DURBAN UNIVERSITY OF TECHNOLOGY

FACULTY OF ACCOUNTING AND INFORMATICS

DEPARTMENT OF INFORMATION TECHNOLOGY

2014

MID-YEAR MAIN EXAMINATION

INSTRUCTIONAL PROGRAMME :	ND: INFORMATION TECHNOLOGY (NDINBI & NDINSI)
INSTRUCTIONAL OFFERING :	DEVELOPMENT SOFTWARE 2 (MODULE 1)
PAPER NUMBER :	1
SUBJECT CODE/S :	DSFW212
DATE :	9 JUNE 2014
DURATION :	2 HOURS
TIME :	14:00 - 16:00
TOTAL MARKS :	70
NUMBER OF PAGES :	10 (EXCLUDING COVER)
EXAMINER/S :	C BOAMAH-ABU AND N S M SHONGWE
MODERATOR/S :	S FOOLCHAND

INSTRUCTIONS/REQUIREMENTS:-

[1] Answer all questions and number your work as per question paper.

[2] Write your student number, surname and initials and the group you belong to on all answer material.

[3] Complete your responses on the answer booklet provided.

[4] Write neatly and set out your work correctly.

[5] Use the mark allocation as a guideline:

[5.1] To estimate the amount of time to spend on a question.

[5.2] To determine the length of your response (s). E.g. do not write half a page for 1 mark.

Do not turn the page until permission is given

Section A

Question 1 [12]

1 State whether the following statements as *true* or *false* (write down ONLY the question number and your answer).

1.1. A subclass can inherit two or more super classes. (1)

1.2. An interface can contain method definitions. (1)

1.3. A class can inherit an interface. (1)

1.4. A class can implement one or more interfaces. (1)

1.5.A class can inherit another class and also implement one or more interfaces. (1)

2 The UML diagram (**Figure 1**) shows the classes **Product** and **Phone** and an interface **ITransaction** (with two method signatures: **calcvat**( ) which returns the VAT chargeable on a transaction, and **calctotalcharge**( ) which returns the total amount payable on a transaction). Use this information to answer the following questions.

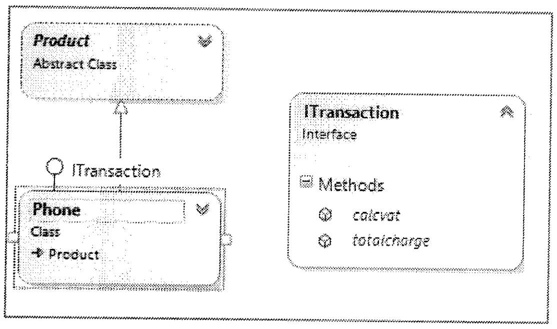


Figure 1: UML diagram

```
//2.1.1 Write the missing line of code
  {
    //2.1.2 Write the missing lines of code
  }
```

2.1. Complete the code for **ITransaction** as follows:

2.1.1. Complete the missing line of code for 2.1.1 above. (1)

2.1.2. Complete the missing lines of code for 2.1.2 above. (2)

```
public class //2.2 Write the missing line of code
  {

  }
```

2.2. Complete the missing line of code for 2.2 above for the class declaration of **Phone**. (2)

```
    2.3 Product p=new Product( );
```

2.3. Explain why the statement 2.3 above is *invalid*. (2)

Section B

Important instructions to note before answering any questions:

A. A comprehensive case study has been provided, you are required to thoroughly read through and understand the case study before answering any questions.
B. Ensure that you clearly indicate which question number you are answering.

Background

Sharon's on 11 operates a bed and breakfast. The owner has commissioned your team to develop a web application to support its business operations. Your team is currently implementing a component that manages rooms and bookings. This consists of the following pages:

