D09 – Functional testing

This document describes Deliverable "D09 – Functional testing", whose deadline is March 27, 2015 at 23:59h. Please, recall that you must produce your deliverable and upload it to the USE's e-learning platform by the deadline; failing to produce your deliverable in time amounts to failing the subject.

We, the lecturers, assume that if you submit this deliverable, then you understand what the consequences are if you lie regarding the following items, where "I", "me", "my", and other first-person pronouns refer to you, the student who is reading this document:

[x] I'm the legitimate author of this deliverable; I'm not cheating.

[x] If I have some partners, then I've collaborated with them on producing this deliverable; in other words, neither am I riding their coattails nor gobbling them up.

[x] I've learnt from working on this deliverable, so that I can pass my control checks.

[x] I've organised the deliverable according to the guidelines that are available in document "On your deliverables.pdf", which is available at the USE's e-learning platform.

[x] I've made sure that this deliverable fulfils the requirements to get a/an A (C, B, A, or A+).

[x] I fully understand that my deliverable will be considered failed if I fail to meet any of its requirements, or if I do not deliver a filled, dated, and signed copy of this document.

[x] I fully understand that failing this deliverable amounts to failing the subject.

(Please, check the previous items and write the level of your deliverable. It's compulsory that you fill in this form and that you date and sign it; otherwise, your deliverable won't be evaluated, which amounts to failing the subject.)

https://repositorio.informatica.us.es/svn/gq7l9tnvlygkwnmr6aw

URL of your deliverable in ProjETSII

Seville, 27/03/2015 — Place, date

32 — Group number

Juan Carlos Pérez García — Julio Carmona Ferri — Raúl C. Díaz León — Javier Carmona Ferri




Authors' names and signatures

Instructions

Please, select a level and produce the items that are specified below:

- Level C: Items 1-4 regarding the C-level requirements of project “Acme Pilgrim 2.0”.
- Level B: Items 1-4 regarding the C- and B-level requirements of project “Acme Pilgrim 2.0”.
- Level A: Items 1-4 regarding the C-, B-, and A-level requirements of project “Acme Pilgrim 2.0”.
- Level A+: Items 1-4 regarding the C-, B-, and A-level requirements of project “Acme Pilgrim 2.0”, plus Items 5 and 6.

The lecturers will evaluate your results as follows:

- Regarding laws:
  - They’ll check that your project complies with the minimum requirements that we’ve presented in the lecture notes about the following laws: LOPD, LSSI, and Transpositions, except for keeping your files and communications secure and confidential.
- Regarding management:
  - They’ll check your project in ProjETSII. They’ll check that you have a project, and that you’ve created and reported on the appropriate tasks. Don’t forget to report on the time that you spend at your lectures, at co-ordination meetings, or at studying your lessons.
  - They’ll pay special attention to checking that the tasks were created at reasonable moments and that the reports happened at reasonable moments. In other words, a project in which every task and report was created on the last day will not be accepted.
- Regarding documentation:
  - They’ll check that you’ve produced a document with an estimate of the total number of hours you’ve spent in this project and the total cost expected.
  - They’ll check that you’ve produced a good conceptual model.
  - They’ll check that you’ve produced a good UML domain model.
  - They’ll check that you’ve produced comments in your code where appropriate, e.g., to document complex queries.
- Regarding JSP views:
  - They’ll check that you’re using customs tags to make your views more compact and less error prone.
- Regarding query efficiency:
  - They’ll go through your repositories and check that you’ve defined appropriate indices for your domain entities.
- Regarding hacking:
  - They’ll go through your services to check that you check the principal in order to prevent GET hacking.
  - They’ll analyse your views and form objects, as well as your controllers and services to check that you’re preventing POST hacking.
  - They’ll make sure that your forms do not suffer from SQL injection or cross scripting.
- Regarding functional testing:
  - They’ll check that you’ve followed the guidelines that we’ve provided regarding how to organise test cases, test classes, and test suites.

    - They'll run your test suite and will check that JUnit does not report any exceptions, that is, it must show a green bar.
    - They'll go through your test cases and will check that you have at least a positive and a negative test case per functional requirement or group of related functional requirements. **IT IS MANDATORY THAT YOU DOCUMENT EACH TEST CASE**; Otherwise the lecturers won't be able to find out what they are intended to test. Please, copy the functional requirement(s) that it is intended to test as a comment at the beginning of your test cases; add a short remark regarding what the test case is intended to test.
    - They'll check that you've followed the guidelines that we've provided regarding how to design test cases.
    - They'll check that your test cases are completely automatic, that is, that they do not output any information to the console and that every check is performed by means of the appropriate assertion.
- Regarding the requirements:
    - They'll run your project as indicated in document "On your deliverables.pdf", which is available at the USE's e-learning platform.
    - They'll check that you've implemented every information and functional requirement well.
    - They'll enter as much invalid and malicious data as possible in the edition forms, just to check that you keep them under control.

Requirements

Item 1. Deliver a report with the tasks that you've accomplished to produce this deliverable. The report must list the name of every task, the moment when it started, the moment when it finished, and the number of hours spent on it. It must also report on the total time spent in the project and the total cost.

Item 2. Deliver a conceptual model and a UML domain model regarding your project.

Item 3. Deliver an Eclipse/Maven project that fulfils the requirements in the Acme Pilgrim 2.0 project statement.

Item 4. Deliver a script to create the corresponding database in the pre-production environment and a war artefact that implements your project and can be run on domain "www.acme.com".

Item 5. Testing domain objects, converters, repositories, and services is relatively easy with the theory that we provide in this lesson. Unfortunately, testing controllers is not easy at all. Currently, there are a number of frameworks available to test controllers, or, more generally, servlets. Choose the servlet testing framework that you prefer, learn to use it, and deliver a version of your project in which you include at least a test case to test a controller.

Item 6. Write a report in which you explain what the lecturers have to do to check your project. Please, report on the testing framework that you've chosen and comment on the test case that you've implemented. The report's expected to be 3 000-word long; illustrations are strongly recommended.