Subject: Homework 4
Date assigned: Monday, November 14, 2016
Date due: At start of class on Wednesday, December 7, 2016

General: This assignment gives you a chance to apply the skills learned in chapters 1 - 12 to create a library book checkout system for use by Belk Library. This application runs on an OpenShift tomcat 7 web server with a MySQL database and allows librarians to check out a book to a library patron and to manage checked out books to determine which are overdue and to check in books when they are returned.

Objectives: The objectives of this homework are:

1. To demonstrate your ability to **create a database** and table(s) in MySQL by creating a SQL script with the necessary commands and running the script using MySQL Workbench.

2. To demonstrate your ability to **use JDBC** in a data access layer to perform insertion, deletion and reading of library checkouts to your MySQL database.

3. To demonstrate your mastery of **Model View Controller** architecture by using servlet controllers with all views displayed with JSP and business logic written in java to create model objects for use by views.

Collaboration Requirements: This is a pairwise assignment. You are only allowed to discuss the coding of the assignment with your assigned partner and Dave Powell. I will treat any violations of this requirement as an honor code violation. You must place a copyright statement as a comment near the top of each html, css, jsp, sql and java file that indicates the work is completely and originally your own.

Requirements: The specific requirements are listed below. If you have any questions on the requirements then please see me for interpretation.

1. Create a Java Web Application Project with a project name of **homework4** with a context path of **/homework4**. You should use an OpenShift Tomcat 7 application with the MySQL Cartridge for your application deployment.

2. Your html, css and jsp files must follow the ***Google HTML/CSS Style*** guide. In addition to the style guide, you must place a Copyright with your name and your team mates name on each html, css and jsp file indicating that the work was originally and completely done by you and your team mate.

3. All css styles should come only from external files. The files should be in a **styles** folder.

4. Your servlet code and model code must follow the ***Google Java Style Guide*** coding conventions. You must use a **Model 2 MVC** architecture with all control coming from java

servlet controllers, all model calculations coming from java classes and all views coming from html and jsp files.

5. You must create a SQL script file with a copyright that when run from MySQL workbench will create the table that you need to represent checked out books. The table should at a minimum hold data for the user's first name, user's last name, user's email address, book title, and due date. You can add other columns/properties/fields as you deem necessary. Note: the due date should be set at book checkout to be two weeks from the current date of checkout. Include your SQL script file in a folder under Web Pages called **sql**.

6. Your application must support two ways of being started. Your application should be started by either typing in the OpenShift Application URL followed by the ContextPath or the application URL followed by the ContextPath followed by the name of the controller servlet of **/library**.

   My application is called gameon-eloncsprof.rhcloud.com and my Context Path is homework4. When invoked as

   http:// gameon-eloncsprof.rhcloud.com/homework4/

   or as

   http:// gameon-eloncsprof.rhcloud.com/homework4/library

   then a jsp or html page should be displayed similar to that shown in Figure 1. You should use an image of Belk Library that you can download from http://www.elon.edu/images/e-web/library/belk-02.jpg.