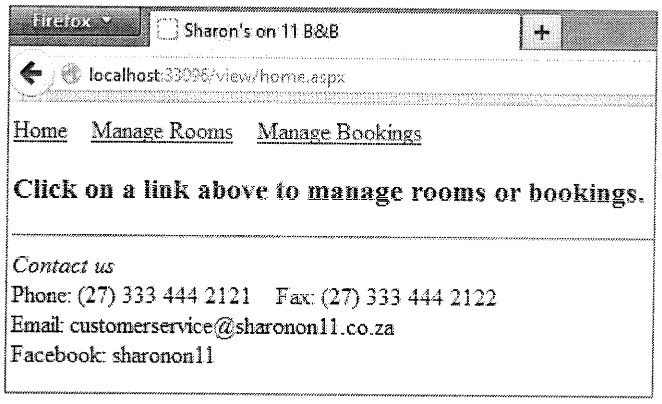
1. *home.aspx* (**Figure 1**): home page; it provides information about **Sharon's on 11**. Clicking on **Manage Rooms** navigates to the web page *rooms.aspx* while **Manage Bookings** navigates to *bookings.aspx* page
2. *rooms.aspx*: allows a user to manage rooms (i.e., add new rooms, view all rooms, etc.)
3. *bookings.aspx* (**Figure 2**): allows a user to manage bookings (i.e., view all bookings, add new booking, search for a specific booking, etc.).

home.aspx and *rooms.aspx* web pages have been implemented and tested. Your task is to develop *bookings.aspx* page to support **three tasks**: displaying, adding and searching for bookings. The following design specifications have been provided.

Presentation layer

home.aspx

This page (**Figure 1**) provides information about **Sharon's on 11**. Clicking on **Manage Bookings** navigates to *bookings.aspx* page (**Figure 2**)



Click on a link above to manage rooms or bookings.

Contact us
Phone: (27) 333 444 2121   Fax: (27) 333 444 2122
Email: customerservice@sharonon11.co.za
Facebook: sharonon11

Figure 1: home.aspx shows Manage Rooms and Manage Bookings links

bookings.aspx

This page (**Figure 2**) supports the following tasks: "**Display all bookings**", "**Add new booking**" and "**Search booking by id**". It contains:

- A mulitiview (**mvbook**) with two views (**viewbook0** and **viewbook1**).
- **viewbook0** contains a drop down list (**ddlbook**), two invisible panels (**paddbooking)** and (**psearchbooking**), and "Process" button (**btnprocess**).
- **ddlbook** is populated with "**Select a task**", "**Display all bookings**", "**Add new booking**" and "**Search booking by id**" at design time as shown in **Figure 2**.
- **paddbooking** contains a drop down list (**ddladd)** and a text box **txtcustomer**. **ddladd** is populated at run time (**Figure 4**).
- **psearchbooking** contains a dropdown list (**ddlsearch). **This is populated at run time (**Figure 6**).
- **viewbook1** contains a label (**lbbookdetails**) and a "Back" button (**btnback**).

When the page loads for the first time, the first view (**viewbook0**) is active as shown in Figure 2.

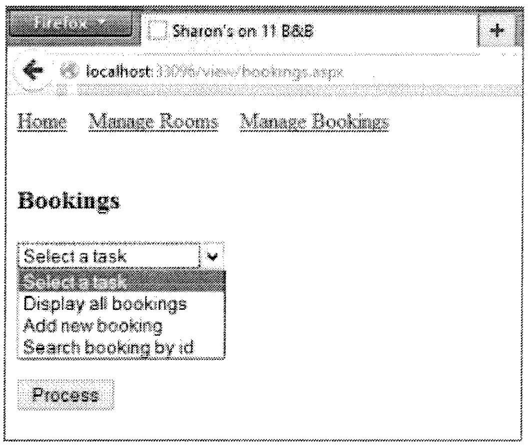
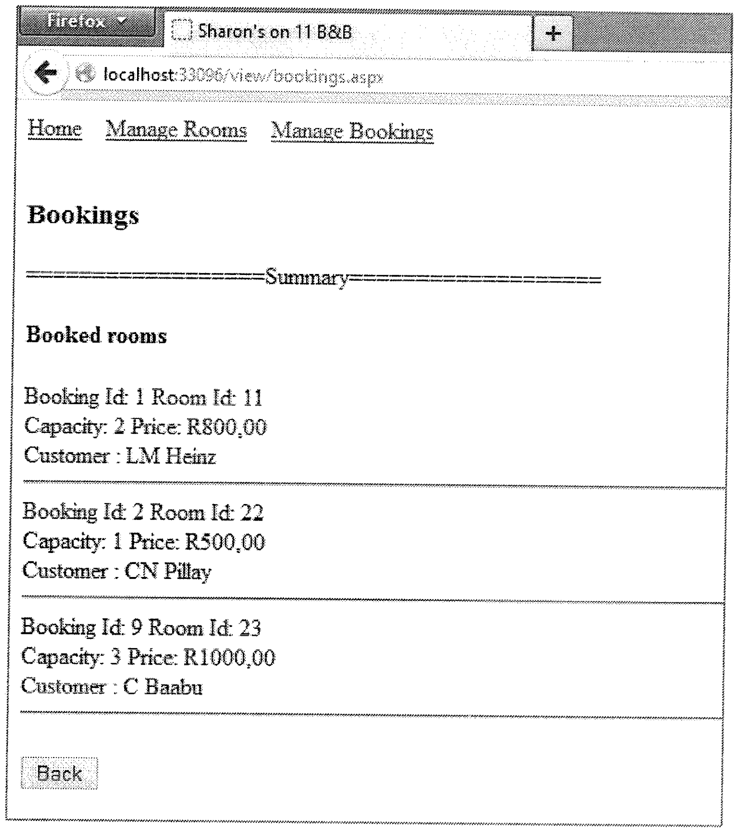


