

References

- [1] www.peoplesbank.lk/site_map.shtml
- [2] www.eosintl.com/
- [3] www.goalexandria.com/
- [4] www.gscweb.com/lmsonline.html
- [5] www.searchwindevelopment.techtarget.com/definition/C
- [6] Marciniak J.J. (2001.), Encyclopedia of Software Engineering, 2nd Edition, John Wiley and Sons, Inc, New York.
- [7] Summerville I. (2004), Software Engineering, Pearson Education Ltd, 7(1), pp 87-144
- [8] www.en.wikipedia.org/wiki/Feasibility_study#What_is_Feasibility_Study
- [9] www.developer.com/design/article.php/2247041
- [10] www.ibm.com/developerworks/rational/library/3101.html
- [11] www.databases.about.com/cs/specificproducts/g/er.htm
- [12] www.wikipedia.org/wiki/Relational_database
- [13] www.comp.lancs.ac.uk/computing/resources/lans/SE6/PDF/Introduction.pdf
- [14] www.en.wikipedia.org/wiki/Software_design
- [15] Ramez Elmasri , Shamkant Navathe, Fundamentals of Database Systems (6th Edition)
- [16] www.ambyssoft.com/essays/userInterfaceDesign.html

Use Case Diagrams

1. Member use case diagrams

Member has given facilities for View Books, Search Books, Issue Books, Reserve Books, Return Books and View Members.

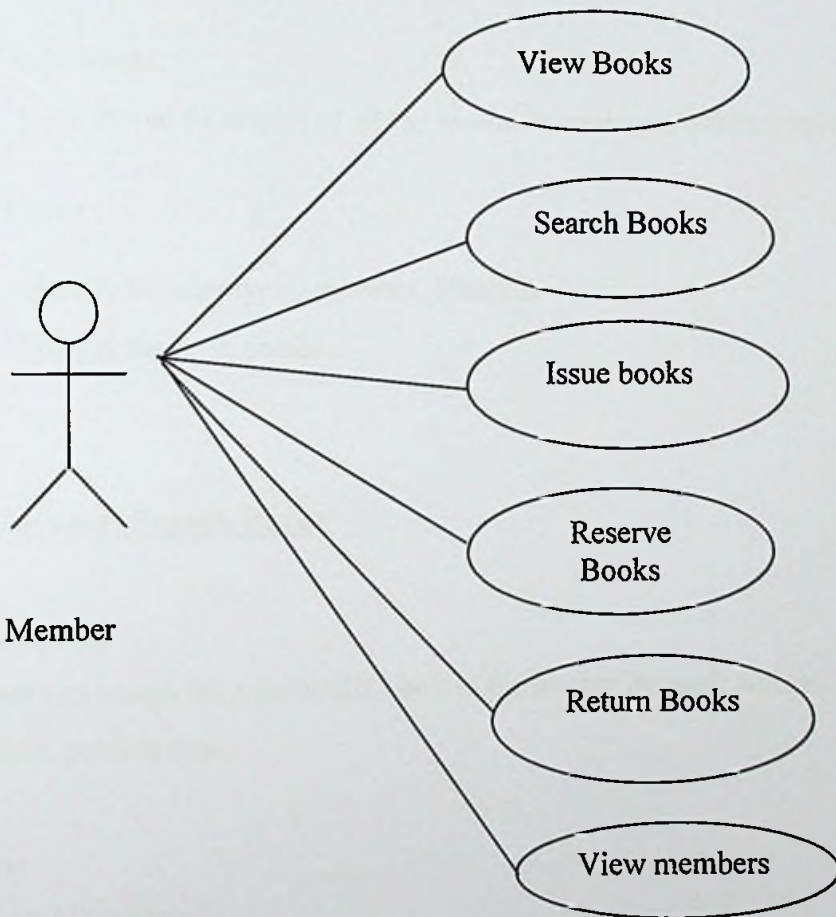


Figure A 1: Member use case diagram

1.1. Use case: View Books

Introduction:

To display the details to member or Librarian view about available books.

Actors:

Member, Librarian

Pre – Conditions:

Before this use case can be initiated, the user has already connected to the network

- User must have proper client installed on user terminal for log into the system.
- Books data should be stored in the database & available to retrieve.

Post – conditions:

- User should be displayed all the available books with the complete details.

Basic Flow:

- Identify the user type – member, librarian.
- Display the book details.

1.2. Use case: Search books

Introduction:

Member can search for a particular book in the library by book author, Title, ISBN, Publisher, publish date.

Actors:

Member, Librarian

Pre – Conditions:

Before this use case can be initiated, the user has already connected to the network

- User must have proper client installed on user terminal for log into the system.

- Books data should be available to search in the database.

Post – conditions:

- User should be displayed results of the search with the complete details.

Basic Flow:

- Identify the user type – member, librarian.
- Select the search type.
- Check the availability.
- View the search results.

Alternative Flow:

- The search is not found, a message will be prompted.
- The incorrect data entered, a message will be prompted.

1.3. Use case: Issue Books

Introduction:

This use case describes the process of issuing a certain book for a member by the librarian.

Actors:

Member, Librarian

Pre – Conditions:

Before this use case can be initiated, the user has already connected to the network

- User must have proper client installed on user terminal for log into the system.
- Member should enter member Id to the system.
- Books should be available to issue.

Post – conditions:

- Member should have the issued book.

- If member login into the system outside the head office library terminal, then librarian received the message with delivery details.
- Librarian send related book to related member through inter courier service.
- Confirmation message should be prompted.
- Database should be updated-issued.
- Member details should be updated.

Basic Flow:

- Get the member Id and book Id.
- Check the availability.
- Check no. of books taken.
- Issue the book.
- Update the system.

Alternative Flow:

- Member Id or book Id is incorrect, user will be prompted a message regarding the error.
- The given book is already issued prompt a message regarding that.
- User has already taken 2 books prompts a message.

1.4. Use case: Return Books

Introduction:

This use case describes the process of returning a certain book to the library by the member.

Actors:

Member, Librarian

Pre – Conditions:

- Librarian should be logged into the system.

- Member should be borrowed books.
- Member should give the member Id to the librarian.

Post – conditions:

- Library should have the returned book.
- Confirmation message should be prompted.
- Database should be updated-available.
- Member details should be updated.

Basic Flow:

- Get the member Id and book Id.
- Check the Id accuracy.
- Enter returned the book id.
- Update the system.

Alternative Flow:

- Member Id or book Id is incorrect, user will be prompted a message regarding the error.

1.5. Use case: View Members

Introduction:

This use case describes the displaying of registered member's details to member or librarian.

Actors:

Member, Librarian

Pre – Conditions:

- Librarian should be logged into the system.
- Member data should be stored in the database and available to retrieve.

Post – conditions:

- Display all the members with the complete details.

Basic Flow:

- Identify the user type – member or librarian.
- Enter Id.
- Display the member details.

Alternative Flow:

- User does not want to print the details, and then can ignore the step.

2. Librarian: Use case diagrams

Librarian has to do functions, such as add/remove books, and add/remove members.

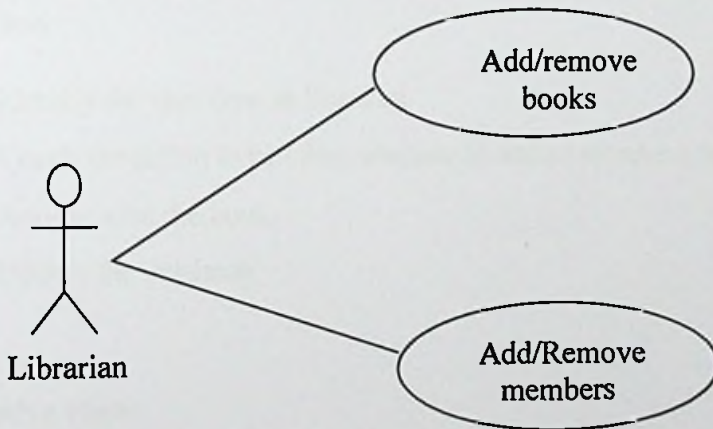


Figure A 2: Librarian use case diagram

2.1. Use case: Add/Remove Books

Only librarian allowed adding or removing books from the library database.

Actors:

Librarian

Pre – Conditions:

- Librarian should be logged into the system.
- Books should be available to remove.
- Book details should be available to add.

Post – conditions:

- Confirmation message displayed after remove/add successful.
- Database should be updated.

Basic Flow:

- Identify the user type as librarian.
- Check the action to be taken whether to add or remove a book.
- Add/remove the book.
- Update the database.



Alternative Flow:

- The books Id is already available, and then prompts the error message.
- The book Id is invalid when removing, prompt the error.
- The book is not available to remove, prompt the error.

2.2. Use case: Add/Remove Members

Only librarian allowed adding or removing members from the library database.

Actors:

Librarian

Pre – Conditions:

- Librarian should be logged into the system.
- Member should be available to remove.
- Member details should be available to add.

Post – conditions:

- Confirmation message displayed after remove/add successful.
- Database should be updated.

Basic Flow:

- Identify the user type as librarian.
- Check the action to be taken whether to add or remove a member.
- Add/remove the member.
- Update the database.

Alternative Flow:

- The member Id is already available, and then prompts the error message.
- The member Id is invalid when removing, prompt the error.
- The member is not available to remove, prompt the error.

