

CSC309, Summer 2007, Assignment 2

In this assignment you will build an issue tracking system.

Notes:

- Your code should function correctly in the CDF environment. Part marks may be available for design aspects rather than just functional aspects of your code, but your code will not be debugged by the marker to make it work.
- It is not sufficient to just test your code; functionality alone does not guarantee maximum marks. Your solution should be professionally designed to handle concurrency, scalability, and maintainability issues. Ensure mutual exclusion of concurrent updates wherever necessary, while keeping mutual exclusion to a minimum. Organize your code in a way that will survive further changes and extensions.
- Your solution files should never have public read, write, or execute permissions until after we stop accepting submissions. Any “group” or “other” permissions must be disabled; enable “user” permissions only for you to test the code in your account.
- Provide a README file with clear instructions for all the steps to install and run your solution files in a CDF account. *You may chose to use Apache for web pages and Tomcat for servlets, or just Tomcat for both web pages and servlets.*
- This assignment is marked out of 100, i.e., up to 10 bonus points are available.

Deliverables:

- Archive (tar and gzip) the entire application folder deployed in the Tomcat server, containing all the files you have created for this solution, including at least the following files:
 - Revised order form for Questions 1 and 5.
 - Clerk web page for Questions 2 and 3. (One page containing answers to both questions.)
 - Source code and class files for Ordering, Pickup, Delivery, and Tracking servlets. (Only deliver the final versions of these servlets.)
 - Deployment descriptor file (web.xml).
 - README with configuration and installation information, including but not limited to all the relevant environment variables, setup steps you used for the server or servers, and setup steps for Pointbase and the database you used.

Question 1. [10 points]

Extend the order form of the midterm with Javascript validation to ensure that the phone number, address 1, first name, last name, and card inputs in the user contact information were filled in. The order form is posted here:

http://www.cdf.toronto.edu/~radu/csc309/pizzapronto/midterm_csc309_20070704.html

Question 2. [20 points]

Implement a servlet, call it Ordering, that takes orders from the order form in Question 1. The Ordering servlet stores the incoming orders into an orders pipeline (array or Vector). Implement the orders pipeline as a member variable (a field) of the Ordering servlet. Record the order timestamp (time of arrival) together with the order itself. The Ordering servlet delivers a confirmation page as part of the HTTP response.

Question 3. [30 points]

Implement a web interface for a pizza shop clerk to update the orders pipeline of Question 2. The entire current orders pipeline is displayed on a web page to be used by the clerk. Next to each order, there is a button labeled "Send" that the shop clerk will click when the pizza is ready for pickup. On pressing "Send", the order will be moved from the orders pipeline into a pickup pipeline. Implement this move operation by a service method of a second servlet, call it Pickup. Implement the pickup pipeline as a member variable (field) of the Pickup servlet.

Question 4. [20 points]

Refine the solution to Question 3 by adding a button for each order, labeled "Delivered", that the shop clerk will press when the pizza is delivered. On pressing "Delivered", the order will be removed from the delivery pipeline and stored in a permanent storage as a database record. Implement this operation by a third servlet, called Delivery, using a connection to an existing Pointbase database. You may set up the database and tables manually using the Pointbase tools, and include the steps in the README file submitted.

Question 5. [20 points]

Implement an order tracking function which shows the order status to a returning buyer. Insert a hyperlink into the order form saying "click here to check order status". This hyperlink accesses a fourth servlet, called Tracking, which displays the status (received, sent) of each order from that buyer that is currently in either the orders pipeline or the pickup pipeline. If no orders are found, the Tracking servlet should print "No orders found". Use cookies to identify returning buyers, and store the cookies in the orders and pickup pipelines.

Question 6. [10 points]

Indicate the relevant advantages and disadvantages of cookie-based versus session-based solutions for Question 5.

Good luck, and have fun!