Figure 2: bookings.aspx for managing room booking

Display all bookings

1. If a user selects "**Display all bookings**" in the **ddlbook** drop down list in **Figure 2** and clicks on the "Process" button (**btnprocess**) the following occur:
- The method **getbookedrooms** ( ) in **BookingsOperations** class (**Figure 9**) is called to return a list of all **Booking** records in the **Bookings** table.
- The method **display** (Booking bpar) in **BookingsOperations** class is called to display each **Booking** object in the list on the **lbbookingdetails** label and
- **viewbook1** is made active as shown in **Figure 3**.


Firefox
Sharon's on 11 B&B
localhost:33096/view/bookings.aspx
Home Manage Rooms Manage Bookings
Back

Bookings

=================Summary==================

Booked rooms

Booking Id: 1 Room Id: 11
Capacity: 2 Price: R800,00
Customer : LM Heinz

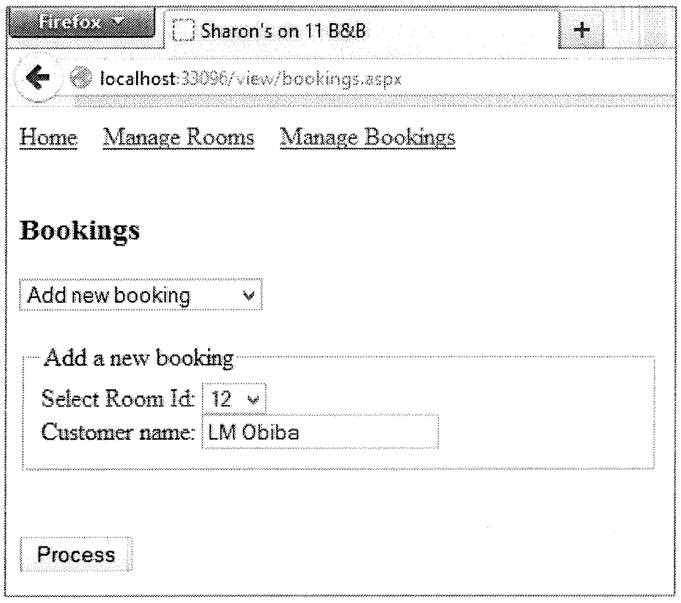
Booking Id: 2 Room Id: 22
Capacity: 1 Price: R500,00
Customer : CN Pillay

Booking Id: 9 Room Id: 23
Capacity: 3 Price: R1000,00
Customer : C Baabu

Figure 3: bookings.aspx shows all booked rooms

Add new booking

1. If a user selects "**Add new booking**"in the **ddlbook** drop down list in **Figure 2** the following occur:
- The method **getavailablerooms** ( ) in **BookingsOperations** class (**Figure** 9) is called to populate the **ddladd** drop down list with the list of **roomid** field of all **Room** objects in the **Rooms** table which are available (i.e., **booked** field equals *false*).
- Panel **paddbooking** becomes visible as shown in **Figure 4**.
- A user can select a value in **ddladd** drop down list and enter the customer's name in **txtcustomer** textbox.
2. If a user clicks on "Process" button (**btnprocess**) the following occur:
- The selected value in **ddladd** drop down list and value in **txtcustomer** textbox are used to create a new **Booking** object. This object is passed as a parameter to the method **addnewbooking**(Booking bpar) in **BookingsOperations** class (**Figure 9**).
- Feedback returned from **addnewbooking**(Booking bpar) is displayed on **lbbookdetails** label and
- **viewroom1** is made active as shown in **Figure 5**.