Activity Diagrams

Activity diagrams for each operation

1. Activity diagram for Case : View User Details

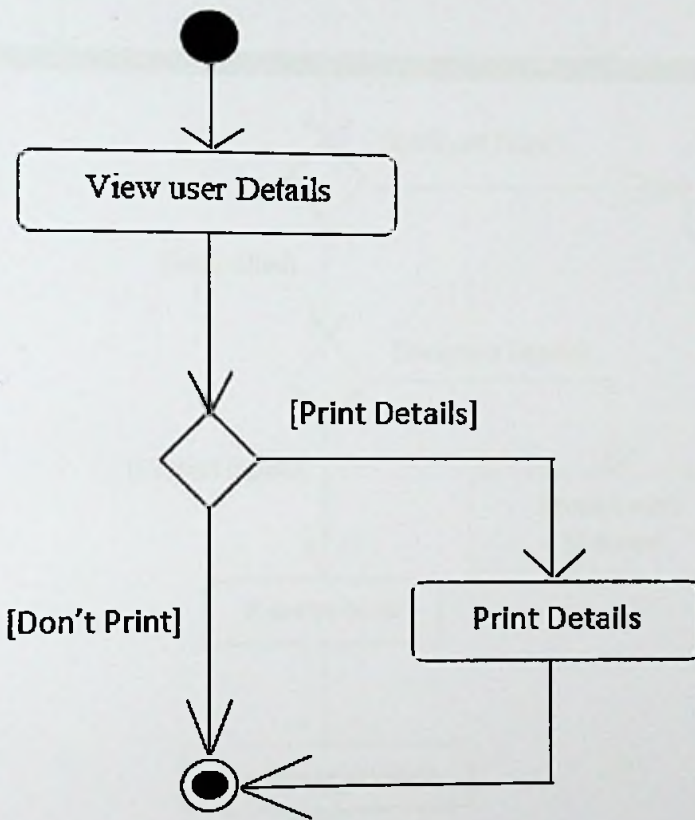


Figure B.1 : View user details

2. Activity diagram for Case : Reserve Books

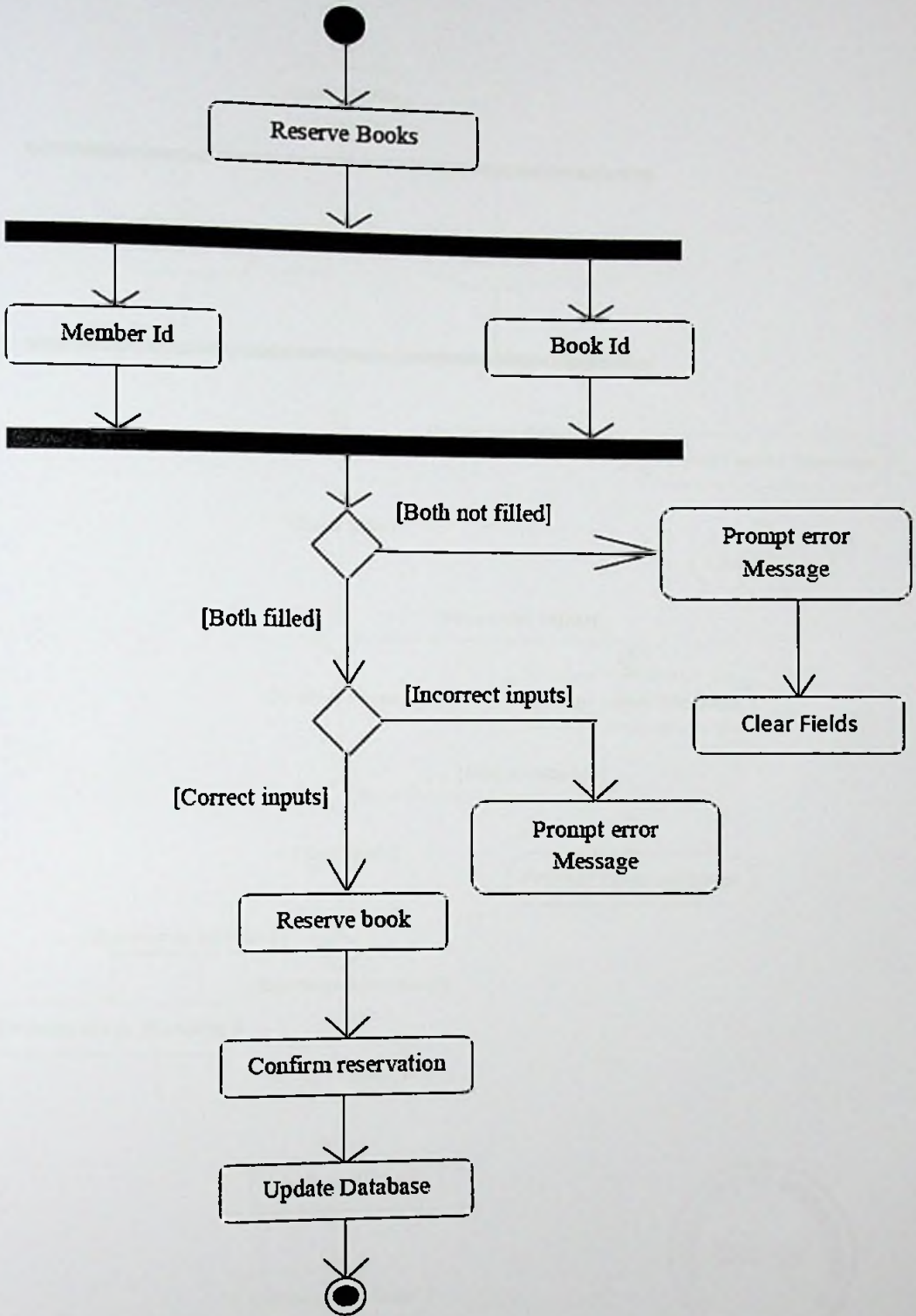


Figure B.2 : Reserve Books

3. Activity diagram for Case : Issue Books

