E-commerce Coursework 2005

A web-based publications database

The purpose of your assignment is to develop a web-based system for building and searching a publications database.

What is a publication?

Publications take many forms such as papers in journals, papers at conferences, book chapters, internal technical reports, monographs and abstracts. The range of paper types can be easily seen by visiting the publications pages of research groups in the department or home pages of individual academics. Certain information is to be found in every publication (e.g. author name, title, year, place of publication) while some information is either optional or depends on the form of the publication (e.g. abstract, keywords, page numbers, journal volume number, name of conference, technical report number, ISBN for a monograph, publisher, keywords, etc). Your system should support as rich a set of information as possible.

Requirements

- The system should be accessible by a normal web-browser such as Firefox or Internet Explorer.
- Registered users must be able to add new entries. They must also be able to modify and delete entries that they have created.
- Any user should be able to browse or search the database by specifying such things as author names, title words or keywords. They should also be able to restrict the search e.g. by date.
- It should be possible to specify the order in which search results are returned (e.g. date order, alphabetical order of first author).
- The system should be secure, easy to use and sufficiently fast.
- The system should be robust in use and perform data validation where necessary.
- The system should be written largely using Java technologies covered on the course. It is acceptable to use non-Java technologies such as JavaScript for minor aspects of the system.
- The system should support initialisation and maintenance operations by users who have administrator permissions.
- The architecture should be based on a 3-tiered design.

Stage 1 (40 %) – hand in by Friday week 7 (April 15th)

Your goal in stage 1 is to experiment with the architecture of the system. The most important aspect is to ensure that all the software you plan to use is working, so that all you have to do in stage 2 is to fill out the content of the system. Putting together a distributed architecture for the first time is often the most difficult part of system development, so you should get started on this process as soon as possible by downloading those applications you think you may need and going through a process which has been termed "shooting tracer bullets". This means that you should test all communication paths in your distributed architecture. For example, if you plan to use a backend database (e.g. MySQL), a middle tier servlet engine (e.g. Tomcat), and a presentation tier based on templates (e.g. Velocity), you should ensure that you can successfully generate dynamic pages with at least some data being taken off the database. If you plan to use the Ant build tool (highly recommended), then install and use it from the very start and include this as part of your architectural testing phase.

Note that during the 2nd phase, you will either have to deploy your system on the departmental architecture (within the firewall) or deploy it on your own server to enable stage 3 (testing) to take place. You should therefore ensure that your system either makes use of departmental options (essentially, MySQL and Tomcat), or that it can be easily ported to use those components. For instance, you may decide to develop database components using Access locally, but deploy using the departmental MySQL server. For deployment, it is sufficient to use the departmental

MySQL server and your own installation of Tomcat. It is possible to use the departmental Tomcat server, but it may be easier for you to run your own (within the department) since you need to restart it after every code change.

This phase is your chance to compare and evaluate different options for e.g. presentation (JSP, XMLC, Velocity or something else) and data persistence (XML or relational or a combination of the two).

You should hand in a brief report (maximum of 6 pages) which contains the following sections only, in the order shown:

- *Introduction:* Here, you should merely clarify any requirements and state the interpretation you have made (1 paragraph)
- *Chosen architecture:*
  - explain the principal features of the architecture and why you chose it (1 para)
  - provide a figure depicting the detailed architecture of your system (it is not enough to simply show 1 box for each tier – you should produce something akin to the diagrams used in the case study)
  - include a table showing all software components used
- *Tracer bullets:*
  - explain what testing you did (1 para)
  - describe the alternatives you considered and why you rejected them (1 para)
  - describe the problems you encountered (1 para, or a table)
  - describe what tests you have carried out to ensure that your system will deploy within the Department, or, if you plan to deploy elsewhere, describe this (1 para)
- *Plans and timetable:* for complete design (1 page, max)
- *Subdivision of work:* amongst the team (brief para)

Stage 2 (40%) – hand in by Wednesday, Week 10 (May 4th)

This stage involves filling out the system using the architecture developed and tested in stage 1. The end result is a working system, ready for independent testing.

You should hand in a report (max 10 pages) together with a copy of your first stage report. Your stage 2 report should contain precisely the following sections:

- *Introduction:* describe the basic functionality of your system, indicating (using a table) the compliance of your system with the list of requirements above (max 2 pages)
- *Security:* describe briefly what security policy your application follows (1 para).
- *Architectural changes:* if you were forced to depart from the architecture described in stage 1, explain why and describe the new architecture.
- *Problems encountered:* describe the main difficulties you faced, how you overcame them, and roughly how much time you spent on their solution.
- *User's guide* (up to 4 pages including screenshots). This should be as clear as possible since it will form the basis for the testing phase. It should also be available online and easily accessible from your deployed application.
- *Administrators guide:* this should include any one-off or regular maintenance instructions, and should indicate where the source code and API documentation can be found. Don't include API documentation in your hand-in (1 page)

Stage 3 (20%) – hand in by Friday, Week 11 (May 13th)

Stage 3 involves testing each other's systems. Each of you, as individuals, will be allocated one system to test. This will take about one hour . You will fill in a questionnaire and return it to me. You will receive individual marks for testing the system (up to 10%). A further 10% will be awarded to each group based on testing reports.