Firefox
Sharon's on 11 B&B
localhost:33096/view/bookings.aspx
Home Manage Rooms Manage Bookings

Bookings

Add new booking

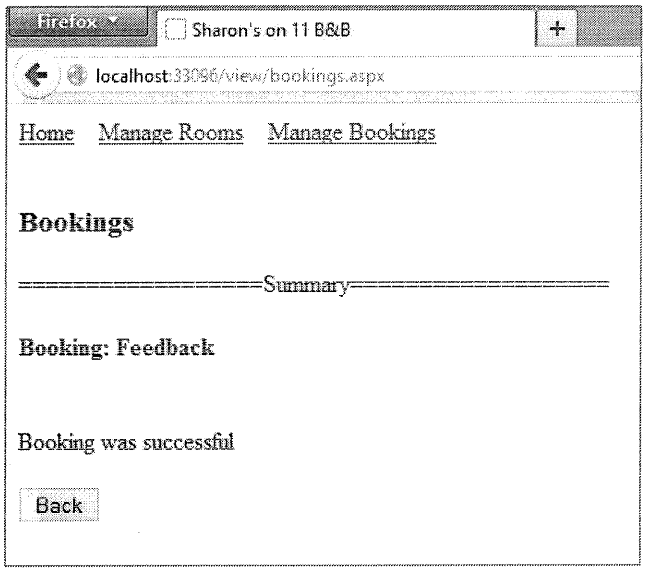
Add a new booking

Select Room Id: 12

Customer name: LM Obiba

Process

Figure 4: bookings.aspx shows panel for adding new Booking



Firefox
Sharon's on 11 B&B
localhost:33096/view/bookings.aspx
Home Manage Rooms Manage Bookings

Bookings

================Summary=================

Booking: Feedback

Booking was successful

Back

Figure 5: bookings.aspx showing feedback after adding a new Booking

Search by booking id

1. If the user selects "**Search by booking id**" in the **ddlbook** drop down list in **Figure 2** the following occur:

- The method **getbookedrooms** ( ) in **BookingsOperations** class (**Figure 9**) is called to populate the **ddlsearchbook** drop down list with *bookingid* field of all **Booking** objects in the **Bookings** table
- Panel **psearchbooking** becomes visible as shown in **Figure 6**

2. If a user select a *bookingid* in **ddlsearchbook** drop down list and clicks "Process" button the following occur:
- The selected value in **ddlsearchbook** drop down list is passed as parameter to the method **findabooking**(int bookingidpar) in **BookingsOperations** class (**Figure 9**). This method returns a **Booking** object in the **Bookings** table whose *bookingid* matches the parameter. This object is displayed on the **lbbookingdetails** label and
- **viewbook1** becomes visible as shown in Figure 7.

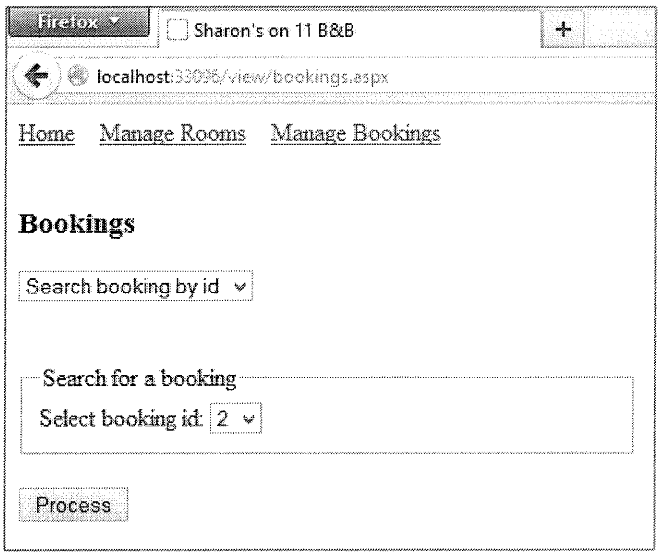
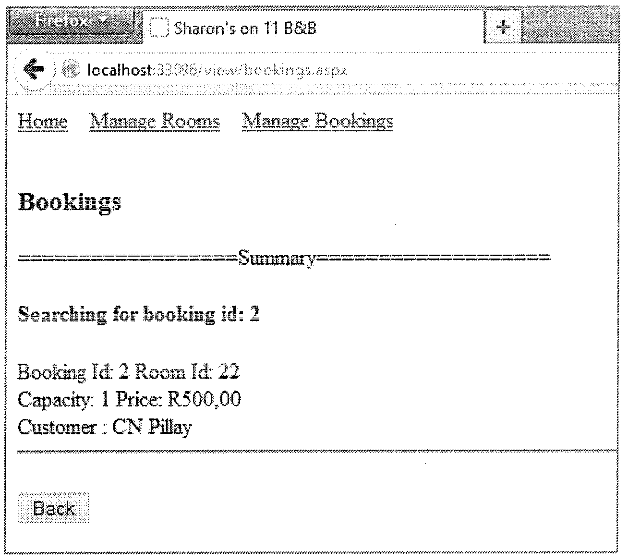