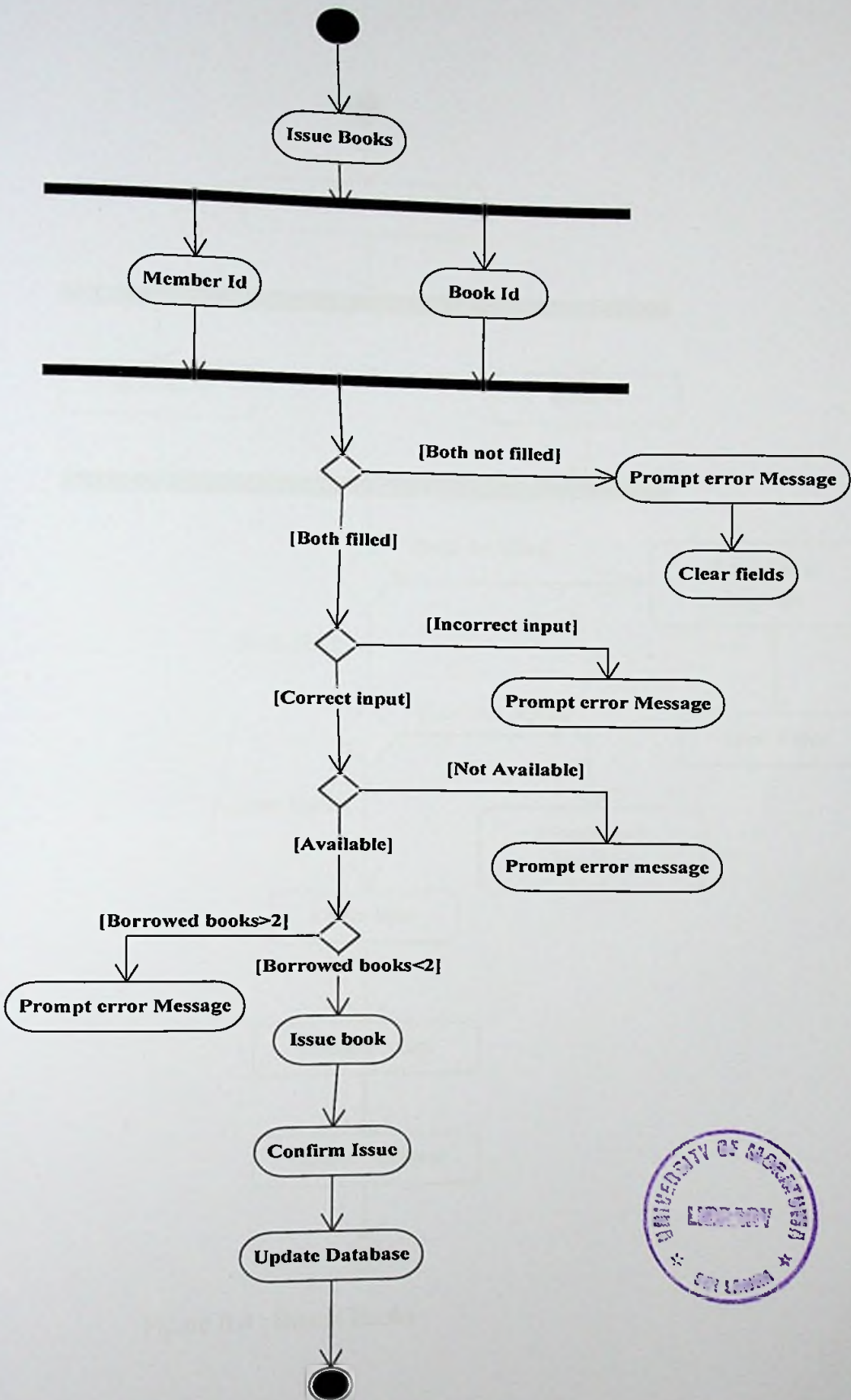


Figure B.3 : Issue Books

4. Activity diagram for Case : Return Books

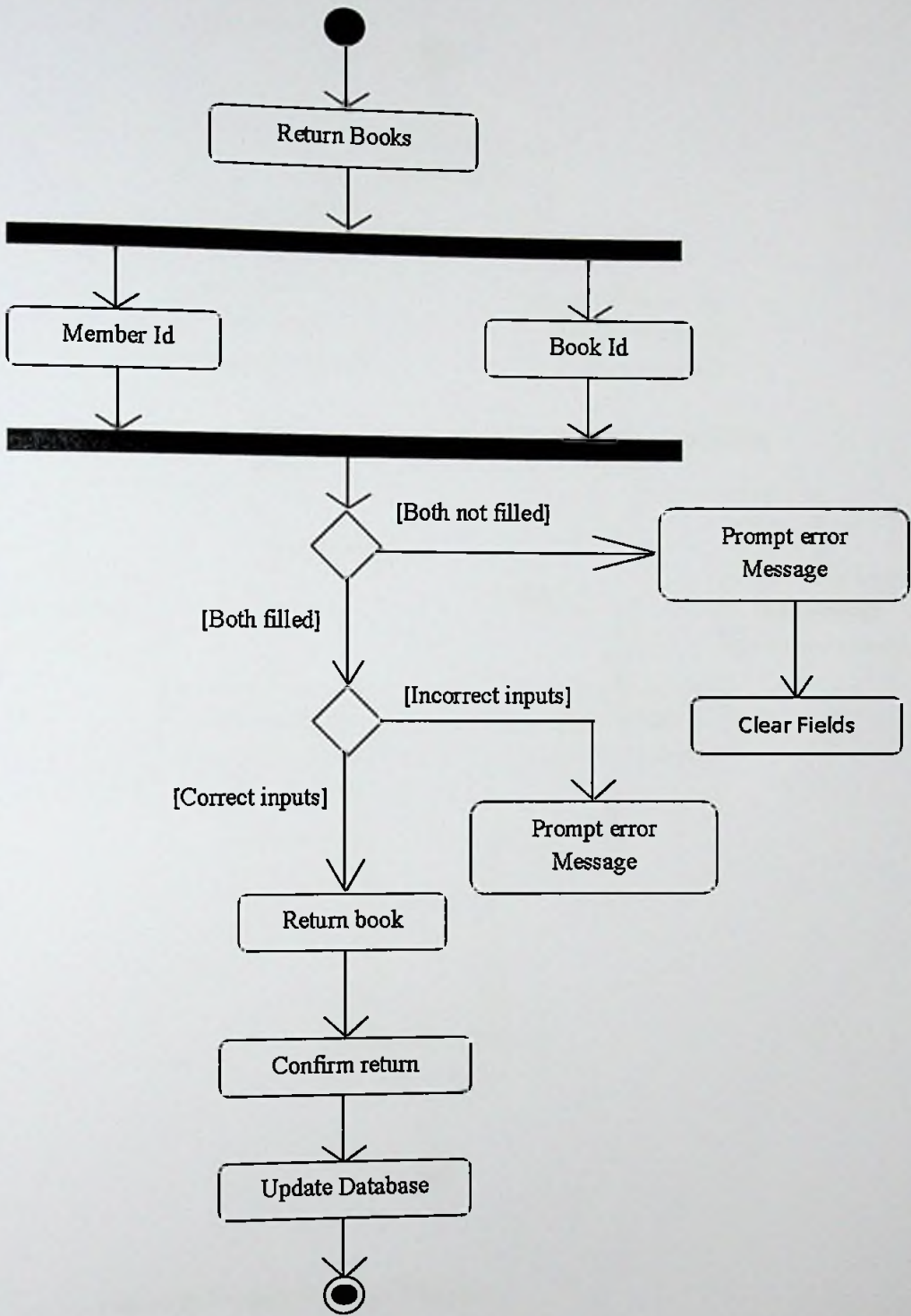


Figure B.4 : Return Books

5. Activity diagram for Case : Add/Remove Member

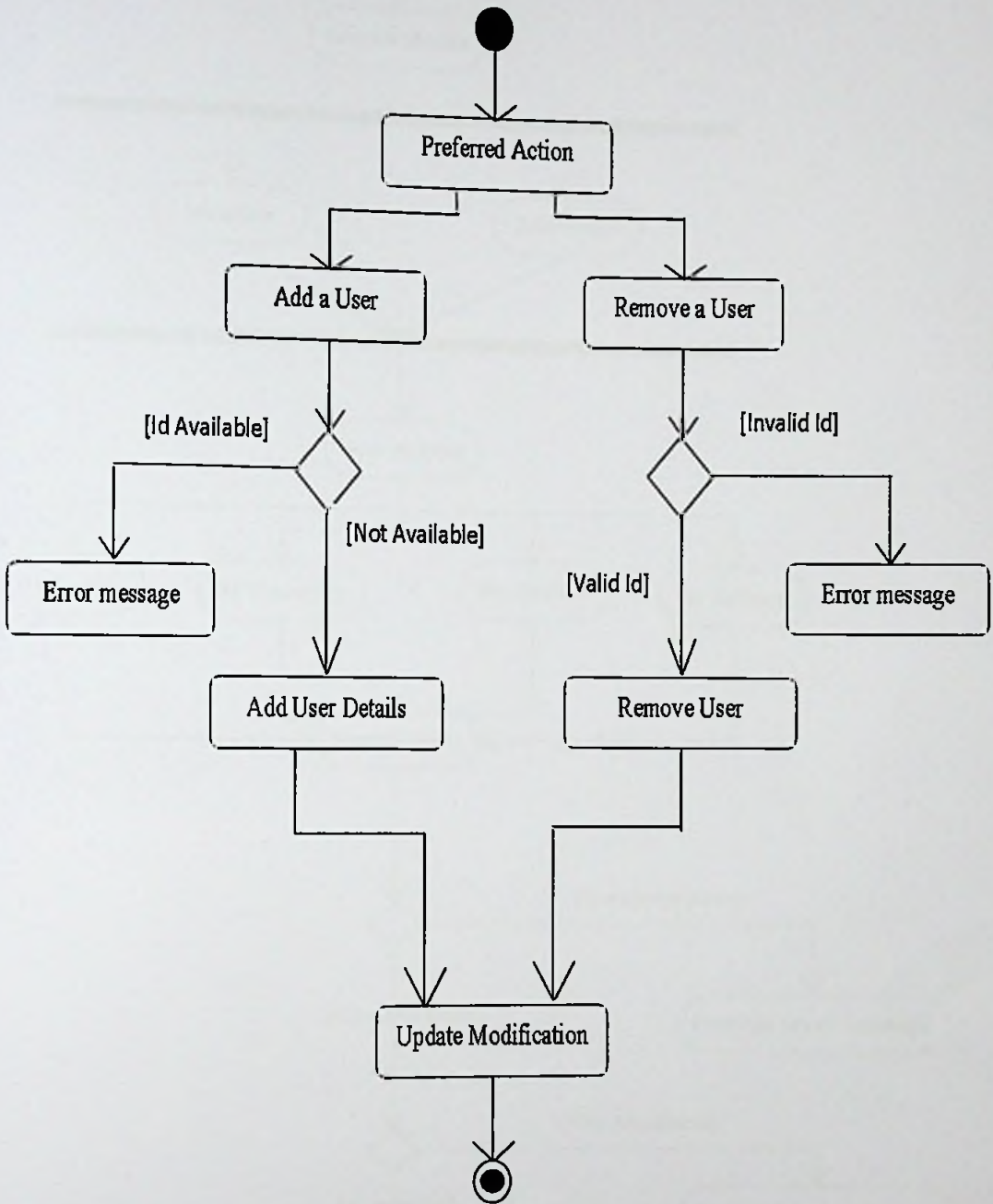


Figure B.5 : Add/Remove Member

