

## **Homework Three: ServletApp-ServletApp Communication**

**Assigned: November 3, 2014**

**Due: 9:30 a.m. November 21, 2014**

**Points: 100**

### **Overview:**

Read this requirements document carefully.

This assignment is intended to familiarize students with more advanced functionality that includes communication between two servlet-based applications and the utilization of AJAX. The students will be using AJAX technology for interaction between applications.

For this assignment, you will continue working on the servlet/jsp-based airline reservation application from Assignment 2. You will also create a servlet-based banking application to complete the purchase. The assignment adds AJAX functionality to the previous homework to establish communication between the airline reservation application and the banking application. Carefully follow the procedure below for adding AJAX functionality to your web application.

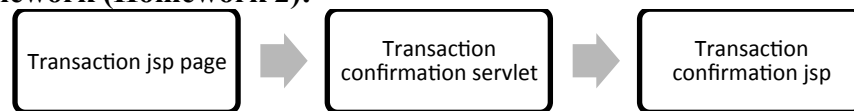
This assignment is to be done by a team of two students.

### **Assignment Description:**

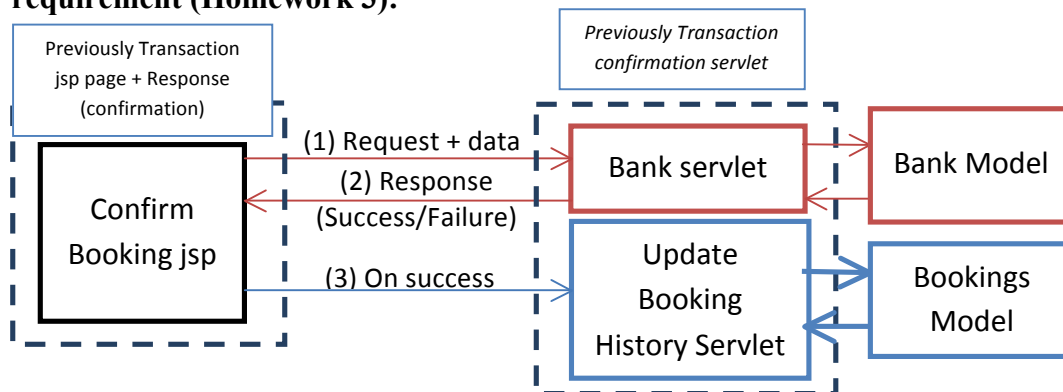
- Retain the jsp and servlet functionalities of the airline reservation application from Homework 2.
- Add the “Shopping Cart” functionality by modifying the ViewAndBook jsp page as follows:
  - The flight number is received from the Flight Search Results servlet.
  - The page must also have the option to select number of seats for a particular flight.
  - Once the number of seats is chosen, user may click on the “Add to Cart” button.
  - This will add their flight to the cart and display a meaningful message (.g., Flight is added). You must use AJAX to implement the shopping cart functionality.
  - User can add more than one flight to the cart.
  - The page should have a “Check Out” button to lead the user to the ShoppingCart jsp page.
  - User can use the “Back to search results” button to go to the results page. They can select “Search for new flights” to search in the query again.
- Add ShoppingCart jsp page the includes the following:
  - User should be able to access this page from any page
  - This page displays a list of flights that the user has added to the cart.
  - It should show the cost for each flight and the total cost.
  - “Continue Shopping” button leads the user to the flight search page.
  - Clicking the “Check Out” button calls the ViewAndBook servlet passing list of added flight numbers and number of seats.

- Modify the previous ViewAndBook servlet as follows:
  - Get the flight parameters passed by ShoppingCart.jsp page.
  - Verifies the availability of the requested number of seats and then redirects the user to the Transaction.jsp page with the flight details, number of seats and total cost.
- Divide the functionalities of the “Transaction confirmation” servlet from homework 2 into two different servlets: “Bank” and “Update Booking History”. Make a Bank Model.

