Advanced Data Centric Web Applications

Library Project

1 Description 3
2 Requirements 3
3 Submission 3
4 Marks 4
4.1.1 Marking Scheme 4
4.1.2 Plagiarism 4
5 Skeleton 4
6 Functionality 5
6.1 Main Screen 5
6.2 List Books 6
6.3 Add New Book 7
6.3.1 Success Conditions 7
6.3.2 Invalid user input conditions 8
6.4 List Customers 9
6.5 Add New Customer 10
6.5.1 Success Conditions 10
6.5.2 Invalid user input conditions 12
6.6 List Loans 13
6.7 Add New Loan 14
6.7.1 Success Conditions 14
6.7.2 Business Logic for Order Creation 16
6.7.3 Invalid user input conditions 17
6.8 Delete Loan 19
6.8.1 Invalid user input conditions 20
6.9 Security 21
6.9.1 Logout 21
6.10 Testing ..... **Error! Bookmark not defined.**

1 Description

Write a Spring Boot MVC application that allows a logged-in user to manage customers, books and loans in an online library.

2 Requirements

The application should be written in Spring Boot based on the skeleton application available on Moodle and should use the Spring MVC n-tier architecture – Controllers, Services, Repositories, Views, with the appropriate data/logic in each tier.

The Model for the application consists of three objects:

- Customer
  - Customer ID
  - Customer Name
  - Load Period
- Book
  - Book ID
  - Title
  - Author
- Loan
  - Loan ID
  - Customer ID
  - Book ID
  - Due Date

These models have been given in the skeleton code for the project and their dependencies already set up.

3 Submission

A zipped file containing the source code of the entire application should be uploaded to Moodle no later than **Monday April 29th 2019 at 9:00am**.

4 Marks

This assignment is worth 50% of the marks for the module.

4.1.1 Marking Scheme

85% of the marks will be awarded for implementing the functionality described in this document.

15% of the marks will be awarded for innovation and good programming practice.

Please describe your innovation (if any) in a document entitled *innovation.doc* which should be stored in the root folder of your project.

4.1.2 Plagiarism

Plagiarism will be dealt with in accordance with the institute's Plagiarism policy.

5 Skeleton and Databases

The application should be created by updating the skeleton application available on Moodle.

6 Functionality

6.1 Main Screen

When the application starts, the user should see the following screen:

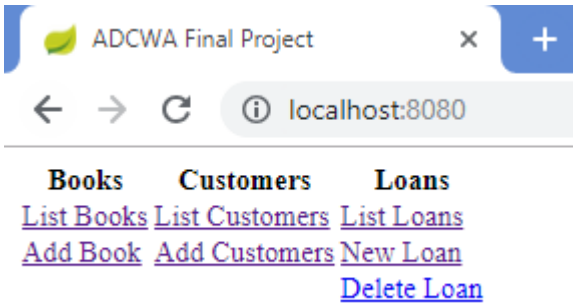


Figure 1 Main Screen

6.2 List Books

This screen shows the list of Books in the *books* database:

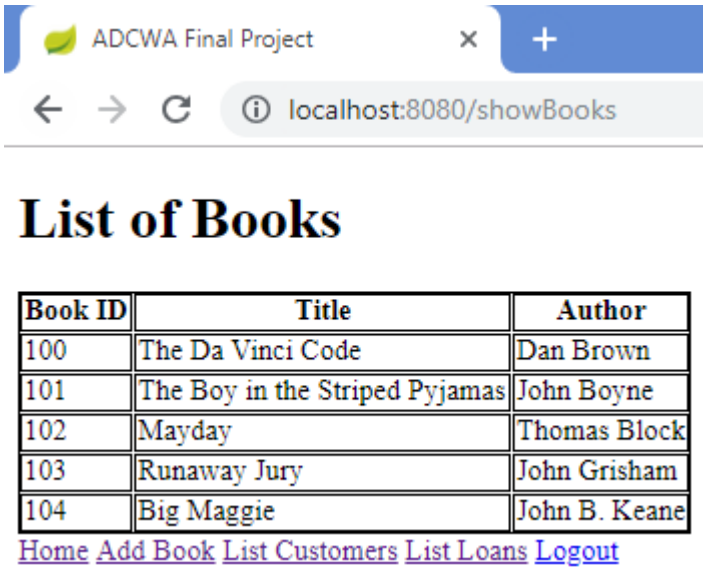


Figure 2 List Books

6.3 Add New Book

This screen allows the user to add a new Book.