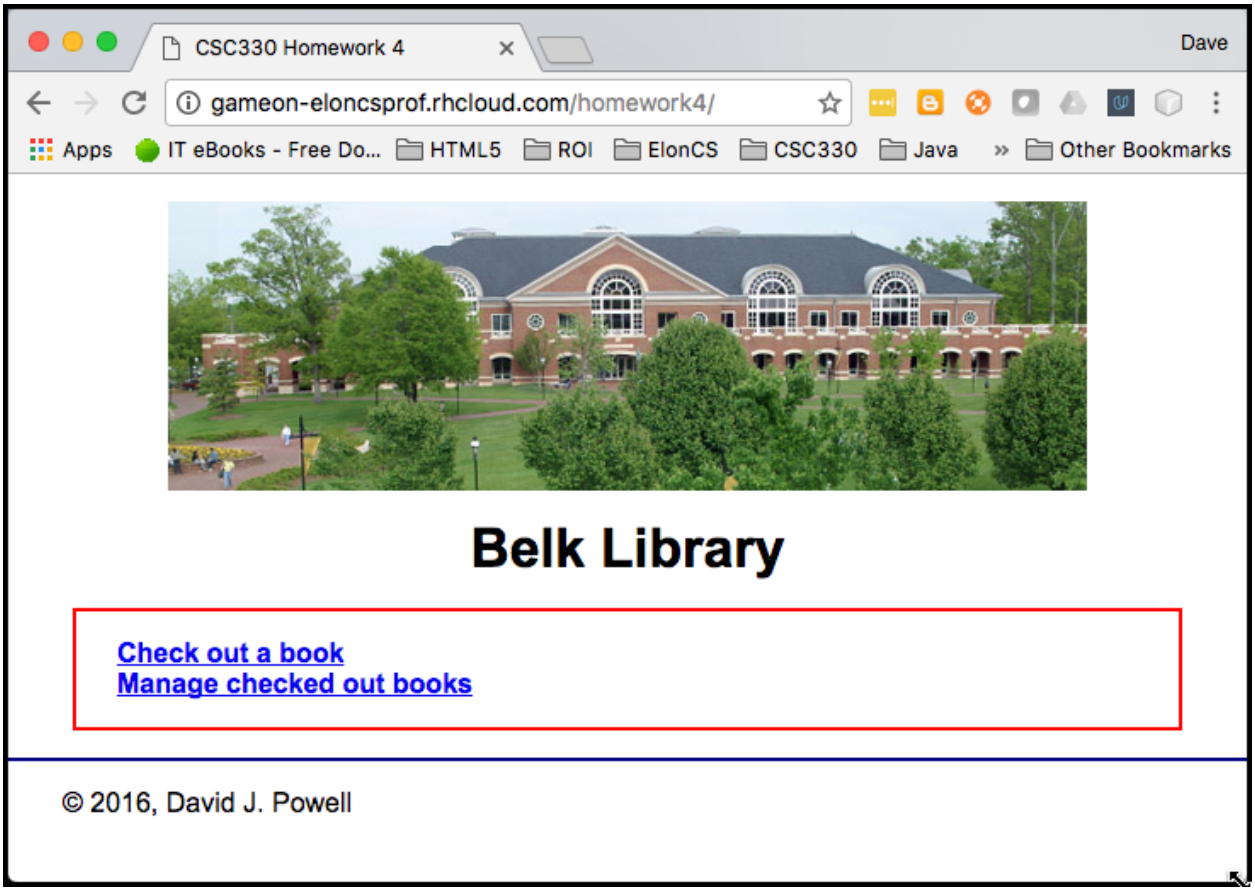

Figure 1: Initial Display of View with the standard header for all Views in the Elon Library application and 2 links to available operations.

7. If the user clicks on **Check out a book** link in Figure 1 then a form should be displayed as shown in Figure 2. Note the same header and footer in Figures 1 and 2. You are **required** to use the same templates for all views to provide a standard header and footer. All footers should have a copyright with the names of each of your team displayed to indicate the work is originally and completely your own.