Figure 6: bookings.aspx showing a list of bookingid



```
=================Summary===================

Searching for booking id: 2

Booking Id: 2 Room Id: 22
Capacity: 1 Price: R500,00
Customer : CN Pillay
```

Figure 7: bookings.aspx shows results of a search

Data layer

Listing 1 shows the SQL script for database and tables.

```
CREATE DATABASE bandbdb;
USE bandbdb;
CREATE TABLE Rooms
(       roomid INT PRIMARY KEY NOT NULL,
        descr NVARCHAR(20) NOT NULL,
        capacity INT NOT NULL,
        price DECIMAL NOT NULL,
        booked BIT NOT NULL
);
CREATE TABLE Bookings
(       bookingid INT IDENTITY PRIMARY KEY NOT NULL,
        roomid INT FOREIGN KEY REFERENCES Rooms(roomid) NOT NULL,
        customer NVARCHAR(20) NOT NULL
);
INSERT INTO Rooms VALUES
(11, '1st Floor', 2, 800, 'False'),
(12, '1st Floor', 1, 450, 'True'),
(13, '1st Floor', 2, 600, 'False'),
(14, '1st Floor', 2, 600, 'False'),
(21, '2nd Floor', 2, 600, 'False'),
(22, '2nd Floor', 1, 500, 'True'),
(23, '2nd Floor', 3, 1000, 'True'),
(24, '2nd Floor', 1, 600, 'False');
INSERT INTO Bookings VALUES
(11, 'LM Heinz'),
(22, 'CN Pillay'),
(23, 'BS Ngcobo');
```

Listing 1: SQL script

Figure 8 shows the entity framework classes (*BandbEntities*) generated from the database and tables above.

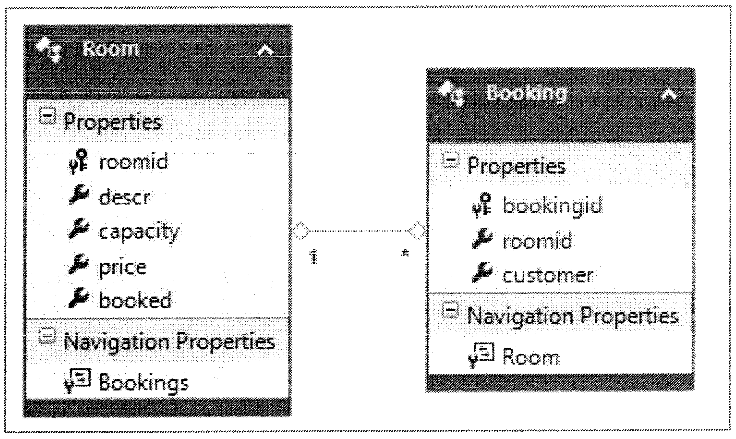
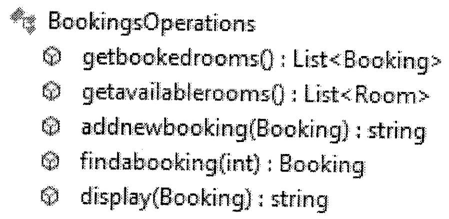


Figure 8: Entity framework classes (BandBEntities)

