page:



## 6-View Book:

Previous Books

| Book name | Author | Starting Time | Finishing Time |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |

Current Books

| Book name | Author | Starting Time | Finishing Time |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |

## 7-Pay Fine Page:

Previous Penalty

| Book name | Starting Time | Finishing Time | Late Days | Penalty Results |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

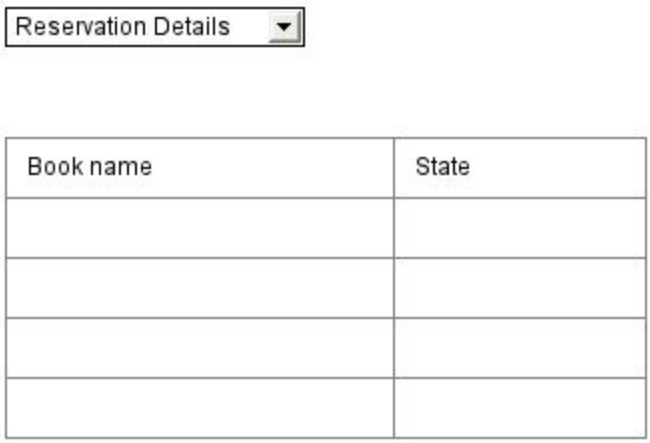
Current Penalty

| Book name | Penalty Results | Select |
|---|---|---|
| | | ○ |
| | | ○ |
| | | ○ |
| | | ○ |

Pay

## 8-Book waiting list page:


## 9-Notify of book being available page:



## 10-Self Checkout:

Self Checkout

| Book Name | Username | Starting Time | Finishing Time | Borrowing |
|---|---|---|---|---|
| | | | | ☑ |
| | | | | ☑ |
| | | | | ☑ |
| | | | | ☑ |

SAVE

## 11-Self Return:

Self Return

| Book Name | Username | Starting Time | Finishing Time | Self Return |
|---|---|---|---|---|
| | | | | ☑ |
| | | | | ☑ |
| | | | | ☑ |
| | | | | ☑ |

SAVE

# 5. Business Rules

## 5.1.<Calculation of User Fine Rules>

***<1.Fine rule >***

If the user does not bring the book on time, the user will be charged a fine amount of 0.50 □ each day.

***<2.Issue late fine rule>***

If the user has fine amount ,the system sent to notification to this user.

## 5.2.<Borrow Books Rules>

***<1.Borrow a book rule>***

If a book is available ,the user can borrow this book.

***<2.Delivery date rule>***

If a user borrow a book , it must deliver book at maximum one month after.

***<3.Number of borrowed books rule>***

If a user borrowed three books before, the system show the message to user "You can not borrow book more than three" and not allowed borrow operation to fourth book.

## 5.3.<Sent the Notifications to Users Rules>

***<1.Added a book to Waiting list rule>***

If a book is in one more waiting list, the system controls the added time and sent the notifications according to the order of added time to users.

# 6. System Constraints

The project is a web based project, so the constraint for environment is internet. It is used in every environment which have internet. It does not contain any constraints for hardware. In the project, software language is chosen as php. Also, the project has a database which determined as Mysql.

## 7. System Compliance

### 7.1 Licensing Requirements

The project does not have any licensing enforcement requirements.

### 7.2 Legal, Copyright, and Other Notices

The project is prepared for the library of Hacettepe University. Copyright of this project belongs team. Guarantee of proper operation of the project is under team responsibility.

### 7.3 Applicable Standards

The following industry standards are used because the project is it prepared according to user requirements and the working performance will be tested:

-ISO/IEC 25062:2005: software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Common Industry Format (CIF) for usability test reports

-ISO/IEC 25064:2013: Systems and software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Common Industry Format (CIF) for usability: User needs report

In this project, the used quality standard is IEE STD 1061-1992.This standard works this way:it defines quality metrics and implementing and analyzing all process of project.

The project has interoperability property in every environment which has internet.

## 8. System Documentation

In this system has a “Help” button to help the users.If the user click the Help button ,it shows a page which contains the knowledge of how to use the system.All group members are responsible for the preparation this page.