6. Activity diagram for Case : Search Books

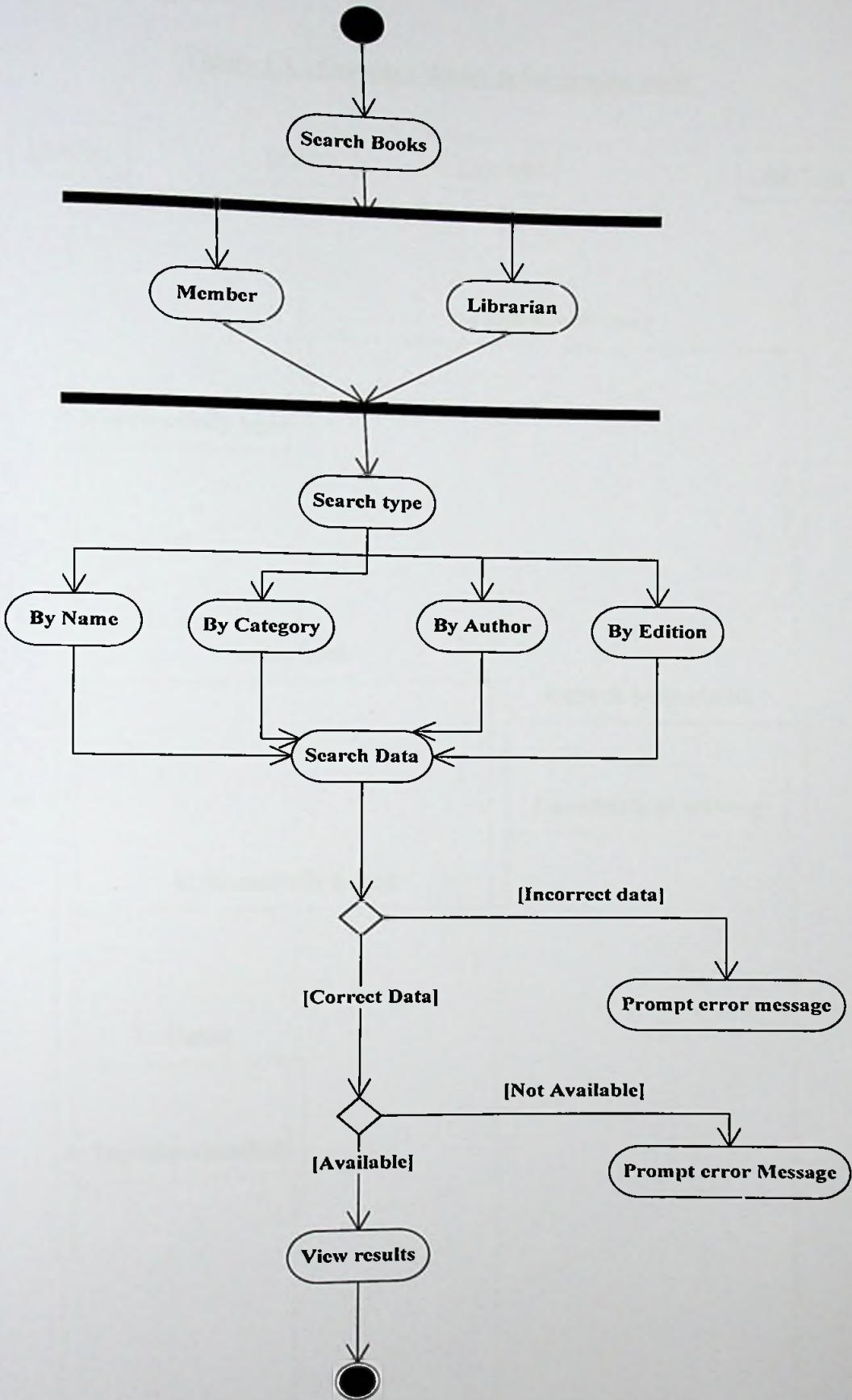


Figure B.6 : Search Books

Figure C1 : Sequence diagram for issuing book

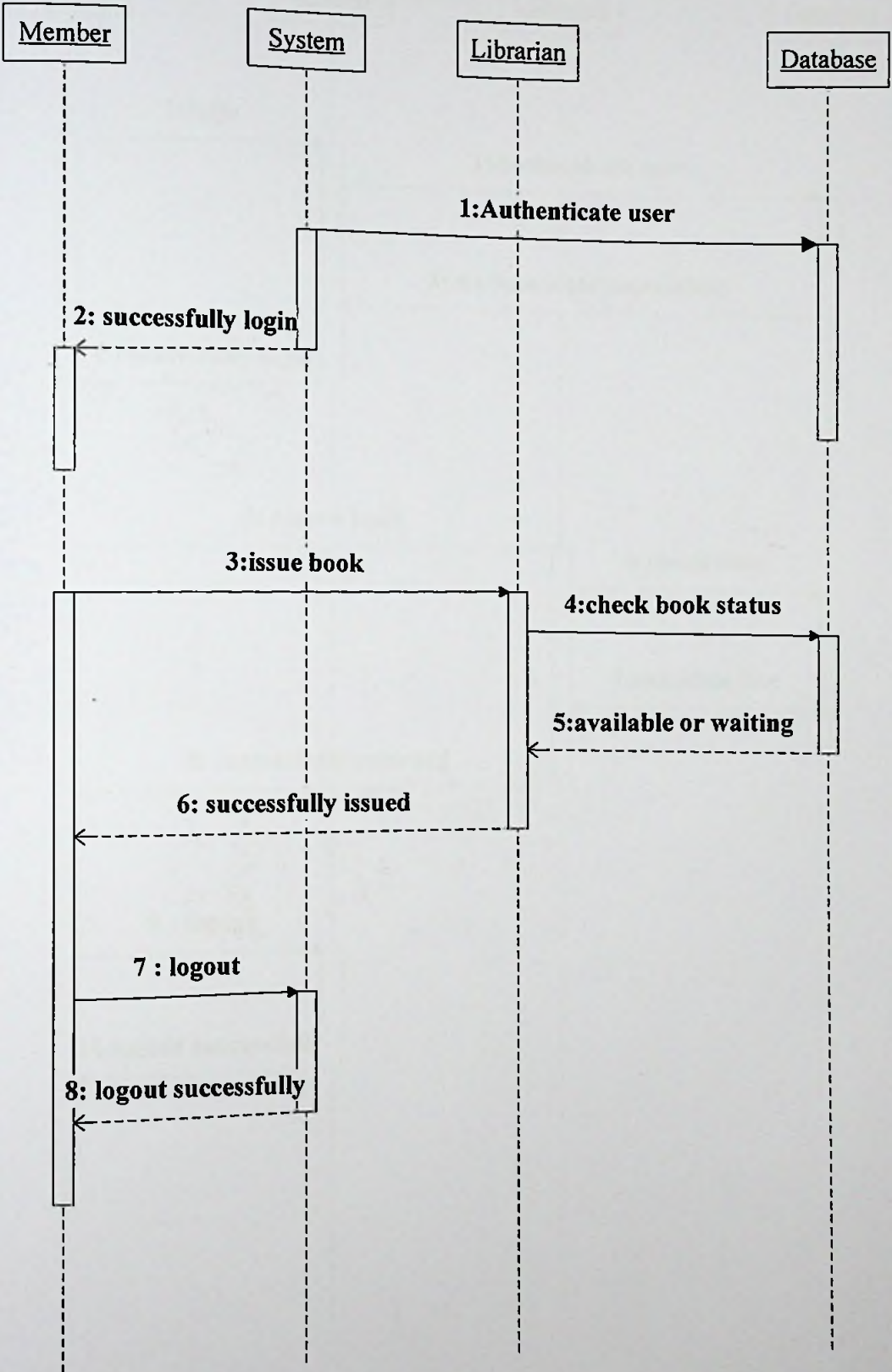


Figure C 2 : Sequence diagram for returning book:

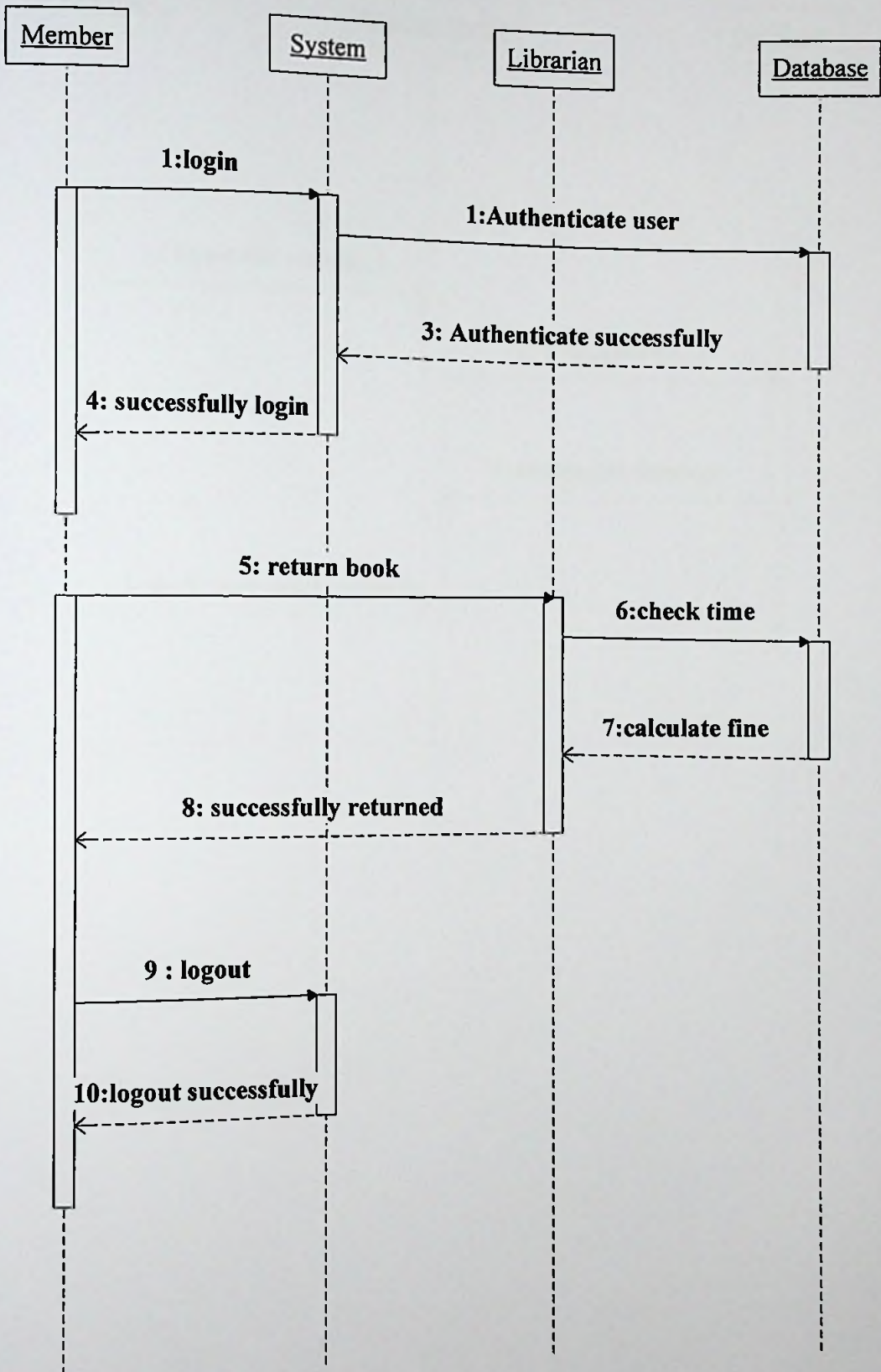
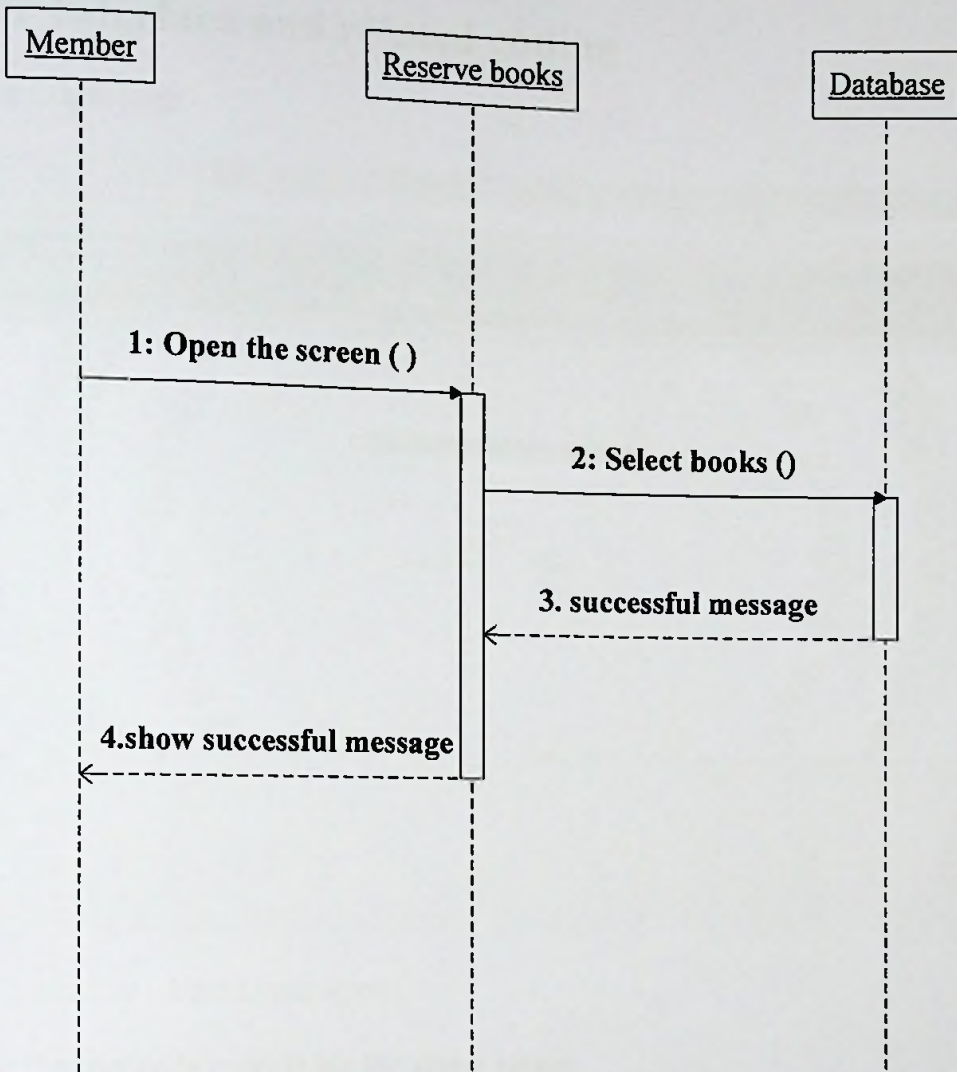