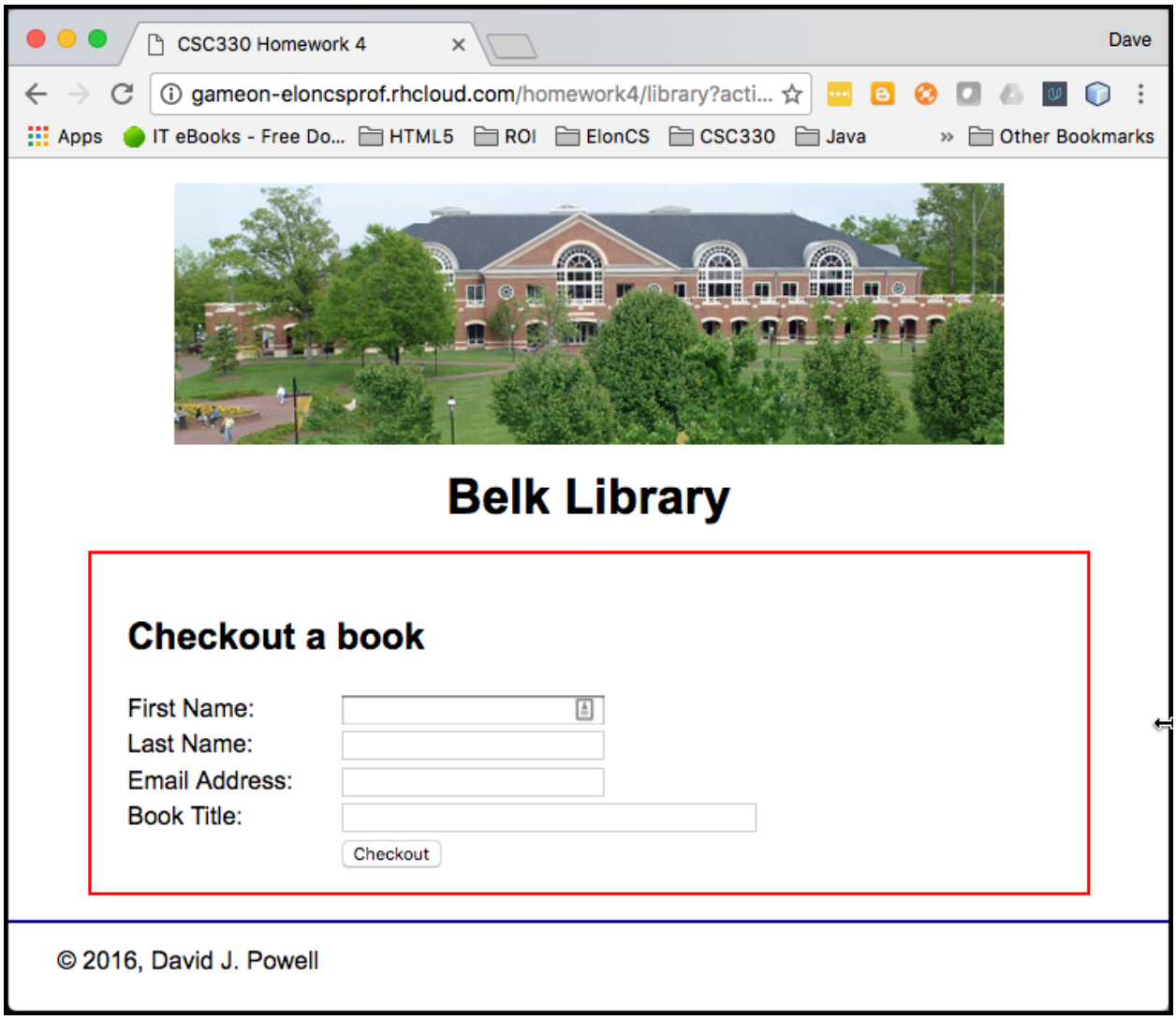

Figure 2: User has entered valid data for each input field and is about to click Calculate

8. You should use HTML 5 types and attributes to insure that all fields in Figure 2 are required and that a valid email formatted entry is added before the **Checkout** button will successfully submit.

9. Figure 3 shows a form with all needed data filled in. When Checkout is clicked then a page indicating successful checkout with the required due date similar to Figure 4 is displayed. The Checkout data should be stored in your MySQL database.