6.3.1 Success Conditions

If all information is entered correctly, the books database is updated with the new Book information:

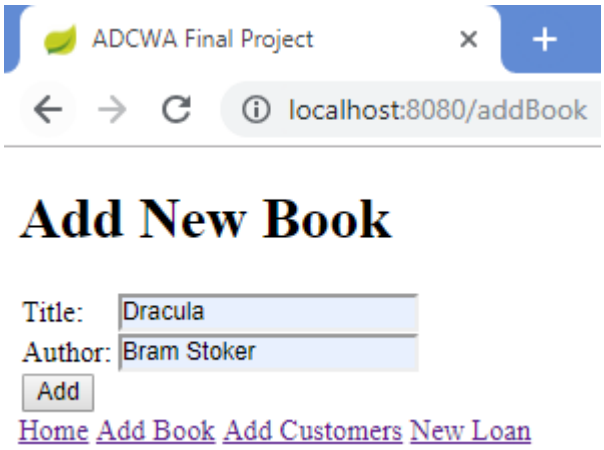


Figure 3 Add New Book – OK

And the user is immediately brought to the List Books page, where the book just entered is shown:

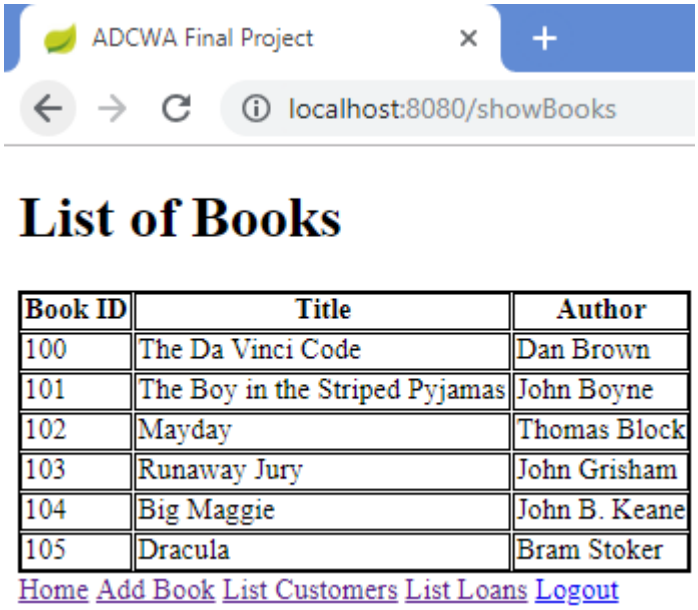


Figure 4 List of Books after new book added

6.3.2 Invalid user input conditions

If the book title and/or author is blank the following errors should be displayed as appropriate, and no book should be added to the books database.

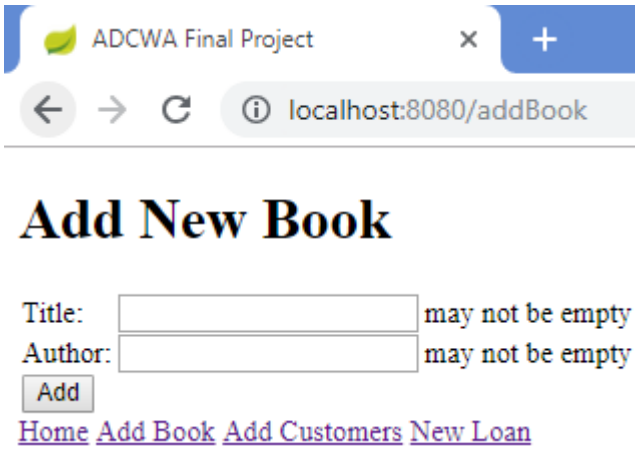


Figure 5 Add New Book – NOK

6.4 List Customers

This screen shows the list of Customers in the books database.

Underneath each Customer is the list of loans they currently have, if any.