Figure C 3 : Sequence diagram for reserving book:



User Interface and related coding

1. User Login page

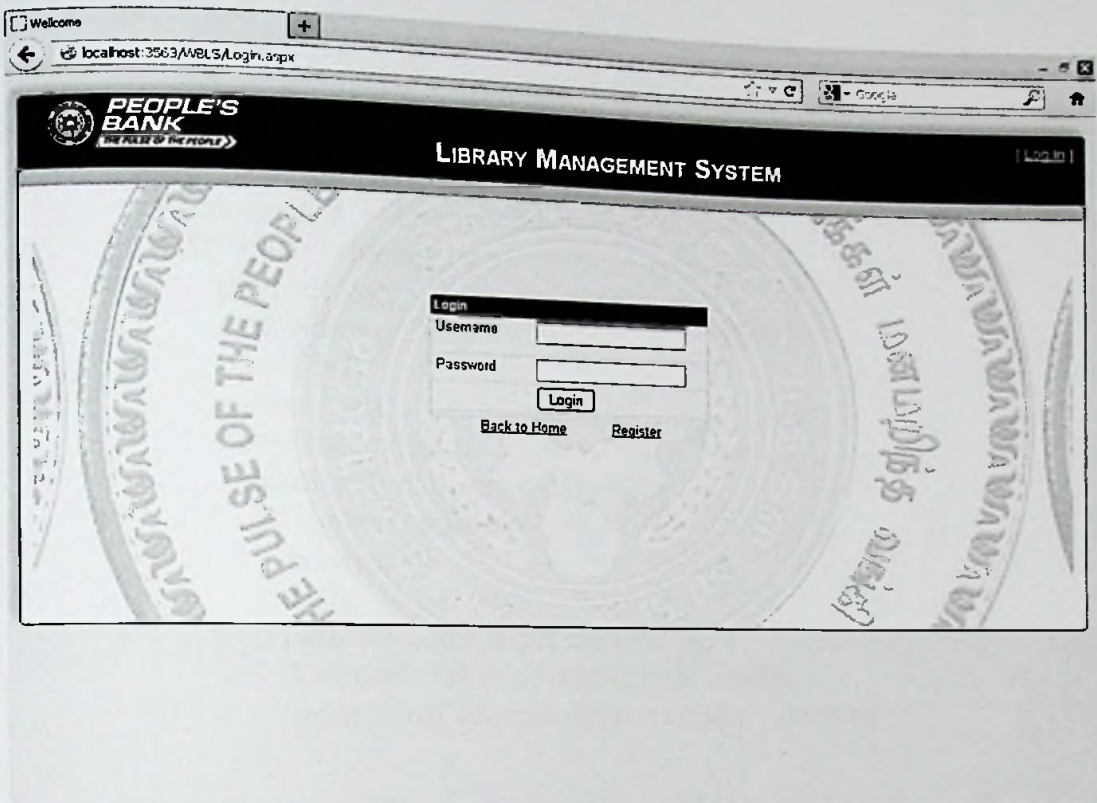


Figure D1: User Login screen

Source code related for the above screen

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Login.aspx.cs"
Inherits="Login" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <link href="Styles/Site.css" rel="stylesheet" type="text/css" />
  <title>Wellcome</title>
  <script src="Scripts/js/jquery-1.7.2.min.js"
type="text/javascript"></script>
  <script src="Scripts/js/jquery-ui-1.8.22.custom.min.js"
type="text/javascript"></script>
  <script src="Scripts/js/slides.min.jquery.js" type="text/javascript"></script>
  <script src="Scripts/jcarouselite_1.0.1c4.js" type="text/javascript"></script>
  <script type="text/javascript">
$(function () {
```



```

$( "#newsticker-demo" ).jCarouselLite({
    vertical: true,
    hoverPause: true,
    btnPrev: ".previous",
    btnNext: ".next",
    visible: 3,
    auto: 2000,
    speed: 500
});
});
</script>
<script type="text/javascript">
$(function () {
    $('#slides').slides({
        preload: true,
        preloadImage: 'img/loading.gif',
        play: 5000,
        pause: 2500,
        hoverPause: true,
        animationStart: function (current) {
            $('.caption').animate({
                bottom: -35
            }, 100);
            if (window.console && console.log) {
                // example return of current slide number
                console.log('animationStart on slide: ', current);
            };
        },
        animationComplete: function (current) {
            $('.caption').animate({
                bottom: 0
            }, 200);
            if (window.console && console.log) {
                // example return of current slide number
                console.log('animationComplete on slide: ', current);
            };
        },
        slidesLoaded: function () {
            $('.caption').animate({
                bottom: 0
            }, 200);
        }
    });
});
</script>
<style type="text/css">
.style1
{
    color: #FFFFFFF;
    font-weight: bold;

```


2. User interface for add books

Figure D 2 add books

Source code related for the above screen

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.master"
AutoEventWireup="true" CodeFile="AddBook.aspx.cs" Inherits="AddBook" %>

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent"
Runat="Server">
  <script type="text/javascript">
    $(function () {
      $('#<%=txtPublishDate.ClientID %>').datepicker({ dateFormat: "yy/mm/dd"
    }).val()
    });
  </script>
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent"
Runat="Server">
  <div class="mid">
    <asp:Panel ID="Panell" runat="server" BackColor="#E4E4E4" Font-
Bold="True"
HorizontalAlign="Center">
```




```

<asp:RadioButtonList ID="rblVersion" runat="server" AutoPostBack="True"
RepeatDirection="Horizontal" Width="300px"
onselectedindexchanged="RadioButtonList1_SelectedIndexChanged">
  <asp:ListItem Value ="NEW_BOOK" Selected="True">New
Book</asp:ListItem>
  <asp:ListItem Value ="NEW_VERSION">New Version</asp:ListItem>
</asp:RadioButtonList>
</asp:Panel>
  <asp:Panel ID="PanelNew" runat="server">
<table style="width: 100%">
  <tr>
    <th style="width: 100px">
      &nbsp;
    </th>
    <th style="width: 10px">
      &nbsp;
    </th>
    <th style="width: 130px" >
      &nbsp;
    </th>
    <th style="width: 10px">
      &nbsp;
    </th>
    <th style="width: 100px">
      &nbsp;
    </th>
    <th style="width: 10px">
      &nbsp;
    </th>
    <th >
      &nbsp;
    </th>
  </tr>
  <tr>
    <td>
      Title</td>
    <td>
      &nbsp;:
    </td>
    <td colspan="5">
      <asp:TextBox ID="txtTitle" runat="server" Width="255px"
MaxLength="60"></asp:TextBox>
      <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
ControlToValidate="txtTitle" ErrorMessage="*"
ForeColor="Red"></asp:RequiredFieldValidator>
    </td>
  </tr>
  <tr>
    <td>
      Author</td>

```



```

<td>
  &nbsp;:
</td>
<td colspan="5">
  <asp:TextBox ID="txtAuthor" runat="server" Width="255px"
MaxLength="40"></asp:TextBox>
  <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
ControlToValidate="txtAuthor" ErrorMessage="*"
ForeColor="Red"></asp:RequiredFieldValidator>
</td>
</tr>
<tr>
  <td>
    Category
  </td>
  <td>
    &nbsp;:
  </td>
  <td>
    <asp:DropDownList ID="ddlCategory" runat="server" Width="120px">
</asp:DropDownList>
  </td>
  <td>
  </td>
  <td>
    ISBN
  </td>
  <td>
    &nbsp;:
  </td>
  <td>
    <asp:TextBox ID="txtISBN" runat="server" MaxLength="40"
Width="120px"></asp:TextBox>
    <asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server"
ControlToValidate="txtISBN" ErrorMessage="*"
ForeColor="Red"></asp:RequiredFieldValidator>
  </td>
</tr>
<tr>
  <td>
    Publisher
  </td>
  <td>
    &nbsp;:
  </td>
  <td>
    <asp:TextBox ID="txtPublisher" runat="server" MaxLength="30"

```

```

Width="120px"></asp:TextBox>
</td>
<td>
    &nbsp;
</td>
<td>
    Publish Date</td>
<td>
    &nbsp;:
</td>
<td>
    <asp:TextBox ID="txtPublishDate" runat="server" MaxLength="10"
Width="120px"></asp:TextBox> <asp:RequiredFieldValidator
ID="RequiredFieldValidator4" runat="server"
ControlToValidate="txtPublishDate" ErrorMessage="*"
ForeColor="Red"></asp:RequiredFieldValidator>
    </td>
</tr>
<tr>
<td>
    Is Borrowable ?</td>
<td>
    &nbsp;:
</td>
<td>
    <asp:RadioButtonList ID="rblIsBorrowable" runat="server"
RepeatDirection="Horizontal">
    <asp:ListItem Text="Yes" Selected="True" Value="1"></asp:ListItem>
    <asp:ListItem Text="No" Selected="False" Value="0"></asp:ListItem>
    </asp:RadioButtonList>
</td>
<td>
    &nbsp;</td>
<td>
    Version </td>
<td>
    &nbsp;:</td>
<td>
    <asp:TextBox ID="txtVersion" runat="server"
Width="120px"
    ></asp:TextBox>
    <asp:RequiredFieldValidator ID="RequiredFieldValidator5"
runat="server"
ControlToValidate="txtVersion" ErrorMessage="*"
ForeColor="Red"></asp:RequiredFieldValidator>
    <asp:RangeValidator ID="RangeValidator1" runat="server"
ControlToValidate="txtVersion" ErrorMessage="?" ForeColor="Red"
MaximumValue="99" MinimumValue="1"
Type="Integer"></asp:RangeValidator>
    </td>
</tr>