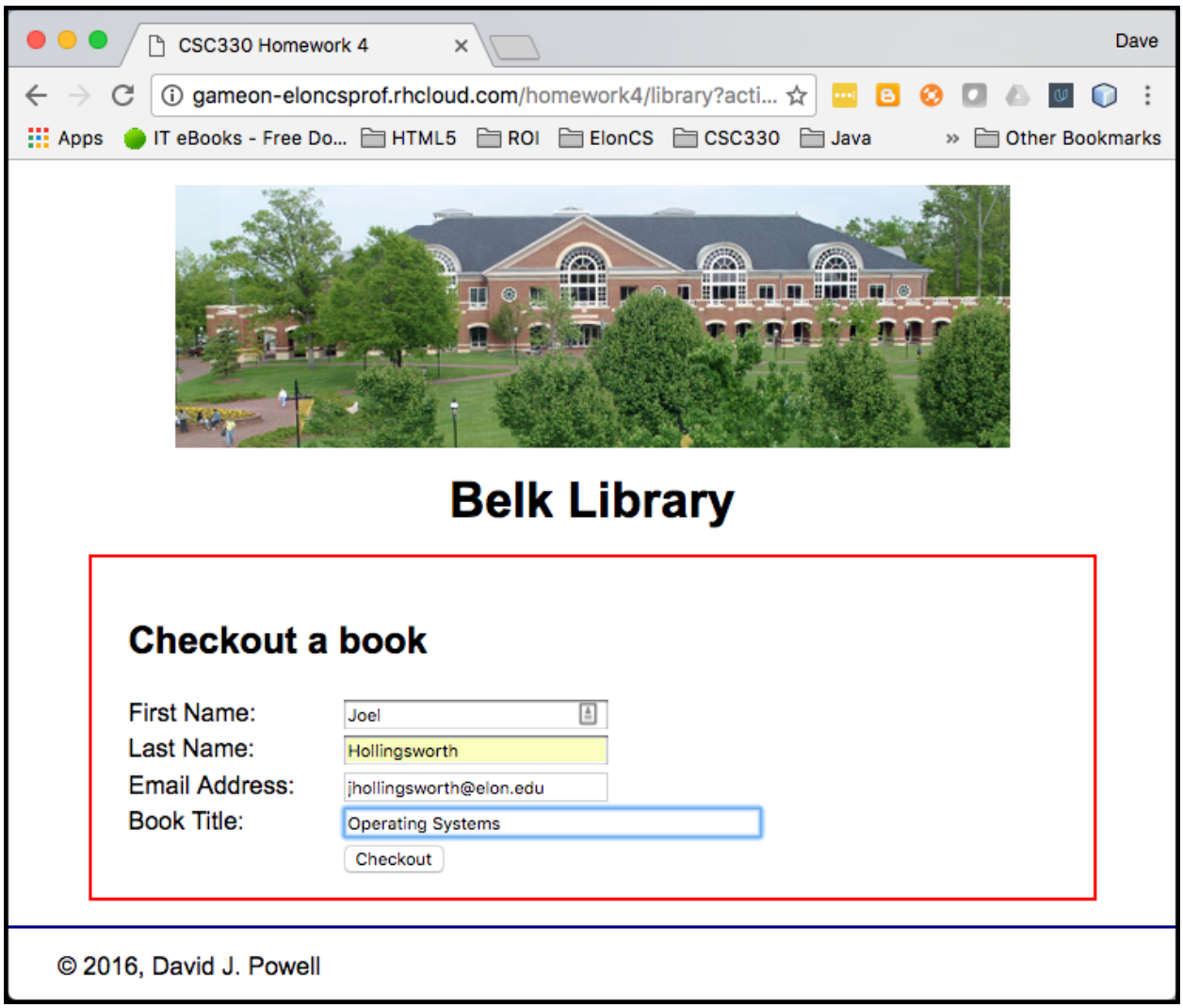

Figure 3: Required data provided to checkout a book

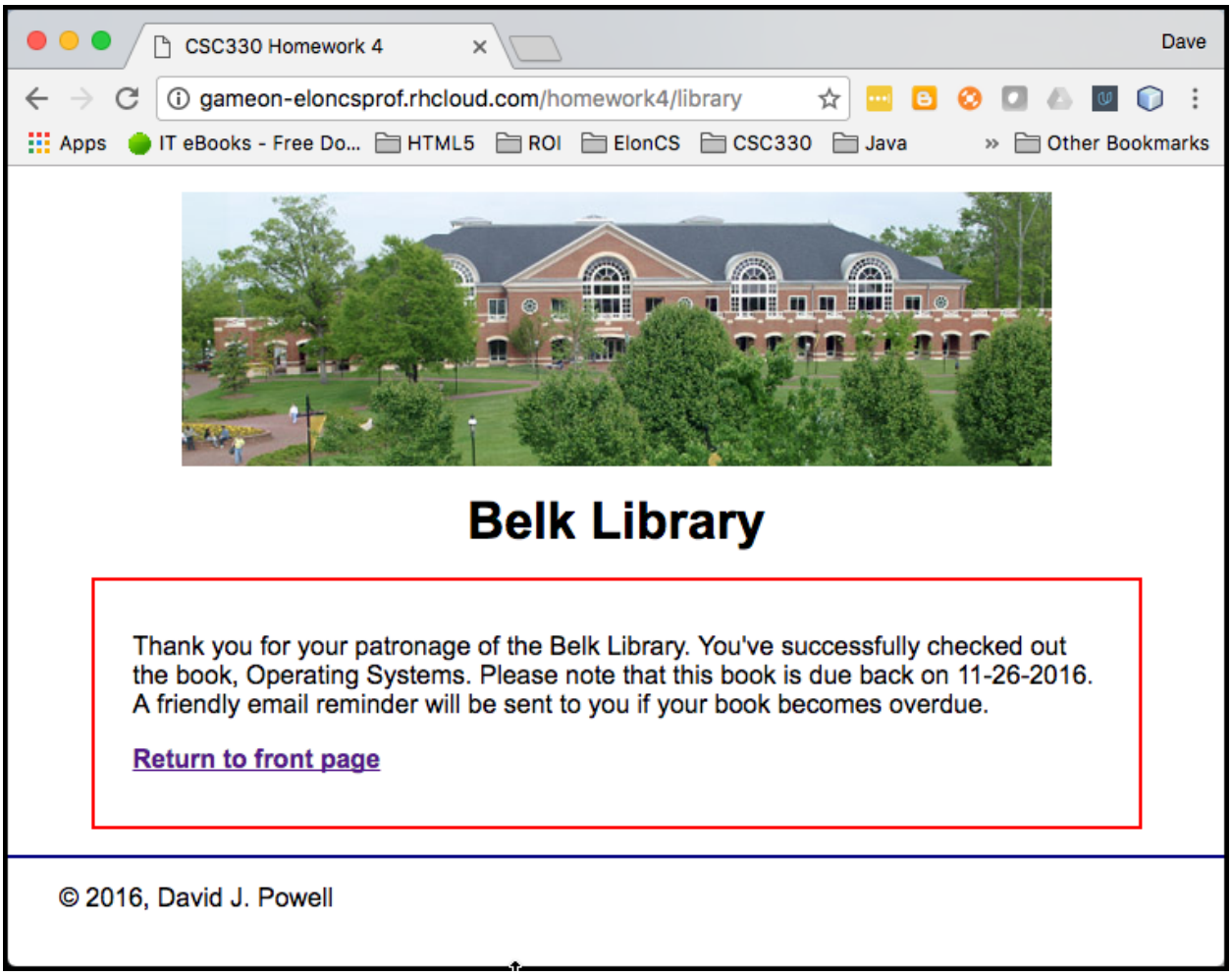

Figure 4: Checkout confirmed for book title and due date provided. Book was checked out on Nov.12, 2016.

10. The due date should be 2 weeks from the day the checkout is entered. (Hint: look at the **GregorianCalendar**, **java.util.Date** and **java.sql.Date** classes for java library methods for working with dates.) If the user clicks the link **Return to front page** then Figure 1 should be displayed.

11. If the user clicks the link **Manage checked out books** then all of the checked out books should be selected from the database and displayed as shown in Figure 5. Each book should display the user name, email address, title, due date, overdue status and have a button to check it in. Figure 5 shows 3 books checked out when I clicked it on November 12. At this time, Shannon Duvall had one overdue book. If the book is not overdue then there will be no entry in the overdue column.