Business Logic

Figures 9 shows the **BookingsOperations** class and methods in the business logic.



```
BookingsOperations
    getbookedrooms() : List<Booking>
    getavailablerooms() : List<Room>
    addnewbooking(Booking) : string
    findabooking(int) : Booking
    display(Booking) : string
```

Figure 9: BookingsOperations class

The methods are described below:

1. *public List<Booking> getbookedrooms*( ): returns all **Booking** objects in the **Bookings** table
2. *public List<Room> getavailablerooms*( ): returns only **Room** objects in **Rooms** table whose *booked* field equals *false*
3. *public string addnewbooking*(Booking bpar): takes a **Booking** object as parameter, inserts the object into **Bookings** table, set the **booked** field of a Room object with a matching **roomid** to *true*, and returns "Booking was successful"; otherwise "Booking failed"
4. *public Booking findabooking*(int bookingidpar): takes an integer (*bookingid*) as a parameter and returns a **Booking** object in the **Bookings** table whose *bookingid* matches the parameter.
5. *public string display* (Booking bpar): takes a **Booking** object as parameter and returns a string representing *bookingid*, *roomid*, *capacity*, *price* and *customer* from the **Bookings** and **Rooms** tables as shown in **Figures 3** and 7.

Use the Sharon's on 11 case study above to answer the following Questions 2 and 3 below.

Question 2 [35]

With reference to the **BookingsOperations** class and the methods described above, answer Questions 2.1 to 2.5 below.

```
public List<Booking> getbookedrooms()
{
   //2.1 Write the missing lines of code
}
```

2.1. Complete the missing lines of code for 2.1 above. (3)

```
public List<Room> getavailablerooms()
{
   //2.2 Write the missing lines of code
}
```

2.2. Complete the missing lines of code for 2.2 above. Use Lambda. (6)

```
public string addnewbooking(Booking bpar)
{
   //2.3 Write the missing lines of code
}
```

2.3. Complete the missing lines of code for 2.3 above. (9)

```
public Booking findabooking(int bookingidpar)
{
   //2.4 Write the missing lines of code
}
```

2.4. Complete the missing lines of code for 2.4 above. Use Linq. (7)

```
public string display(Booking bpar)
{
   //2.5 Write the missing lines of code
}
```

2.5. Complete the missing lines of code for 2.5 above. Use Linq (10)

Question 3 [23]

With reference to the **bookings.aspx** web page and code behind, answer Questions 3.1 to 3.4.

```
<asp:DropDownList id="ddlbook" //3.1.1 Write the missing lines of code
        //3.1.2 Write the missing lines of code
        //3.1.3 Write the missing lines of code ="ddlbook_SelectedIndexChanged">
         <//3.1.4 Write the missing lines of code >
        <//3.1.5 Write the missing lines of code >
        <//3.1.6 Write the missing lines of code >
 </asp:DropDownList>
```

3.1. Write the tag for the **ddlbook** drop down list by completing the missing line of code for 3.1.1 to 3.1.6 above. (6)

```
    protected void Page_Load(object sender, EventArgs e)
     {
        //3.2 Write the missing line of code
     }
```

3.2. Complete the missing line of code for 3.2 such that the first view (**viewbook0**) in multiview **mvbook** is active when **bookings.aspx** page loads for the first time. (1)

```
    protected void ddlbook_SelectedIndexChanged(object sender, EventArgs e)
     {
        if (ddlbook.SelectedIndex == 2)
        {
           //3.3. Write the missing lines of code
        }
     }
```

3.3. Complete the missing lines of code for 3.3 such that if the selected index in **ddlbook** (**Figure 2**) is 2 (i.e., "**Add new booking**" is selected), the application will call **getavailablerooms ( )** in **BookingsOperations** class to populate the **ddladd** drop down list and make panel **paddbooking** visible (**Figure 4**). (7)

```
    protected void btnprocess_Click(object sender, EventArgs e)
     {
        if (ddlbook.SelectedIndex == 2)
        {
           //3.4. Write the missing lines of code
        }
     }
```

3.4. Complete the missing lines of code such that if a user selects *roomid* from **ddladd** drop down list, enters customer name and clicks "Process" button (**Figure 4**), the application will insert a **Booking** object into **Bookings** table, display feedback on **lbbookdetails** and make **viewbook1** active (**Figure 5**). (9)