```

```

<tr>
  <td>
    Price
  </td>
  <td>
    &nbsp;   :
  </td>
  <td colspan="5">
    <asp:TextBox ID="txtPrice" runat="server" MaxLength="8"
Width="120px"></asp:TextBox>
    <asp:RequiredFieldValidator ID="RequiredFieldValidator6" runat="server"
ControlToValidate="txtPrice" ErrorMessage="*"
ForeColor="Red"></asp:RequiredFieldValidator>
    <asp:RangeValidator ID="RangeValidator2" runat="server"
ControlToValidate="txtPrice" ErrorMessage="?" ForeColor="Red"
Type="Currency" MaximumValue="9999999999"
MinimumValue="0"></asp:RangeValidator>
  </td>
</tr>
<tr>
  <td>
    Description
  </td>
  <td>
    &nbsp;   :
  </td>
  <td colspan="5">
    <asp:TextBox ID="txtDescription" runat="server"
MaxLength="50"
Rows="2" TextMode="MultiLine" Width="255px"></asp:TextBox>
  </td>
</tr>
<tr>
  <td>
    Select Image
  </td>
  <td>
    &nbsp;   :
  </td>
  <td colspan="5">
    <asp:FileUpload ID="FileUploadSelectImage" runat="server"
Width="255px" />
  </td>
</tr>
<tr>
  <td>
    Book Value</td>
  <td>
    &nbsp;   :

```



```

</td>
<td>
    <asp:DropDownList ID="ddlBookValueType" runat="server"
Width="120px">
    </asp:DropDownList>
    </td>
    <td>
        &nbsp;&nbsp;&nbsp;</td>
<td>
    &nbsp;&nbsp;&nbsp;
</td>
<td>
    &nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
    <td>
        Is Recommend?</td>
    <td>
        &nbsp;&nbsp;&nbsp;:
    </td>
    <td>
        <asp:RadioButtonList ID="rdblIsRecommended" runat="server"
RepeatDirection="Horizontal">
        <asp:ListItem Text="No" Selected="True" Value="0"></asp:ListItem>
        <asp:ListItem Text="Yes" Selected="False" Value="1"></asp:ListItem>
        </asp:RadioButtonList>
    </td>
    <td>
        &nbsp;&nbsp;&nbsp;</td>
    <td>
        </td>
    <td>
        &nbsp;&nbsp;&nbsp;</td>
    <td>
        </td>
    <td>
        </td>
</tr>
<tr>
    <td>
        &nbsp;&nbsp;&nbsp;</td>
    <td>
        &nbsp;&nbsp;&nbsp;
    </td>
    <td>
        &nbsp;&nbsp;&nbsp;
    </td>
    <td>
        <asp:Button ID="btnAdd" runat="server" Text="Add"
onclick="btnAdd_Click" />
    </td>
    <td>
        &nbsp;&nbsp;&nbsp;</td>
    <td>
        &nbsp;&nbsp;&nbsp;
    </td>

```



```

</td>
<td>
    &nbsp;  </td>
</tr>
<tr>
<td></td><td></td>
<td colspan="5">
    <asp:Label ID="lblMessage" runat="server"></asp:Label>
</td>
</tr>
</table>
<asp:Image ID="Image1" runat="server" ImageUrl=""
    Visible="False" />
</asp:Panel>
</div>
</asp:Content>

```

3. User interface for add branch

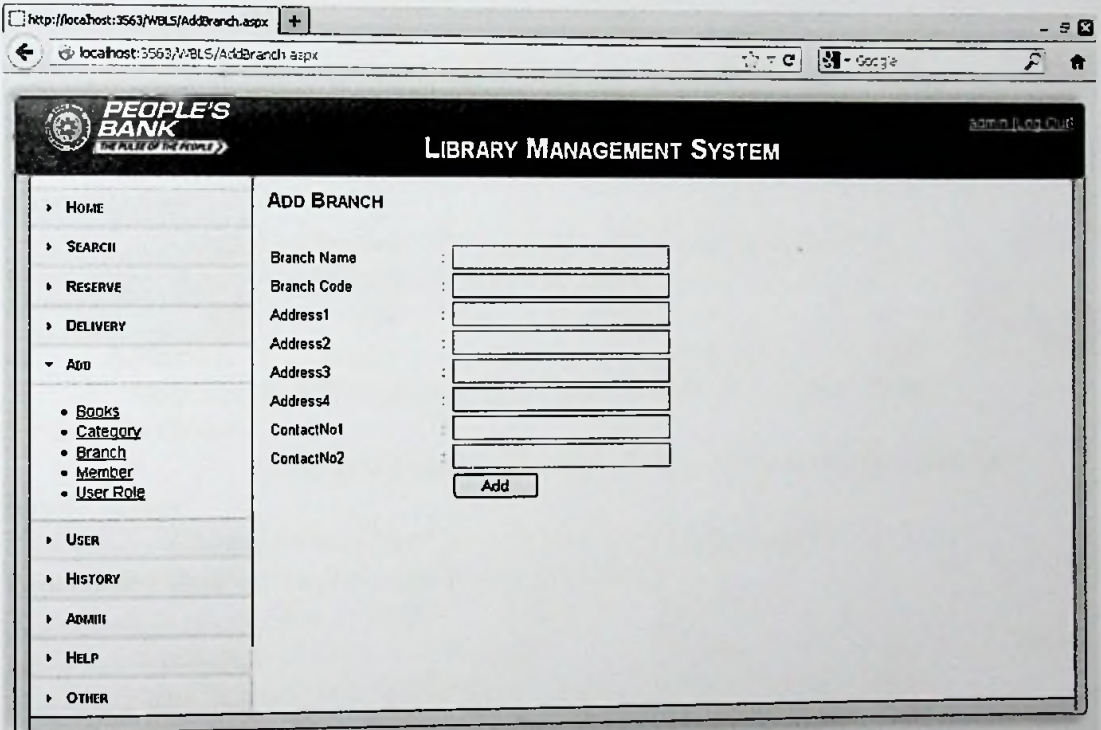
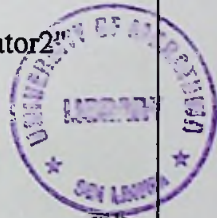


Figure D 3: Add Branch screen

Source code related for the above screen

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.master"
AutoEventWireup="true" CodeFile="AddBranch.aspx.cs" Inherits="AddBranch" %>
<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent"
Runat="Server">
</asp:Content><asp:Content ID="Content2" ContentPlaceHolderID="MainContent"
Runat="Server">
<h2>Add Branch</h2>
<div class="mid"> <table style="width: 100%">
<tr>
<th style="width: 150px">
</th>
<th style="width: 10px">
</th>
<th >
</th>
<th >
</th>
<tr>
<td>
Branch Name
</td>
<td>
<asp:TextBox ID="txtBranchName" runat="server"
Width="200px"></asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidator1"
runat="server"
ControlToValidate="txtBranchName" ErrorMessage="*"
ForeColor="Red"></asp:RequiredFieldValidator>
</td>
<td>
Branch Code
</td>
<td>
<asp:TextBox ID="txtBranchCode" runat="server" Width="200px"
MaxLength="3"></asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidator2"
runat="server"
ControlToValidate="txtBranchCode" ErrorMessage="*"
ForeColor="Red"></asp:RequiredFieldValidator>
</td>
<td>
Address1
</td>
<td>
<asp:TextBox ID="txtAddress1" runat="server" Width="200px"
></asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidator3"
runat="server"
ControlToValidate="txtAddress1" ErrorMessage="*"
ForeColor="Red"></asp:RequiredFieldValidator>
</td>
<td>
Address2
</td>
<td>
<asp:TextBox ID="txtAddress2" runat="server" Width="200px"
></asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidator5"
runat="server"
ControlToValidate="txtAddress2" ErrorMessage="*"
ForeColor="Red"></asp:RequiredFieldValidator>
</td>
<td>
Address3
</td>
<td>
<asp:TextBox ID="txtAddress3" runat="server" Width="200px"
></asp:TextBox>
</td>
<td>
Address4
</td>
</tr>
</table>
</div>
```



```

<asp:TextBox ID="txtAddress4" runat="server" Width="200px"
></asp:TextBox>
</td> </tr> <tr> <td>
ContactNo1 </td> <td> &nbsp;: </td> <td>
<asp:TextBox ID="txtContactNo1" runat="server" Width="200px"
></asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidator4"
runat="server"
ControlToValidate="txtContactNo1" ErrorMessage="*"
ForeColor="Red"></asp:RequiredFieldValidator>
</td> </tr> <tr> <td>
ContactNo2</td> <td> &nbsp;: </td> <td>
<asp:TextBox ID="txtContactNo2" runat="server" Width="200px"
></asp:TextBox>
</td> </tr> <tr> <td> </td> <td> </td> <td>
<asp:Button ID="btnAdd" runat="server" Text="Add" Width="80px"
onclick="btnAdd_Click" />
</td> </tr> <tr> <td> </td> <td> &nbsp;: </td> <td>
<asp:Label ID="lblMessage" runat="server" Text=""></asp:Label>
</td> </tr> </table> </div>
</asp:Content>

```

4. User interface for add member

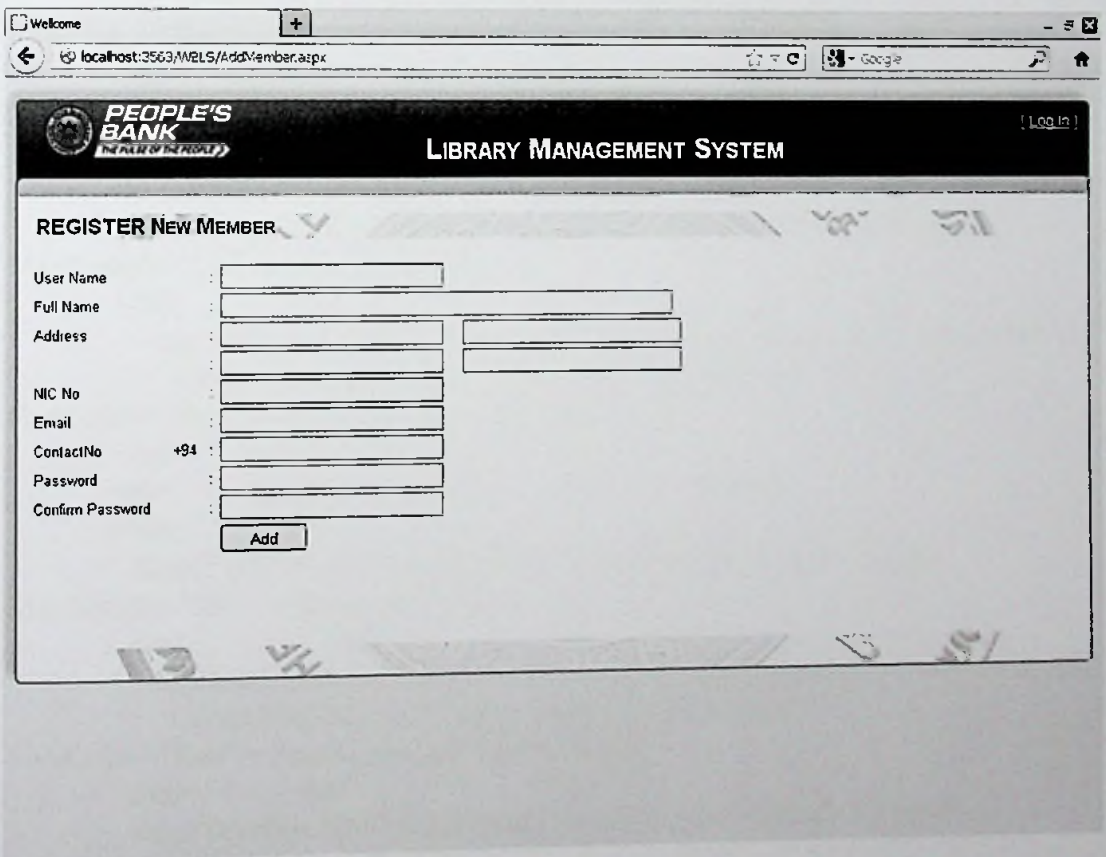


Figure D 4: Add Member screen

Source code related for the above screen