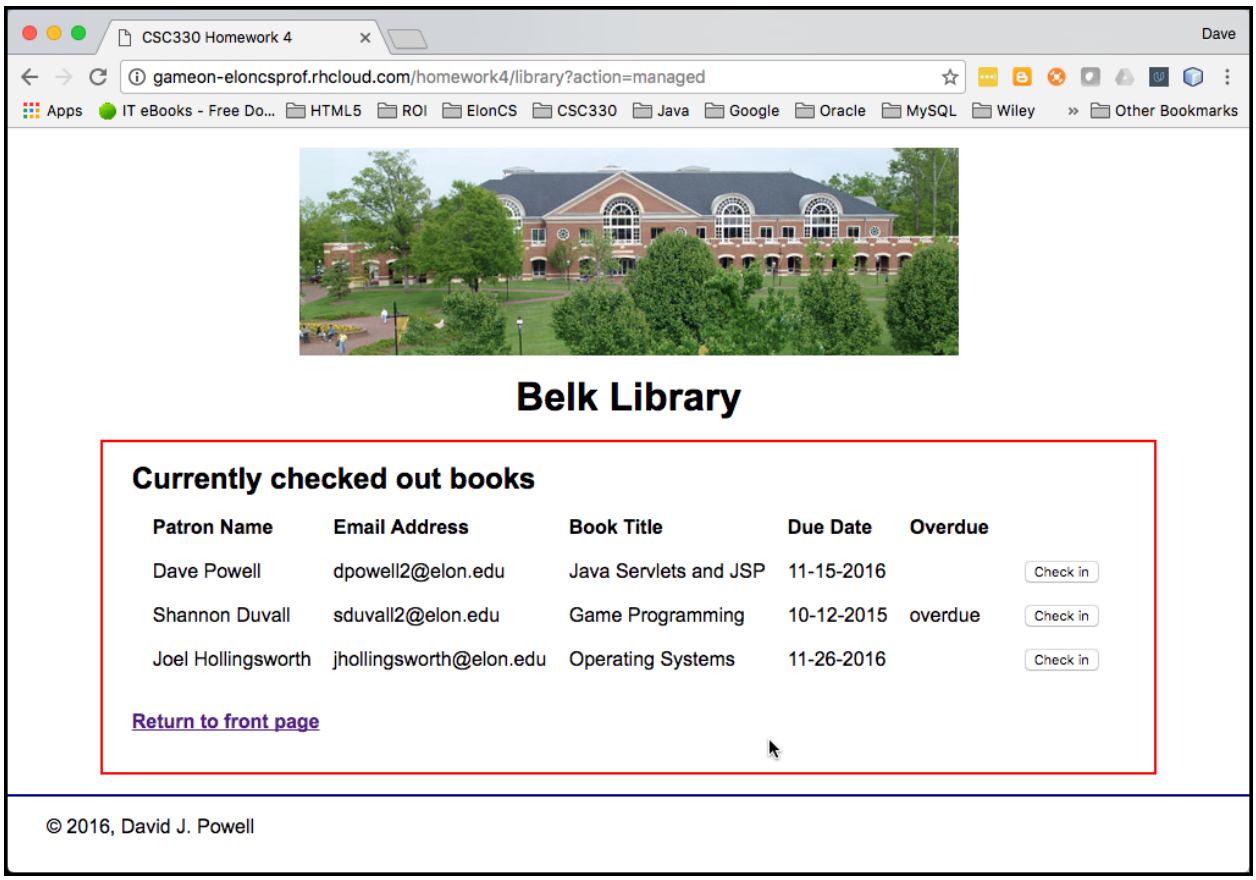

Figure 5: All checked out books shown. Each book has an option to check it in.

12. If Shannon brings her book titled “Game Programming” back to the library and the librarian clicks the **Check In** button then the book is removed from the Database and the page is redisplayed with all of the books that are still checked out. Figure 6 shows that only two books remain checked out and that both are not overdue. At this point, the user can **Check In** another book or click the link **Return to front page**.

