

Java E-Commerce Group Project 2011-12

This document sets out the assessment rules for the Java E-Commerce group project, which contributes 100% to the assessment of the module. Please familiarise yourself with these instructions and follow them carefully.

Dr Anthony J H Simons

A.J.Simons@sheffield.ac.uk

9 February, 2012

Project Overview

The project will be conducted in stages, which have separate deadlines and for which credit will be gained according to the assessment weightings.

Project Stages

Stage 0: Team Selection. You must arrange to work in teams of 4 persons and email me the details of your team by the **end of week 3**.

Stage 1: System Architecture. Your team will develop the architectural design for an E-commerce system and write a short **interim report** by the **end of week 6** detailing the design and any tests conducted to demonstrate that this design actually works in practice.

Stage 2: System Development. Your team will complete the implementation of your E-commerce system, such that it is ready for testing by others, by the **end of week 10**. You will write a final report detailing how your system complies with the requirements, any problems you encountered and any architectural changes made, by the **start of week 11**.

Stage 3: Independent Testing. Individually, you will be allocated another E-commerce system to test, and you will complete a proforma describing the quality of the tested system. This must be completed by the **end of week 11**.

Assessment Weightings

Stage 1: 25% for the group interim report (physical copy + MOLE backup);

Stage 2: 65% for the stage altogether, of which:

25% for the software system (MOLE upload);

40% for the group final report (physical copy + MOLE backup);

Stage 3: 10% for carrying out the testing competently (using a webform).

Deadlines for each Stage

15.00, Friday week 3, Notification of teams due (by email).

15.00, Friday week 6, Stage 1 (group) interim report due (+ coversheet/barcodes).

15.00, Friday week 10, System implementation finished (MOLE upload)

15.00, Monday week 11, Stage 2 (group) final report due (+ coversheet/barcodes).

15.00, Friday week 11, Stage 3 (individual) test report due (via webform).

Late work will have marks deducted according to the **standard penalty scale** of 5% of the awarded mark, per working day late. Any student failing to pass may attempt a resit when the project is next run in the following year, according to the **resit rule for group projects**.

Stage 0: Team Selection

The group project **must** be undertaken in **teams of four** persons. Students may select their own teams, but the elected team leader must **email me** the details of your team **in the exact format** below by the **end of week 3**. After this date, teams may be imposed by me (and **only I** may authorise teams of a different size, if people are left over).

Leader: John Smith

John Smith, MEng Software Engineering, aca05jms@shef.ac.uk

Mary Jones, MSc Software Sys. and Internet Tech., aca05mpj@shef.ac.uk

Ahmed Ullah, MSc Advanced Computer Science, acs05amu@shef.ac.uk

Charlotte Leblanc, Visiting Erasmus Student, aca06chl@shef.ac.uk

Stage 1: System Architecture

Your goal in stage 1 is to design the architecture of your system. Putting together a distributed, multi-tier architecture for the first time can be time-consuming, so you should get started on this process as soon as possible, by downloading those applications you think you may need and testing all communication paths in your distributed architecture. For example, you should ensure that you can successfully generate dynamic pages with at least some data being taken off a database. A good (and useful) way to test some of your paths is to develop (a) a login component and (b) a mail component. You should compare and evaluate different options for e.g. presentation; and data persistence.

Eventually, your system will be deployed 'live', either on departmental servers or on some machine acting as a server elsewhere (you must then make your own arrangements). If you intend to deploy your system on the department's servers, make sure that your architecture matches the architecture offered by the department (using MySQL and Tomcat), or that it can be ported easily onto this architecture.

Stage 1 Deliverable: Interim Report

You will hand in an interim report, via the departmental hand-in post-box, with a completed and signed coversheet (submit a physical report, plus an electronic MOLE backup). Your group report should be **no longer than 6 pages** (12pt font) and consist of the following sections:

- Introduction – clarify any requirements that were vague and state the interpretation you have made.
- Architecture – explain the principal features of your architecture and why you chose it; provide a figure depicting the architecture in detail; include a table showing all software components used.
- Testing – explain what tests you conducted; describe the alternatives you considered and why you rejected them; describe any problems you encountered.
- Other aspects – e.g. security, availability
- Division of effort – describe which team members did which parts of the work. If any team member did more or less than expected, highlight this.

Stage 2: System Development

Your goal in stage 2 is to implement the design specified in stage 1. At the end of this stage, your E-commerce system must be complete, working and ready for testing. You will deploy your system, either on the departmental architecture, or on some other architecture using external servers managed by your team. Your system **must be accessible** to those who will act as independent testers in stage 3.

