School of Electrical Engineering & Computer Science

University of Newcastle

Seng4400 – Enterprise Software Architecture

Assignment 2

Due using the submission facility of the Blackboard Assignment facility: 11:59 pm, 8/05/14

One of the enhancements discussed in Assignment 1 was the addition of an auditing system to our hypothetical *Creative Tax Returns* system architecture. Auditing systems are complex, as they must be able to take event information from multiple sources in a way that is both scalable and secure.

Your tasks for this assignment are to

- Create a WAR package that simulates the customer management system discussed in Assignment 1. Create two basic HTTP servlets or JSPs that implement a login page for the system. Validate the user credentials against a static username/password pair, which you can store hardcoded in the code. Whenever a login is attempted the software will record if it was successful or failed.
- Create another WAR package that represents the administration console for the auditing system. This should consist of a startup servlet, which listens for auditing events and stores them, non-persistent, in an application bean. Another servlet/JSP will allow you to view the last 50 records stored in that bean.

Both of these WAR packages will be deployed on the same Glassfish J2EE application server. Glassfish will be used to mark the assignment, though you may choose to develop with whatever application server you like.

The two applications described above, in their respective WAR packages, will communicate auditing events via a publish/subscribe JMS queue (topic). Please name your connection factory and topic as follows

Connection Factory: jms/myConnectionFactory

Topic: jms/seng4400ass2PS

When a login event is recorded in the first application, have it send a string detailing that event to the queue to which the second application subscribes.

Additionally, you are required to:

- Create a third WAR that represents the billing system, which integrates *Creative Tax Returns's* solution for processing payments with the bank or Paypal. When executing, it should expose a SOAP web service over HTTP called 'processPaymentService'. This web service will expose two methods –

processCard(String cardNo) used for credit card payments using the bank; and processPaypal(String paypalID) used for PayPal payments. When this method is called, an event will be logged recording the card number or PayPal Identifier as appropriate to System.out.

This third application will execute in a separate instance of Glassfish. Enhance the customer management system application such that after a user successfully logs in they can provide either a credit card number or PayPal Identifier (but not both). Trigger a payment by having the customer management system call the new back-end system via its web service. For the purpose of this assignment, you can assume the web service is running on 'localhost'. You will need to configure your second Glassfish instance to listen on TCP port 8081 (so that all URLs in the WSDLs are consistent, to assist with marking).

Lastly, provide a document that includes

- A system diagram
- A high-level description of this diagram
- Any additional information the assessor may need to install, run and mark your submission.

The final deliverables into Blackboard include:

- Three WAR packages representing the three applications described above
- Three ZIPs containing the source code used to compile the WAR packages
- A document with the above design/install information
- Assignment cover sheet

Your submission should be made using the Assignments section of the course Blackboard site. Note that large uploads to Blackboard can take time and are best done from on campus.

A completed Assignment Cover Sheet should accompany your submission.

This assignment is worth 20 marks of your final result for the course.

M Wallis