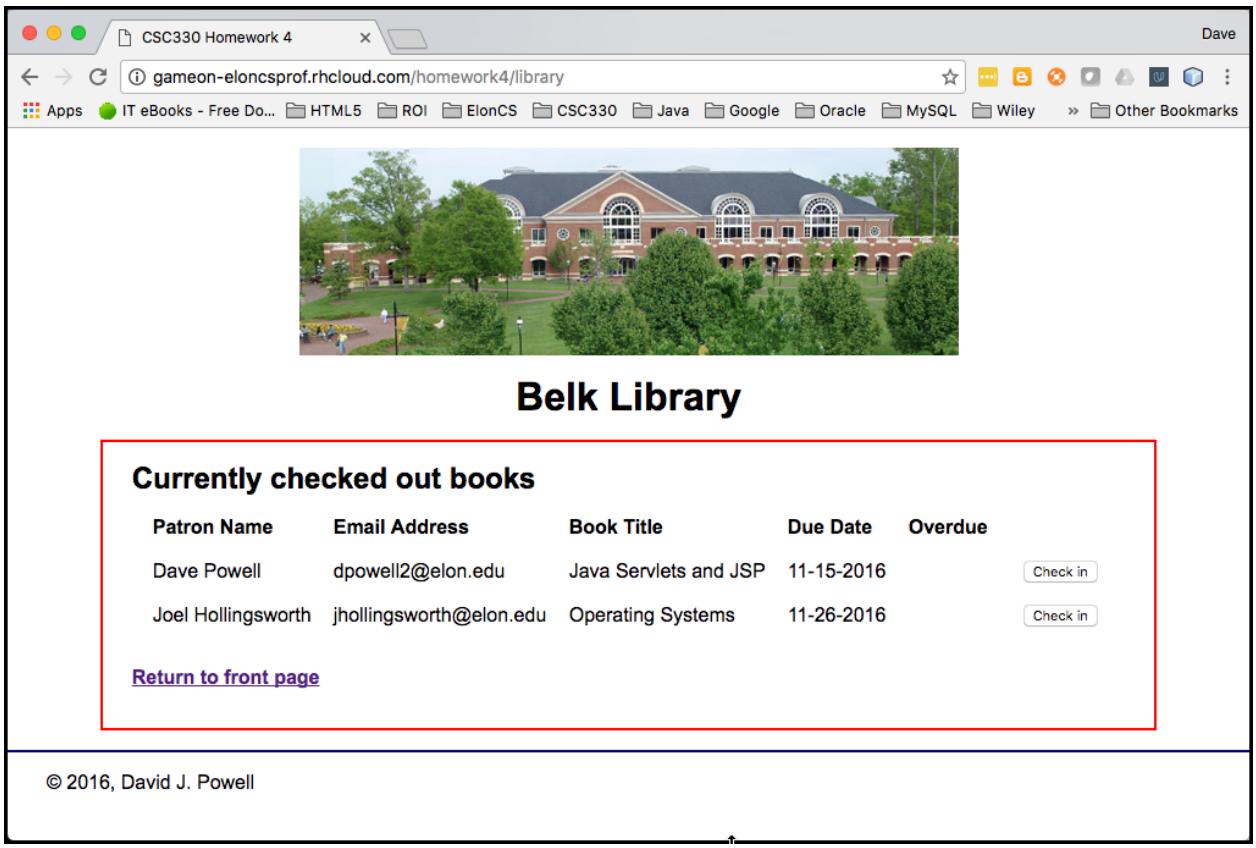

Figure 6: Books checked out after Shannon's book Artificial Intelligence checked in in Figure 5.

13. Your servlet should be the controller for the application and all communication and navigation goes through the servlet.

14. Each html and jsp file must use a DOCTYPE of html and validate for html 5 using the validator at http://validator.w3.org/

15. Each css file must validate for css 3 using the validator at http://jigsaw.w3.org/css-validator.

16. There is a soft copy and hard copy submission requirement.

    - For the softcopy, you must submit one NetBeans exported project file from your team to Moodle. Export your Project as a zip file called homework4.zip and submit it to moodle as an assignment file upload before the start of class on the due date. The softcopy **must include the sql script** to create the tables needed in your database.

    - For the hard copy that you submit, hand in at the start of class your html, css, jsp, web.xml, java and sql files. Insure these files are properly formatted so they are easily readable with no wrapping. **At the top of the hardcopy, hand write or underline the**

full URL on OpenShift that I need to invoke to display your application. Note: double check the correctness and readability of your link to insure that I can easily access your application.