```

<%@ Page Title="" Language="C#" MasterPageFile="~/CommonUsers.master"
AutoEventWireup="true" CodeFile="AddMember.aspx.cs" Inherits="AddMember"
%>
<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <h2>&nbsp;&nbsp;&nbsp;REGISTER New Member </h2>
    <div class="mid" >
        <table style="width: 100%; background-color: White; " >
            <tr>
                <th style="width: 150px"> &nbsp;&nbsp;&nbsp; </th>
                <th style="width: 10px"> &nbsp;&nbsp;&nbsp; </th>
                <th style="width: 200px" > &nbsp;&nbsp;&nbsp; </th>
                <th style="width: 10px"> &nbsp;&nbsp;&nbsp; </th>
                <th style="width: 200px" > &nbsp;&nbsp;&nbsp; </th>
            <tr>
                <td> &nbsp;&nbsp;&nbsp; </td>
                <td> &nbsp;&nbsp;&nbsp; </td>
                <td> &nbsp;&nbsp;&nbsp; </td>
                <td> &nbsp;&nbsp;&nbsp; </td>
                <td> &nbsp;&nbsp;&nbsp; </td>
            <tr>
                <td> User Name </td>
                <td> </td>
                <td> &nbsp;&nbsp;&nbsp; </td>
                <td> </td>
                <td> </td>
            <tr>
                <td> <asp:TextBox ID="txtUserName" runat="server" Width="200px"
MaxLength="50"></asp:TextBox>
                <td> </td>
                <td> </td>
                <td> </td>
                <td> </td>
            <tr>
                <td> </td>
                <td> </td>
                <td> </td>
                <td> </td>
                <td> </td>
            <tr>
                <td> <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
ControlToValidate="txtUserName" ErrorMessage="*"
ForeColor="Red"></asp:RequiredFieldValidator>
                <td> </td>
                <td> </td>
                <td> </td>
                <td> </td>
                <td> </td>
            <tr>
                <td> Full Name </td>
                <td> </td>
                <td> &nbsp;&nbsp;&nbsp; </td>
                <td> </td>
                <td> </td>
            <tr>
                <td colspan="3">
                    <asp:TextBox ID="txtFullName" runat="server" Width="414px"
MaxLength="120"></asp:TextBox>
                <td> </td>
                <td> </td>
                <td> </td>
                <td> </td>
            <tr>
                <td> <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
ControlToValidate="txtFullName" ErrorMessage="*"
ForeColor="Red"></asp:RequiredFieldValidator>
                <td> </td>
                <td> </td>
                <td> </td>
                <td> </td>
                <td> </td>
            <tr>
                <td> Address </td>
                <td> </td>
                <td> &nbsp;&nbsp;&nbsp; </td>
                <td> </td>
                <td> </td>
            <tr>
                <td> <asp:TextBox ID="txtAddress1" runat="server" Width="200px"
MaxLength="50" ></asp:TextBox>
                <td> </td>
                <td> </td>
                <td> </td>
                <td> </td>
            <tr>
                <td> <asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server"
ControlToValidate="txtAddress1" ErrorMessage="*"
ForeColor="Red"></asp:RequiredFieldValidator>
                <td> </td>
                <td> </td>
                <td> </td>
                <td> </td>
                <td> </td>
            <tr>
                <td> <asp:TextBox ID="txtAddress2" runat="server" Width="200px"
MaxLength="50" ></asp:TextBox>
                <td> </td>
                <td> </td>
                <td> </td>
                <td> </td>
            <tr>
                <td> <asp:RequiredFieldValidator ID="RequiredFieldValidator5" runat="server"

```



```

        <asp:RegularExpressionValidator ID="RegularExpressionValidator2"
runat="server"
        ControlToValidate="txtContactNo1" ErrorMessage="Invalid number"
ForeColor="Red"
        ValidationExpression="\d{9}"></asp:RegularExpressionValidator>
    </td>
</tr>
<tr>
    <td>
        Password</td>
    <td>
        &nbsp;:;
    </td>
    <td>
        <asp:TextBox ID="txtPassword" runat="server" Width="200px"
MaxLength="30"
        TextMode="Password"></asp:TextBox>
    </td>
    <td>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator6" runat="server"
ControlToValidate="txtPassword" ErrorMessage="*"
ForeColor="Red"></asp:RequiredFieldValidator>
    </td>
    <td>&nbsp; </td>
</tr>
<tr>
    <td> Confirm Password</td>
    <td>
        &nbsp;:;
    </td>
    <td>
        <asp:TextBox ID="txtRePassword" runat="server" Width="200px"
MaxLength="30"
        TextMode="Password"></asp:TextBox>
    </td>
    <td>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator8" runat="server"
ControlToValidate="txtRePassword" ErrorMessage="*"
ForeColor="Red"></asp:RequiredFieldValidator>
    </td>
    <td>
        <asp:CompareValidator ID="CompareValidator1" runat="server"
ControlToCompare="txtPassword" ControlToValidate="txtRePassword"
ErrorMessage="Password does not match"
ForeColor="Red"></asp:CompareValidator>
    </td>
    <td></td>
</tr>
<tr>
    <td> </td>
    <td> </td>
    <td>
        <asp:Button ID="btnAdd" runat="server" Text="Add" Width="80px"
onclick="btnAdd_Click" style="height: 26px" />
    </td>
</tr>
<tr>
    <td colspan="3"> <asp:Label ID="lblMessage" runat="server"
Text=""></asp:Label>
    </td>
</tr>
</table> <table style="width: 100%; background-color:White;"><tr>
    <td> &nbsp; </td>
    <td>
        &nbsp; </td>
    <td>
        &nbsp; </td>
    <td>
        &nbsp; </td>
</tr>

```

```

&nbsp;</td></tr> <tr> <td>
&nbsp;</td><td>
&nbsp;</td><td>
&nbsp;</td></tr><tr><td>
&nbsp;</td><td>
&nbsp;</td> </td> <td>
&nbsp;</td> </tr>
</table>
</div>
</asp:Content>

```

Pseudo codes - for search a book

“Select the Book button

 Select the search by any

 Display all books

 If required book exists

 Display the message for availability

 Else

 Display the message for non-availability

 End if

 Select the search by author and enter author name Into Database

 Search for this book

 If related author book exists

 Display the message for availability

 Else

 Display the message for non-availability

 End if

 Move to Next search by Title and enter title name Into Database

 Search for this book

 If exit related Title name book

 Display the message for availability

 Else

 Display the message for non-availability

 End if

 Move to Next search by Publisher and enter publisher name Into Database

 Search for this book

 If exit related Publisher book


```

        Display the message for availability
    Else
        Display the message for non-availability
    End if
Move to next search by Publish date and enter publish date Into Database
    Search for this book
        If exit related Publish date book
            Display the message for availability
        Else
            Display the message for non-availability
        End if
    End"

```

Source code related for the above pseudo codes - for search a book

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;

public partial class BookSearch : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
        }
        else
        {
        }
        SetMostViewedBook();
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        ClearControls();
        GetSearchResultNew();
    }
    public void GetSearchResult()
    {
        string searchWord = "prog";

```



```

string searchType = "1";
Book books = new Book();
DataTable dt;
DataTable dtRecomended = new DataTable();
switch (searchType)
{
    case "Author":
        dt = books.SearchByAuthor(searchWord);
        dtRecomended = books.SearchByAuthorPopular(searchWord);
        break;
    case "Title":
        dt = books.SearchByTitle(searchWord);
        dtRecomended = books.SearchByTitlePopular(searchWord);
        break;
    case "PublishDate":
        dt = books.SearchByPublishDate(searchWord);
        dtRecomended = books.SearchByPublishDatePopular(searchWord);
        break;
    case "Publisher":
        dt = books.SearchByPublisher(searchWord);
        dtRecomended = books.SearchByPublisherPopular(searchWord);
        break;
    default:
        dt = books.SearchByAll(searchWord);
        dtRecomended = books.SearchByAllPopular(searchWord);
        break;
}
ListView1.DataSource = dtRecomended;
ListView1.DataBind();
}
public void GetSearchResultNew()
{
    string searchWord = txtSearch.Text.Trim();
    string searchType = ddlSeachType.SelectedValue;
    Book books = new Book();
    DataTable dt;
    DataTable dtRecomended = new DataTable();
    switch (searchType)
    {
        case "Author":
            dt = books.SearchByAuthor(searchWord);
            dtRecomended = books.SearchByAuthorPopularNew(searchWord);
            break;
        case "Title":
            dt = books.SearchByTitle(searchWord);
            dtRecomended = books.SearchByTitlePopularNew(searchWord);
            break;
        case "PublishDate":
            dt = books.SearchByPublishDate(searchWord);
            dtRecomended = books.SearchByPublishDatePopularNew(searchWord);
    }
}