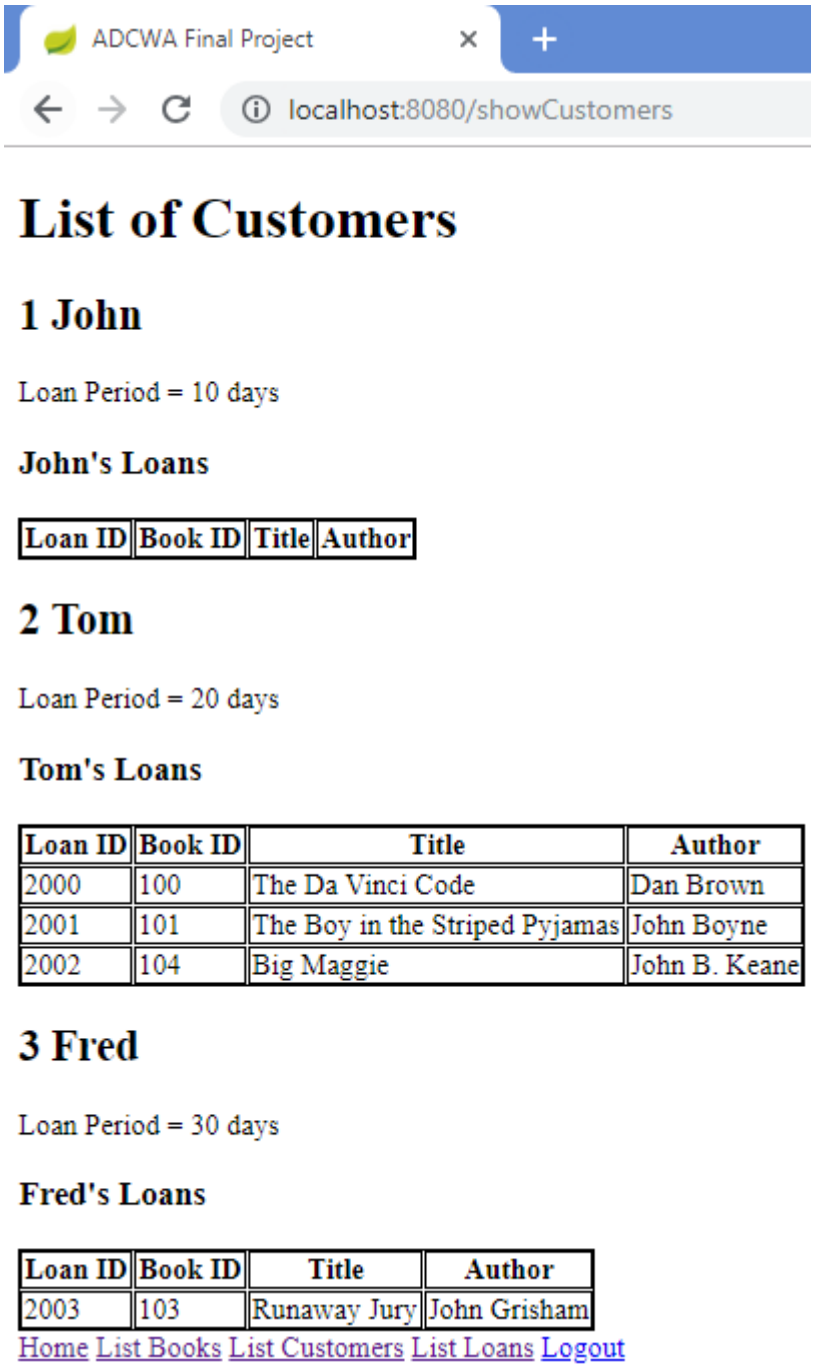


Figure 6 List of Customers

6.5 Add New Customer

This screen allows the user to add a new Customer.

6.5.1 Success Conditions

If all information is entered correctly, the books database is updated with the new customer information:

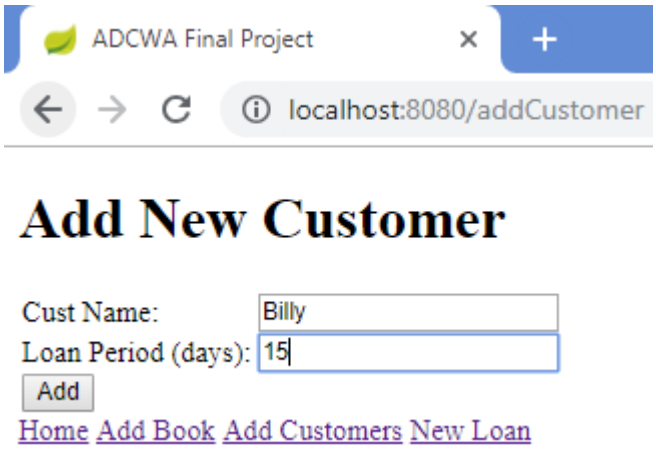


Figure 7 Add New Customer – OK

When a new Customer is successfully added, the user is immediately brought to the List Customers page, where the customer just entered is shown:

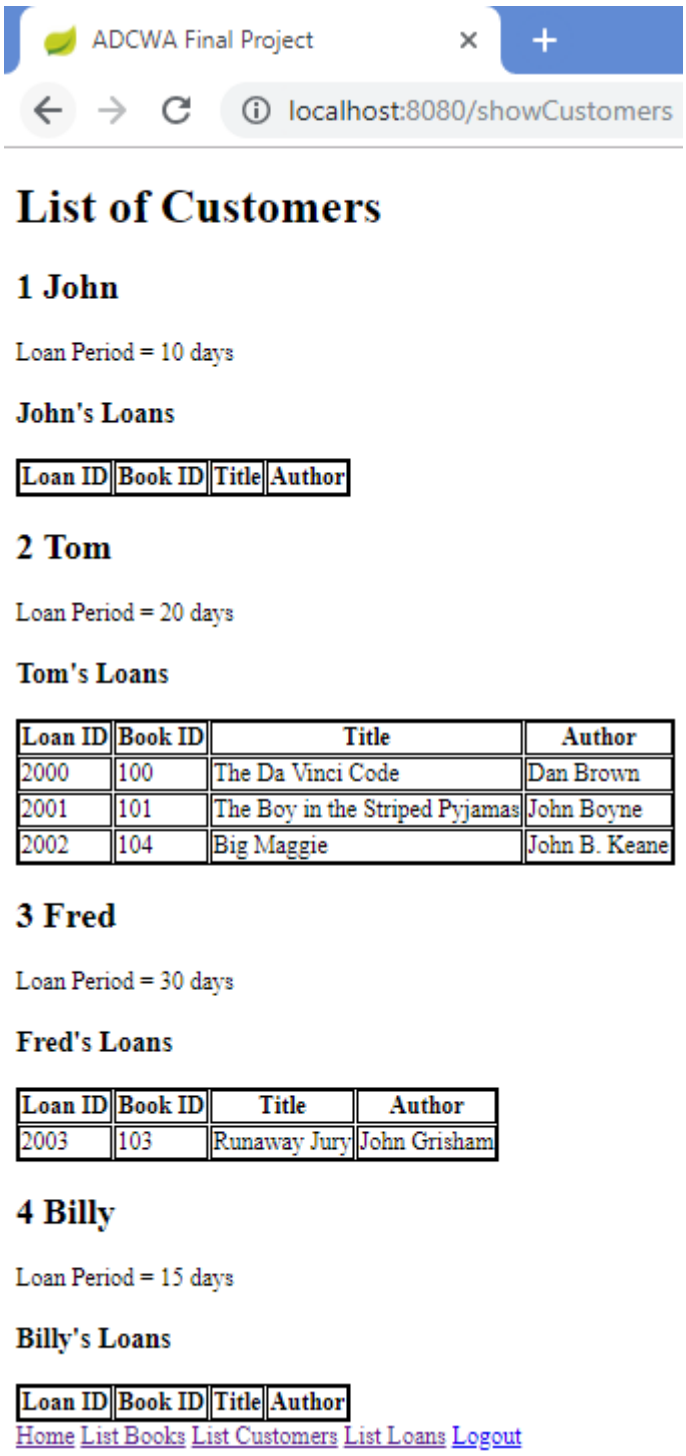


Figure 8 List of Customers after new customer added

6.5.2 Invalid user input conditions

If the customer name is blank, or the loan period is less than 1 day, the following errors should be displayed, and no customer should be added to the books database.