Stage 2 Deliverable: Final Report and Software System

You will hand in a final report, via the departmental hand-in post-box, with a completed and signed coversheet (submit a physical report, plus an electronic MOLE backup). You should also upload a zipped file containing your software system to MOLE. Your group report should be **no longer than 10 pages** (12pt font) and consist of the following sections:

- Introduction – short overview of the report.
- Compliance – describe the main functionality of your system; indicate using a table how it complies with the list of system requirements.
- Architectural changes – if you were forced to change the architecture described in stage 1, explain why and describe the revised architecture.
- Challenges – describe the main challenges and any unforeseen problems you faced and how you overcame them (or tried to).
- Users' guide – this should be very clear; use plenty of screenshots to illustrate the guide (e.g. 4 out of the 10 pages is about right).
- Division of effort – describe which team members did which parts of the work. If any team member did more or less than expected, highlight this.

Stage 3: Independent Testing

Each **individual** will be allocated **one E-commerce website** to test, and will be responsible for contacting the owners of the chosen website and for arranging a time to test their system. The tester will complete an on-line **proforma** detailing how well the system performed. Testing should take **no more than 1 hour**. (I will supply details of the proforma nearer the time).

Getting Help

You are expected to make full use of **online web resources** to handle any **questions about the technology** side of things; but you may **contact me by email** for any **clarifications to the requirements** for the project.

Online Resources

I have compiled a list of useful on-line resources about most of the technologies discussed in the lectures. Please navigate to my teaching page, lectures and Java E-Commerce:

https://staffwww.dcs.shef.ac.uk/people/A.Simons/campus_only/COM4514.html

The Self-Resourcing Electronic Journal

Science and engineering journals are moving increasingly from a paper-based format to an electronic format. Covering the costs of publishing an electronic journal is problematic: either, the publishers charge high subscription fees to academic institutions and restrict access to the journal (bad for the dissemination of scientific results); or they grant open access to all readers, but charge the authors a hefty fee for having their articles published (promoting “vanity publishing” by those who can afford it).

Outline of the System

The idea for this project is to develop a new kind of co-operative business model for an open access, online journal, which effectively pays for itself. The journal will still be free to readers, to foster the dissemination of results, but prospective authors will bear the administrative burden of running the journal in a different way. When an author is ready to submit an article for publication, he (or she) must then also undertake to peer-review other articles submitted by other authors. Every author is effectively also a reviewer (which is good academic practice). Articles may only proceed to publication if (a) they have been judged of an acceptable quality by peer reviewers; and (b) if the article’s authors have also completed their peer-reviews of other work, as “payment in kind” to cover the journal’s operating costs. Finally, a small group of editors must oversee the reviewing process, to make sure this is done to an acceptable quality. Editors have the final say before they publish accepted papers online. The purpose of the project is to develop a web-based system to support the running of an online journal and automate the process of submission, reviewing and publishing as far as possible.

System Requirements

The requirements are described below under separate headings for different stakeholders who have an interest in the way the publishing system is organised. The system should be so designed that a given person could play one or more roles of a **Reader, Author, Reviewer** or **Editor** (the roles may appear coupled; but are in fact independent).

General Requirements

- Each installation of the publishing framework should support the operation of **just one** journal; but it should be possible to tailor the system to a journal with any title, academic aims and goals, and presentation style, as chosen by the editors.
- The look-and-feel of the website for a journal should be consistent and attractive, using e.g. CSS stylesheets to maintain consistency across different web pages. Care should be taken to support accessibility for all readers (e.g. no fixed font sizes).
- The different stakeholder roles should be clearly separated in the way the website is organised and presented, such that readers do not necessarily have to interact with author, reviewer etc. parts of the site. In particular, there should **not be** a general requirement to have to login, unless accessing a restricted part of the site.
- The reader’s view should be the default view of the website, with other stakeholders having to navigate further to access their specific functionality.
- Whereas readers may browse the journal without supplying any login credentials, a secure login system should exist for authors (after submission), reviewers and editors.