```

```

        break;
    case "Publisher":
        dt = books.SearchByPublisher(searchWord);
        dtRecomended = books.SearchByPublisherPopularNew(searchWord);
        break;
    default:
        dt = books.SearchByAll(searchWord);
        dtRecomended = books.SearchByAllPopularNew(searchWord);
        break;
    }
    ListView1.DataSource = dt;
    ListView1.DataBind();
    if (dt.Rows.Count == 0)
    {
        lblMessage.Text = "No search result found";
        lblMessage.Visible = true;
    }
    else
    {
        lblMessage.Text = "";
        lblMessage.Visible = false;
    }
}
public void ClearControls()
{
}
protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
{
}
protected void GridView1_RowCreated(object sender, GridViewRowEventArgs e)
{
    if (e.Row.RowType == DataControlRowType.Header)
    {
        e.Row.Cells[0].Visible = false;
    }
    if (e.Row.RowType == DataControlRowType.DataRow)
    {
        e.Row.Cells[0].Visible = false;
    }
}
protected void GridView1_RowCommand(object sender,
GridViewCommandEventArgs e)
{
}
protected void ListView1_ItemCommand1(object sender,
ListViewCommandEventArgs e)
{
    // Check the command requested
    if (e.CommandName == "MoreDetailsPlease")
    {

```

```

HiddenField id = (HiddenField)e.Item.FindControl("hdnBookId");
if (Session["LastViewed"] == null)
{
    Session["LastViewed"] = id.Value;
}
else
{
    int lastViewBookId = Convert.ToInt32(Session["LastViewed"].ToString());
    int currentBookId = Convert.ToInt32(id.Value);
    if (lastViewBookId != currentBookId)
    {
        Book books = new Book();
        Session["LastViewed"]
=books.EntryOperationWhoViewedThisItemAlsoViewed(lastViewBookId,
currentBookId);
    }
}
Response.Redirect("BookDetail.aspx?bookId="+id.Value+"&&searchKey="+txtSear
ch.Text+""");
}
}
public void SetMostViewedBook()
{
    Book book = new Book();
    book.LoadMostViewedBook();
    ImgBtnMostViewd.ImageUrl = book.ImageURL;
    lnkBtnMostViewdTitle.Text = book.Title;
    hdnMostViewdBookId.Value = book.ID.ToString();
}
protected void lnkBtnMostViewdTitle_Click(object sender, EventArgs e)
{
    string bookId = hdnMostViewdBookId.Value;
    Response.Redirect("BookDetail.aspx?bookId=" + bookId + "&&searchKey=" +
txtSearch.Text + "");
}
protected void ImgBtnMostViewd_Click(object sender, ImageClickEventArgs e)
{
}
protected void ListView1_PagePropertiesChanging(object sender,
PagePropertiesChangingEventArgs e)
{
    lvDataPager1.SetPageProperties(e.StartRowIndex, e.MaximumRows, false);
    GetSearchResultNew();
}
}

```


Pseudo codes - for reserve a book

“Select the Book button

Click Go to search page

Search for required book

Find Book ID

Enter book ID in reserve menu

If book has not reserved

Display the message for availability

Else

Display the message for non - availability

End if

If already reserved two books

Display the message for non - availability

Else

Display the message for availability

End if

End”

“Select the Approval screen

Display the books which are to be approve

Select approve or reject button on reserve list

If book approved

Display the message for approved

Else

If book reject

Display the message for reject

End if

End if

End”

Source code related for the above pseudo codes - for reserve a book

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Web;
```

```

using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;

public partial class ReserveBook : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void btnReserve_Click(object sender, EventArgs e)
    {
        ClearControl();

        Reservation reserve = new Reservation();
        Book book = new Book();

        reserve.BookID = Convert.ToInt32(txtBookId.Text.Trim());
        reserve.MemberID = Convert.ToInt32(Session["USER_ID"].ToString());
        reserve.ReserveDate = System.DateTime.Now;
        reserve.ReserveStatusID = 1;
        reserve.Comment = "Reserved by search";

        book.ID = reserve.BookID;
        if (book.IsExistBookID())
        {
            DataTable reservedList = reserve.GetBookStatus();
            if (reservedList.Rows.Count > 0)
            {
                lblMessage.Text = "This book has following status";
                GridView1.DataSource = reservedList;
                GridView1.DataBind();
            }

            if (reserve.IsReserve())
            {
                lblMessage.Text = "You Have Already reserved this book. Reserved Date : "
+ reserve.ReserveDate.ToString("yyyy/MM/dd");
            }
            else
            {
                if (reserve.IsExceedReservationLimitForMember())
                {
                    lblMessage.Text = "You Have Already reserved 2 books. ";
                }
                else
                {
                    reserve.Reserve();
                    lblMessage.Text = "Book Reserved";
                }
            }
        }
    }
}

```

```

    }
}
else
{
    lblMessage.Text = "No book find for given ID";
}
}

```

```

public void ClearControl()
{
    lblMessage.Text = "";
    GridView1.DataSource = null;
    GridView1.DataBind();
}
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

```

```

public partial class ReserveRejectReason : System.Web.UI.Page
{
    int reserveID = 0;
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Request.QueryString["reserveID"] != null) // this will ensure that below line
        will only fire if querystring exists in the url
        {
            reserveID = Convert.ToInt32( Request.QueryString["reserveID"].ToString());
            ViewReservationDetails();
        }
        else
        {
            Response.Redirect("~/ReserveView.aspx");
        }
    }

    protected void btnAdd_Click(object sender, EventArgs e)
    {
        btnAdd.Enabled = true;
    }
}

```



```

AddHistory( reserveID
=Convert.ToInt32(lblReservationID.Text.Trim().ToString()));
}

public void ViewReservationDetails()
{
Reservation reservation = new Reservation();
reservation.ReserveID = reserveID;

reservation.FetchByID();
lblReservationID.Text= reservation.ReserveID.ToString();
hplMemberName.Text = reservation.MemberID.ToString();
lblReserveDate.Text = reservation.ReserveDate.ToString();
lblReservationStatus.Text = reservation.ReserveStatusID.ToString();
lblComment.Text = reservation.Comment;

Book book = new Book();
book.ID = Convert.ToInt32(reservation.BookID.ToString());
book.FetchByID();

lblBookId.Text = book.ID.ToString();
lblReferenceID.Text = book.ReferenceTypeID;
imgBook.ImageUrl = book.ImageURL;

Member member = new Member();
member.MemberID = reservation.MemberID;
member.FetchByID();
hplMemberName.Text = member.FullName + " [" +
reservation.MemberID.ToString() + "];
}

public void AddHistory(int reserseID)
{
RejectReservation rejectReservation = new RejectReservation();
rejectReservation.ReserveID = reserseID;
rejectReservation.ReasonForReject = txtReason.Text;
rejectReservation.DeliverBranchID = 1;
rejectReservation.DestinationBranchID = 1;

if (rejectReservation.Add() > 0)
{
Reservation reservation = new Reservation();
reservation.ReserveID = reserseID;
reservation.DeleteByID();
lblMessage.Text = "Reserve ID : " + reserseID + " successfully rejected ";
btnAdd.Enabled = false;
}
else
{

```



```

        lblMessage.Text = "Reserve ID :" + reserseID + " Rejection unsuccessful
please try again ";
    }

    #region MyRegion not used
    //if (rejectReservation.Add() > 0)
    //{
    //    Reservation reservation = new Reservation();
    //    reservation.ReserveID = reserseID;
    //    reservation.FetchByID();
    //    rejectReservation.BookID = reservation.BookID;
    //    rejectReservation.MemberID = reservation.MemberID;
    //    rejectReservation.ReserveDate = reservation.ReserveDate;
    //    rejectReservation.ReservationAcceptDate = System.DateTime.Now;
    //    rejectReservation.ReserveComment = reservation.Comment;
    //    rejectReservation.DeliverBranchID = 1;////////////////////////////////////
    //    rejectReservation.DestinationBranchID = 2;////////////////////////////////////
    //    rejectReservation.ReserveID = reserseID;
    //    rejectReservation.ReasonForReject = txtReason.Text;
    //}
    #endregion
}
}

```

Test Cases

1 Test case for user login authentication

Test case No.	Description	Expected out put	System created output	Status
01	User login with valid user name and password	Display the home page	Display the home page	Success
02	Enter valid user name and invalid password	"Invalid password" message prompt for the user login again	"Invalid password" message prompt for the user login again	Success
03	Enter invalid user name and valid password	"Invalid user name" message prompt for the user login again.	"Invalid user name" message prompt for the user login again.	Success
04	Enter invalid user name and invalid password	"Invalid user name" message prompt for the user login again	"Invalid user name" message prompt for the user login again	Success
05	Press the login button without any values	"Invalid user name" message prompt for the user login again	"Invalid user name" message prompt for the user login again	Success

Figure E 1 Test cases for User login authentication

1 Test cases for reserve books

Test case No.	Description	Expected out put	System created output	Status
01	Select a book	Display the selected book	Display the selected book	Success
02	Click on reserve	Display the message	Display the message	Success
03	Enter valid book ID	Display the message	Display the message	Success
04	Enter invalid book ID	"No book find for given ID" message prompt for the user	"No book find for given ID" message prompt for the user	Success
05	Press the reserve button without any value	No message display	No message display	Success

Figure E 2 Test cases for Reserve books

3 Test case for search books

Test case No.	Description	Expected out put	System created output	Status
01	Select the search criteria by any and click the search button	Display the all books	Display the all books	Success
02	Select the search criteria by author and enter the name	Display the relevant books	Display the relevant books	Success
03	Select the search criteria by author and enter the invalid name	"No search results found" message prompt	"No search results found" message prompt	Success
04	Select the search criteria by title and enter the name	Display the relevant books	Display the relevant books	Success
05	Select the search criteria by title and enter the invalid name	"No search results found" message prompt	"No search results found" message prompt	Success

Figure E 3 Test cases for Search books

4 Test cases for change password

Test case No.	Description	Expected out put	System created output	Status
01	Insert valid old password, new password and confirm password	“Successfully changed” message prompt	“Successfully changed” message prompt	Success
02	Insert invalid old password, new password and confirm password	“Invalid password” message prompt	“Invalid password” message prompt	Success
03	Insert valid old password, new password and invalid confirm password	“Invalid password” message prompt	“Invalid password” message prompt	Success
04	Insert old password empty new password and invalid confirm password	“Invalid password” message prompt	“Invalid password” message prompt	Success
05	At least old password, new password or confirm password is empty	“Invalid password” message prompt	“Invalid password” message prompt	Success

Figure E 4 Test cases for Change password

5 Test case for add book

Test case No.	Description	Expected out put	System created output	Status
01	Insert the valid DATE format	Not to show any message	Not to show any message	Success
02	Insert the invalid DATE format	“Error...Please check date format” message prompt	“Error...Please check date format” message prompt	Success
03	Keep the empty DATE, ISBN, Version fields	Display “*”mark on left	Display “*”mark on left	Success

Figure E 5 Test cases for add book