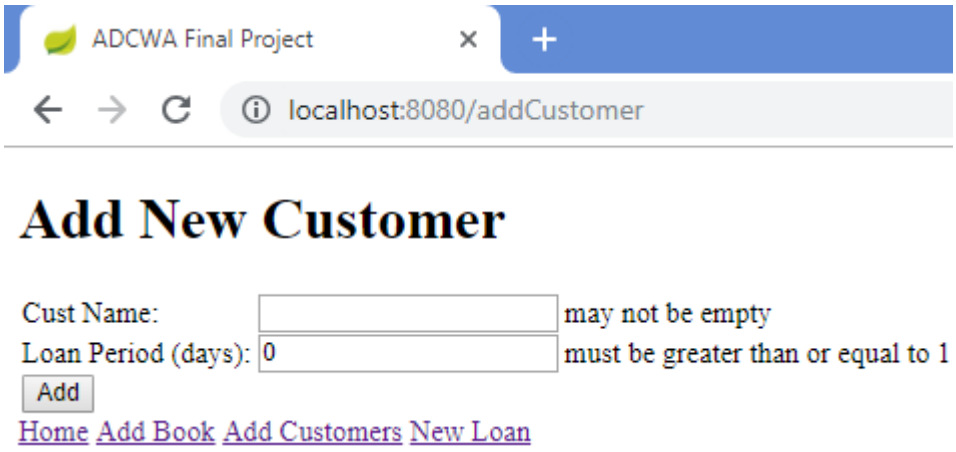


Figure 9 Add New Customer – NOK

6.6 List Loans

This screen shows the list of Loans in the books database. The table consists of the loan ID, Customer ID associated with the loan, Customer Name, Book Title and Author of the Book on loan by the Customer, and due date of the loan.

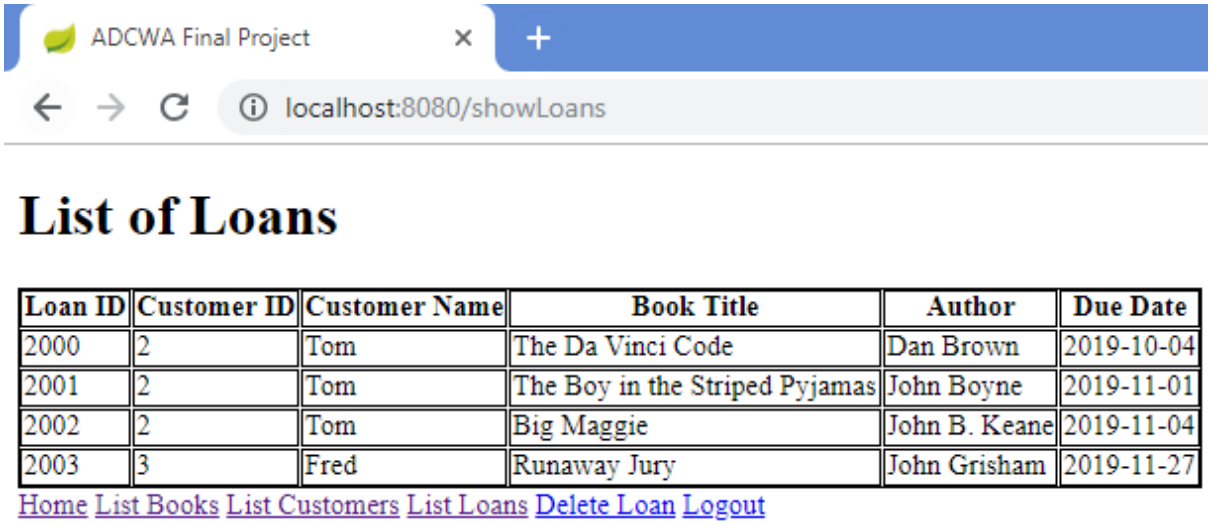


Figure 10 List of Loans

6.7 Add New Loan

This screen allows the user to add a new Loan.

6.7.1 Success Conditions

If all information is entered correctly, the books database is updated with the new customer information:

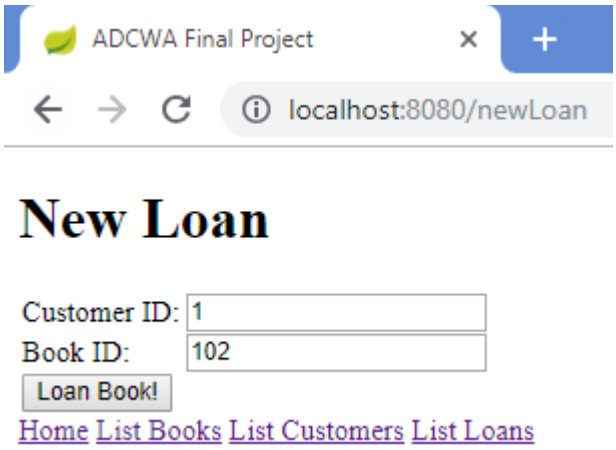


Figure 11 New Loan - OK

And the user is immediately brought to the List Loans page, where the loan just entered is shown:

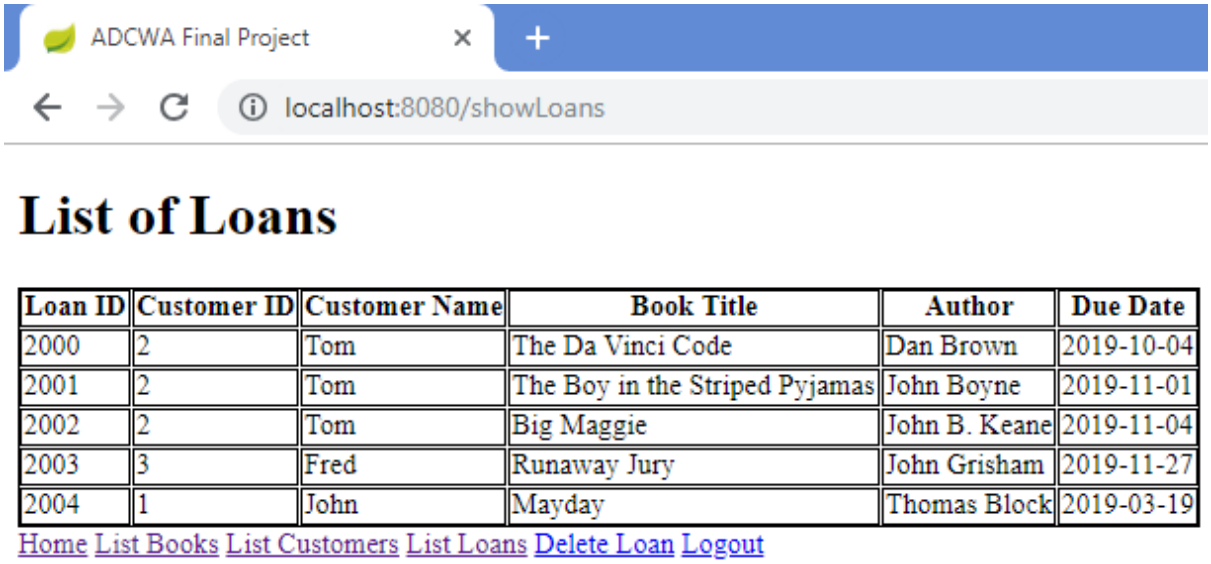


Figure 12 List of Loans after new loan added

6.7.2 Business Logic for Order Creation

- The Due Date is automatically inserted by the application – the user does not have to enter the date.

- Each user has a loan period which is the number of days he/she can borrow books for. So, if a customer is borrowing a book, the due date is calculated as the current date plus the customers loan period. For example:
    - If the Current Date was April 1st 2019, and the loan period was 10 days the due date would be April 11th 2019, but if the loan period was 30 days the due date would be May 1st 2019.

6.7.3 Invalid user input conditions

(Assume the Customer ID and Book ID will never be blank and never be a String).

If the Customer ID does not exist and/or the Book ID does not exist:

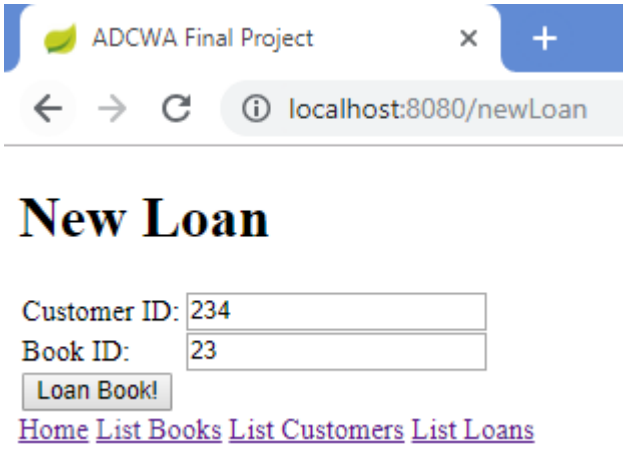


Figure 13 Invalid Customer ID and Book ID entered

The following error page will be shown:

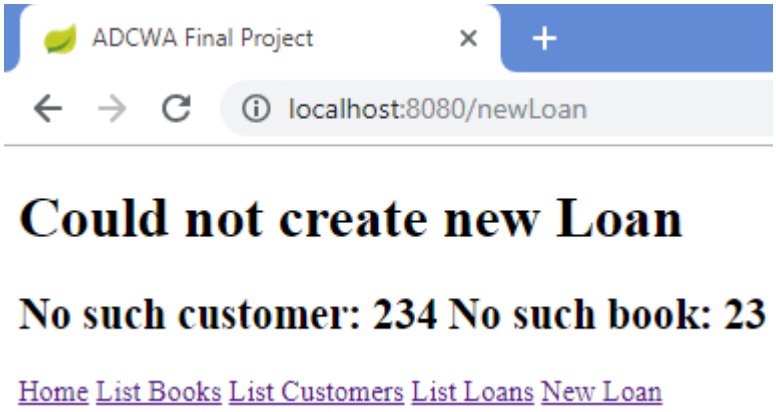


Figure 14 Invalid Customer ID and Book ID

If the Customer ID and Book ID are both valid, but the Book is already on loan:

Figure 15 Book ID 104 already on loan

The following error page is displayed:

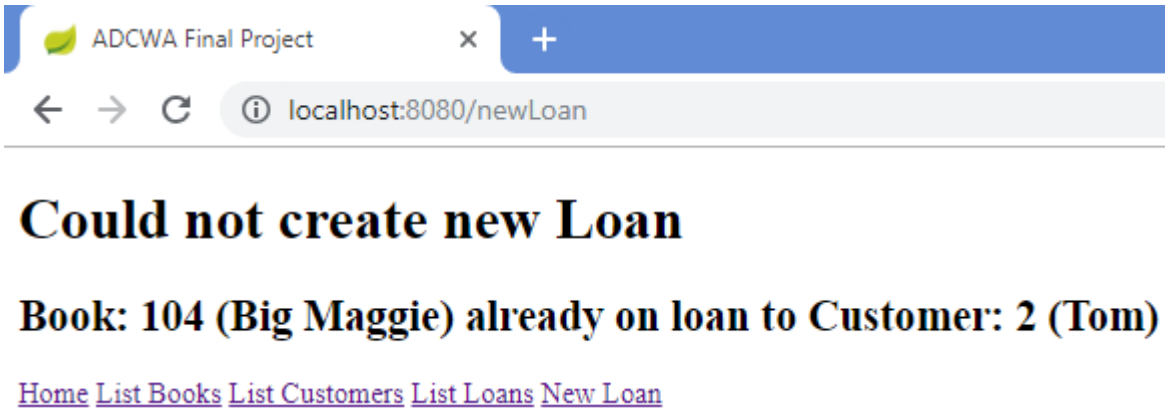


Figure 16 Error showing that Book already on loan

6.8 Delete Loan

A loan can be deleted by entering the Loan ID:

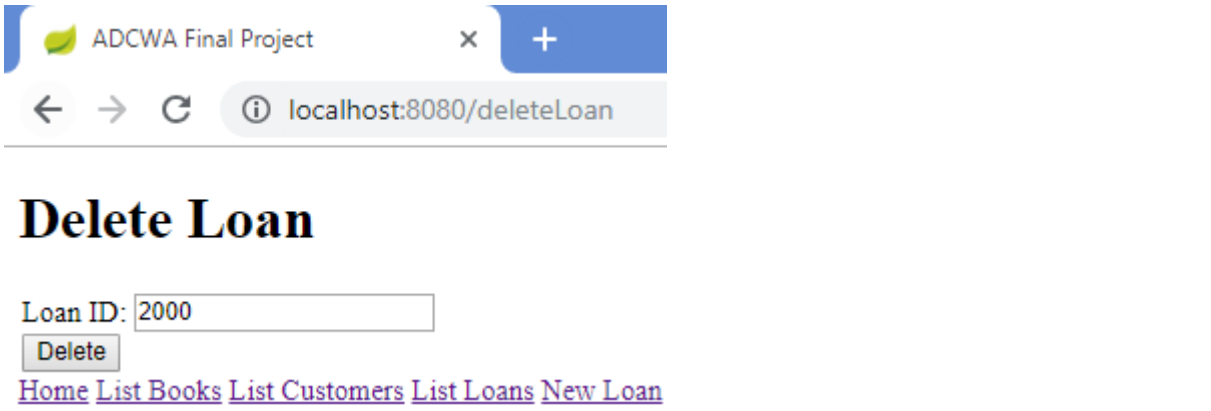


Figure 17 Load ID to be deleted

The loan is then deleted from the database and the user brought to the List Loans page:

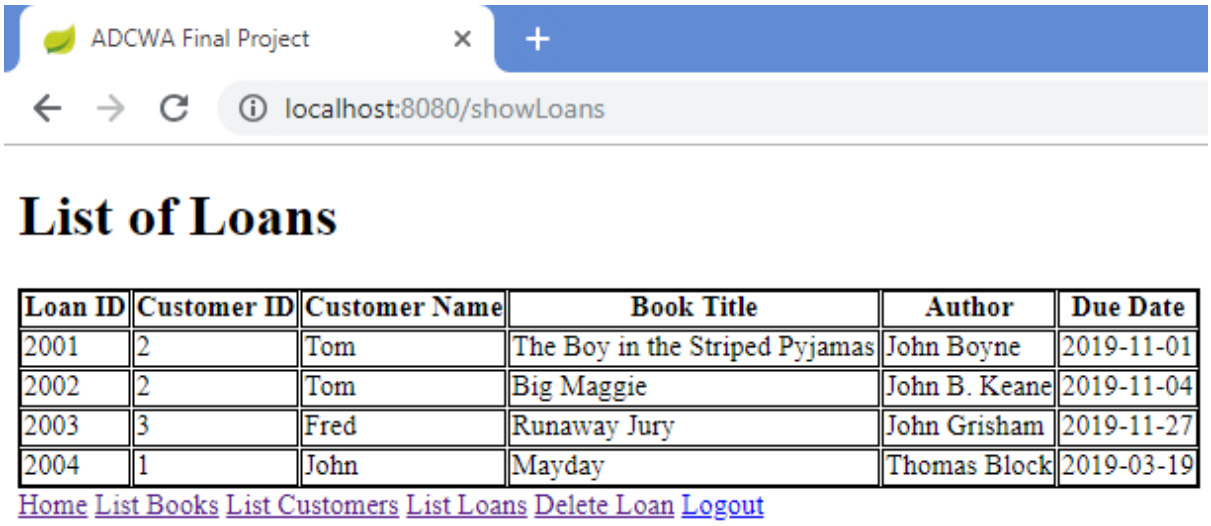


Figure 18 List of Loans after Load 2000 deleted

6.8.1 Invalid user input conditions

If an invalid Loan ID is entered, the following page is displayed:

(Assume that the Loan ID will never be blank, or a String)

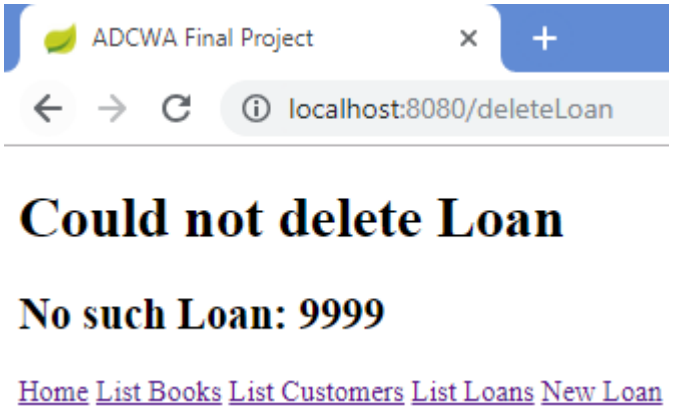


Figure 19 Invalid Load ID to delete

6.9 Security

The application should implement security so that unless a user is logged in they will not be able to navigate to any URL.

The only page they should be able to see is the index.html:

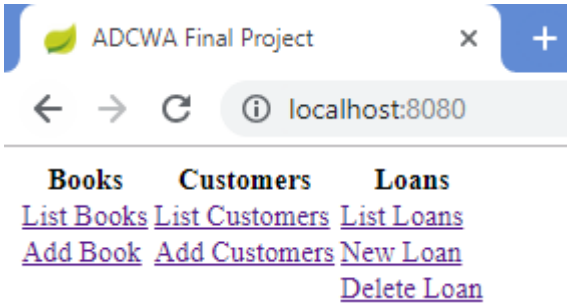
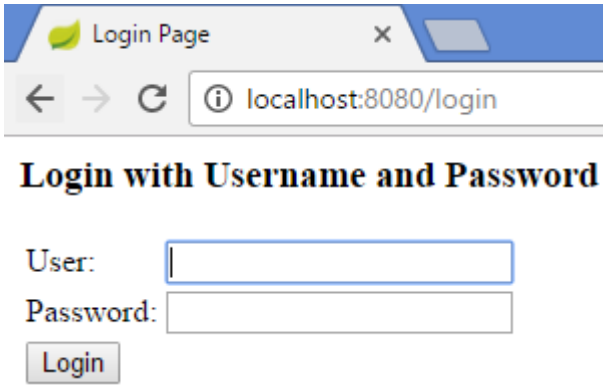


Figure 20 index.html always accessible

If user has not logged in and they attempt to access any of the applications URL endpoints, they should be presented with the following form allowing them to log in:



There should be one user (`user`) with password `user`.

6.9.1 Logout

There should be a logout button on the List Customers screen (see Figure 6 List of Customers).