Reader Requirements

- The journal should be organised along traditional lines, with one volume per year, consisting of several numbers (viz. editions of the journal).
- A typical journal will publish four quarterly numbers per year (but may publish more, e.g. having a special edition devoted to a conference).
- There should be a clear master index to the volumes and numbers, by year. By default, articles appearing in the same (virtual) number will have consecutive page numbers.
- It should also be possible to search for a particular paper by title (or a fragment thereof), or by author(s) name(s), or by date interval, or by a subject keyword search.
- Clicking on the title of an article should take the reader to the abstract page for that article, giving a summary of the article, email contact details for the authors, subject keywords and a link to download the PDF of the full article.
- Readers should be able to register for email notification about papers on particular subject keywords, or for forthcoming editions of the journal, when these are published.
- There should be a facility for readers to send letters to the editor to discuss particular articles, which may appear in edited form, with replies from the article's authors, in a later edition.

Author Requirements

- Prospective authors should be able to navigate to a part of the website describing the academic aims and goals of the journal (e.g. what subject areas are covered; whether the journal publishes early results, or fully mature work, etc.), with a link to details how to submit articles to the journal.
- There should be detailed submission guidelines, including document style templates (e.g. MS-Word or LaTeX stylesheets) describing the submission format expected for the journal, which may vary from one journal to the next.
- Authors should be able to download document style templates, create their article offline using their preferred word processing system then upload their articles in PDF format for review, using a web form.
- The information supplied with any article must include: full details of the names, emails and affiliations of each author, with one author chosen as the main contact; a title and an abstract for the article (the abstract may have a word-limit, e.g. 250 words); a set of keywords (max 10) describing the subject area of the article; and the PDF of the article.
- Successful submission should be acknowledged by a web page, and an email sent to the main contact author, giving details of their login ID and password, which is good for tracking the progress of this article and for selecting other articles to peer-review (see below).
- Login IDs and passwords relate to the unique email address of the contact author, which is maintained in a persistent database. The best login ID is therefore the email address of

the contact person. The same person may use the same login ID for later submitted articles, but access to the journal website will always be restricted to the (tracking; or reviewing) functions that this person is currently expected to perform.

- When the author receives the initial feedback from between 3 and 5 reviewers, he (or she) must revise the article, and submit a revised PDF, with an accompanying special form, explicitly detailing how all of the bad points highlighted by reviewers (see below) have been addressed in the revision. At least 3 peer-reviews must be completed for each submitted paper, to judge its quality; and 5 reviews are sufficient.
- The author's article may eventually be accepted by the reviewers and the editors; but the author must still complete three peer-reviews of other work, as a kind of contribution to the cost of running the journal, to allow his (or her) own article to proceed to publication.

Reviewer Requirements

- There should be a part of the website available only to reviewers, giving clear instructions regarding how to peer-review pending articles, describing what qualities are sought and how to complete the review form.
- A reviewer must login with his (or her) ID and password to obtain access to the part of the website containing unpublished articles that are awaiting peer-review. This is a secure system, preventing the reviewer from casually reading just any or all of these articles. Eventually, three articles must be chosen explicitly and peer-reviewed.
- If certain articles have been awaiting review for a long time, the system may force the reviewer to select one or more of these, in preference to any other more recent submissions to the journal.
- A reviewer may browse pending articles but initially may only read their abstracts, before deciding whether to review the article, by checking a box next to that article. Until the articles are downloaded, the reviewer may change his (or her) mind about which articles to review.
- The system will keep track of how many articles the reviewer has selected so far for review, and will display these titles to the reviewer, along with their review status (selected, downloaded, or review submitted). Periodically, it will send an email reminder to the reviewer, to select more articles, or to submit reviews for chosen articles.
- Once the reviewer has selected one to three articles for review, he (or she) may download the draft PDF versions of those articles, after which he (or she) is committed to review those articles and may not choose to go back, in favour of some other article.
- The reviewer may read their chosen articles for review offline, but must not divulge their contents to any third party, since being a reviewer puts you in a privileged position of seeing other work early, in an unfinished form. Later, they must submit their review(s) of article(s) using a particular review form.
- The review form should contain: an overall judgement (using the “champion/detractor” pattern – see below); the relevant expertise level of the reviewer (on 3-point scale: **expert, knowledgeable, outsider**); a summary of the content and novel contribution of the article (as proof that the reviewer has understood it and believes in its good points); and a structured criticism of any bad points (issues that the authors must explicitly

address in their revision). Each substantive criticism should be described in a separate section of the form, to help editors track authors' responses to the same criticisms. Finally, a detailed list of small errors should be provided for the authors (typographical, or grammatical mistakes, etc.).