**Previous Homework (Homework 2):**



**New requirement (Homework 3):**



- Combine functionalities of “Transaction” and “Transaction Confirmation” .jsp pages from homework 2 into a single “Confirm Booking” .jsp page. This page will have the following components:
  - All elements from the original transaction.jsp page
  - Clicking the “Confirm Transaction” button calls a javascript function (confirm\_func).
- **Javascript functions:**
  - **confirm\_function:**
    - Initiates AJAX call to the bank servlet with the required data.
    - Populate the result elements (described below) with the response received from the bank servlet.
    - On Success: Show Success/Failure message and a form for the user to enter passenger details (Name, Age, Sex, etc...), Print Ticket button. Also calls update\_history\_function.

- On Failure: The page displays a message: “Transaction was not successful”
- **update\_history\_function:**
  - Initiates AJAX call to the update booking history servlet with the booking details.
- **Other elements (To display result):**
  - It can be any HTML element (like div, span etc.) capable of displaying the response of the bank servlet.
  - This will initially contain all flight information passed by the “View and Book” servlet.
  - This also contains a blank “Response” element (div/span etc.) to show the response.
    - This will later be filled by the javascript function with the response returned by the Bank servlet.
- **Update History servlet:**
  - Adds the booking details to booking history via the bookings model.
  - You can copy this functionality from “Transaction Confirmation” servlet in homework 2.
- **Bank servlet:**
  - Receives bank details from the “Confirm Booking” jsp page (using AJAX).
  - Uses Bank Model to check bank account standing.
  - Redirects user to the Transaction Confirmation jsp page with flight details and transaction status (Success/Failure).
- **Bank Model:**
  - Check the given account if its balance is greater than the booking amount. Return transaction success / failure status accordingly. If successful, deduct the amount from the balance.

### **General Requirements:**

- Although the “Airline Reservation” application and “Banking” application in this assignment will reside on the same server, the two applications and related servlets should be treated as though they are on separate servers and may only communicate by sending data to each other over a network using HTTP
- Most of the general requirements from Homework 2 apply here as well.
- AJAX data may be passed in any format desired.
  - HINT (Keywords): Plain Text, JSON, XML.
- Proper HTML form elements and function must be employed for input areas.

- You may change the modify the database based on the assignment requirement.

### Requirements for 864/Bonus for 464:

1. Modify the ViewAndBook.jsp as follows: Use AJAX to check the availability of the seats requested by the user.
2. Modify the ShopingCart.jsp as follows: Add Edit buttons for each flight that allows the user to add/reduce number of seats. Use AJAX to check the availability of the seats requested by the user. Once the total seat number is changed successfully (based on the availability), update cost for the respective flight and the total cost. Add Remove buttons for each flight that allow the user to remove the flight from the cart and update total cost.
3. Implement the include method of RequestDispatcher in at least one of your servlets. See section 15.8 in Hall & Brown's text.

### Assignment Submissions:

What to submit using Webhandin (Email submissions will NOT be accepted):

1. You need to submit **two war files**: AirlineResearvation and Banking. **When creating the war file you must include your java source files.** This war file will be imported into Eclipse for grading.
2. Grading rubric
3. A single document in the pdf format with the following assignment information:
  - a) Discuss division of labor specifying who did what and why this is a fair and equal split (team)
  - b) Explanation of status and stopping point, if incomplete (individual).
  - c) Explanation of additional features, if any. (individual)
  - d) Discuss what was easy with the assignment and what was challenging. How did you overcome all or some of the challenges? (team)
  - e) Documentation in the form of both a programmer's manual and user's manual should be included. (team)

**You MUST use the following naming convention for your WAR, grading rubric and documentation files:**

- team name-HW3-AirlineReservation.war (e.g., Team1-HW3-AirlineReservation.war)
- team name-HW3-Banking.war (e.g., Team1-HW3-Banking.war)
- team name-HW3-Rubric.pdf (e.g., Team1-HW3-Rubric.pdf)
- team name-HW3-Documentation.pdf (e.g., Team1-HW3-Documentation.pdf)