- The review form should also contain an optional section, allowing the reviewer to send secret comments to the editors, which will not be seen by the authors. This is in case they need to say rather impolite things about very poor work, plagiarism, or other kinds of bad practice.
- Successful review submission should be acknowledged by a web page and an email sent to the reviewer, detailing the content of the review. A reviewer should be able to track reviews submitted, and submit a revised review (within a limited time period – say one week), after which the review can no longer be changed, and it is forwarded by the system to the author(s). All reviews are anonymous, so the authors must not be told the identity of the reviewers!
- Eventually, each reviewer will receive from the authors (via the system) a list of detailed corrections describing how he (or she) addressed each of the criticisms highlighted in the review. The reviewer may choose to accept, or reject these changes, by checking off those criticisms that were successfully addressed.
- If accepting, the reviewer may revise the overall judgement on the paper (according to the champion/detractor model). If rejecting, they must explain why the revision is not adequate, and may explicitly highlight further bad points to address in a form.
- A reviewer will not expect to have to go round the corrective and confirming cycle more than twice. If he (or she) still rejects the article after two attempts, it is likely that the article should be rejected overall.

Editor Requirements

- The editors are the people who are nominally in charge of running the journal and who host the electronic journal on a website. They must login with their ID and password, which grants them access to the whole site as editors. Their ID identifies them as a current editor.
- There will be at least one, and possibly many, editors. Any editor may choose to retire, or appoint another person as an editor, so long as at least one editor still exists.
- The editor(s) decide upon the title of the journal, its academic aims and goals, and the styles and templates to be adopted for publication. They have an interface which allows these sections of the website to be populated with this standard information.
- The editor(s) have access to all the articles; and all the reviews for each article, and to each review form with criticisms, and each author-response to a review. The system must track these, so that it is possible to see the revision history of an article.
- The editor's main jobs are to ensure the sensible and fair running of the journal; and to publish articles online, once they are of an acceptable quality and the authors have also fulfilled their peer-reviewing obligations. Fair running is explained in the next four points below.

- The editor may choose to reject a reviewer's choice of paper to peer-review (because of a conflict of interest in a closed group of authors and reviewers, who mutually support each other's work uncritically). In this case, the reviewer must select a different paper to review.
- The editor may choose to reject an overly brief, hurried review (that was clearly done badly, just to get it out of the way and advance the reviewer's own paper). In this case, the reviewer must resubmit the review and put proper effort into it.
- The editor may choose to reject an author's revision to an article, which only pretends to include the changes listed in the response to the reviewer(s).
- The editor may choose to review an article that has been awaiting review for a long time, just to move it through the journal system.
- Eventually, the editor looks at the reviewer's confirmation forms, detailing how they (mostly) accept the corrections made by the authors. If the paper is publishable according to the champion/detractor model (see below), then the editor may publish it officially (as part of the current volume/number of the journal).
- The editor(s) may add published letters to the end of any edition (number) of the journal. These are like short articles, consisting of comments and questions by a reader, followed by the response of the original authors. This interaction may be conducted by email, or by web forms, so long as the editors have final say on how the letter is published.

The Champion and Detractor Pattern

This is a kind of pattern for running editorial panels, originally invented by Dr Oscar Nierstrasz from the University of Berne, Switzerland, as an alternative to the "points scoring" system used on certain conferences and journals. It works better, because it doesn't rely on average scores and more accurately reflects what reviewers feel about articles.

In the champion/detractor model, a reviewer is only given four choices. He is asked whether he is a **champion** who strongly advocates the paper; or a **detractor**, who strongly condemns the paper as badly flawed or worthless. If neither of these positions is chosen, then he must choose one of the less strong positions, whether he is **favourable** and would not object if the paper were published (but will not fight strongly for it); or on the other hand is **indifferent** and would not care if the paper were rejected (but will not fight strongly against it).

Papers with two or more champions and no detractors should always be published. Papers with two or more detractors and no champions should always be rejected. Papers with both a champion and a detractor are interesting, because of the conflict, and the editor should get the reviewers to discuss it and come to some conclusion, usually by one reviewer weakening his position.

Papers with one champion and at least one weak support should eventually be accepted, after all reviewer comments are carefully addressed. Papers that end up with one detractor and at least one weak rejection should be rejected. Papers with only uncommitted judgements are boring and the editor should ask the authors to resubmit the paper to make it more relevant. See the website: <http://scg.unibe.ch/download/champion/> for further details, especially the pattern called *Make the Champions Explicit*.