COMP3207-1516-cw2-spec

From NotesWiki

Contents

- 1 COMP3207 2015/16 Group Coursework Assignment Specification
- 2 Overall
- 3 Scenario
  - 3.1 Source Code Size Restriction
- 4 Key dates
- 5 Submission
- 6 Handing in
- 7 SCB submissions
- 8 Relevant Learning Outcomes
- 9 Assessment
- 10 Marking Scheme

COMP3207 2015/16 Group Coursework Assignment Specification

- Coursework two - "Scripting Demo Prototype"
- Deadline: 1600 on Monday 4 January 2016 (the final week of the semester)
- Presentation: all day Monday 11 January 2016 (all students **MUST** be available this day)

  and all day Tuesday 12 January 2016 (all students **MUST** be available this day)

- Feedback: by Monday 1 February 2016
- Effort: 75 hours per student
- Weighting: 60% of module evaluation

Overall

This group assignment is designed for you to demonstrate your ability to use Python and JavaScript effectively. You will be working as part of a team of seven (or possibly six) students and the marks for this assignment will be awarded as agreed by each team.

Scenario

The class will organise itself into teams of seven (possibly six) students - these teams are in "commercial competition" and should not discuss their work with other teams. The teams are bidding for a commercial contract to be awarded by "Southampton ECS Enterprise Developments"

(SEED), a recent start-up company with ambitious plans for a new product. The precise details of the product are a closely-guarded secret at present, because SEED is concerned that a competitor might launch a competing product.

SEED is aware that it will need to hire a development team with expertise in Python and Javascript to develop their proposed product and wishes to assess the competence of the available teams before awarding the contract. SEED requires each team to provide a "Scripting Demo Prototype" to demonstrate their expertise with these scripting languages and will award the contract to the team that demonstrates the highest level of expertise. It is mandatory that these two defined scripting languages must be used in this demonstration prototype:

- server-side Python
- client-side Javascript

The exact form of the prototype is left open to the competing teams, except that the client application must execute inside ECS and the server-side Python must execute outside ECS on the Google App Engine (N.B. the use of Java is explicitly **prohibited**).

Source Code Size Restriction

This restriction is in terms of LOC ('Lines Of Code')- see Wikipedia (http://en.wikipedia.org/wiki/Source_lines_of_code) . The definition adopted here is physical LOC excluding comment and blank lines. A tool such as CLOC (https://github.com/AlDanial/cloc) will be used to measure the source code size.

The target size for this coursework is 2000 LOC (more details and penalties for code sizes greater than an additional 120% to follow).

Key dates

- 1600 Friday 9 October 2015, team membership to be complete (see the special page on the student wiki here (https://secure.ecs.soton.ac.uk/student/wiki/w/COMP3207-1516-cw2teams))
- 1600 Monday 4 January 2016 submission deadline
- (all day) Monday 11 January 2016 presentations (all team members **must** be present)
- (all day) Tuesday 12 January 2016 presentations (all team members **must** be present)

Note that late submission will be penalized using the standard University rules (10% per working day) and that no work will be accepted that is more than five days late.

Note also the University Regulations Governing Academic Integrity (http://www.calendar.soton.ac.uk/sectionIV/academic-integrity-regs.html) and Academic Integrity Guidance (https://secure.ecs.soton.ac.uk/notes/comp3207/1516/ejz/Academic_Integrity_Guidance.pdf) .

Submission

All source files together with a brief report must be submitted as a ZIP archive to the ECS login server. The report must be provided in .pdf form and use a font size of 12 points or larger. The report is to have a filename "team_x.pdf" (where "x" is the letter identifying the team) and the following contents:

- Page 1 Description of prototype functionality
- Page 2 List of tools and techniques used
- Page 3 Relevant statistics (e.g. lines of code written, plus an assessment of code taken from acknowledged external sources - provide a list giving sources)
- Page 4 Brief overview of design and implementation, including key design decisions
- Page 5 Critical evaluation of the prototype submitted

Only these five pages may be submitted (with an optional cover page): marks will be deducted for any extra pages included in the report. Please ensure you include the link to your prototype on appspot.

Handing in

Please submit using the ECS hand-in system by the deadline set above. You should submit the ZIP file specified above, together with a printed copy of the five-page report and a "Coursework Marks Distribution" form. This form must be signed by all members of the team and propose a marks allocation to individual students. Please note the University Academic Integrity Statement for Students (http://www.calendar.soton.ac.uk/sectionIV/academic-integrity-statement.html) when undertaking this coursework and making your submission.

Further details on the submission arrangements together with the presentation and demonstration arrangements will be supplied later (as a separate link from the main group coursework page).

SCB submissions

Deadline extensions are not available for this coursework. Any student that is affected by special circumstances should apply to the Special Circumstances Board in the normal way - if the case is accepted by the SCB then an appropriate alternative assessment will replace this assignment.

Relevant Learning Outcomes

- Design and implement a scripting application.
- Explain (and justify) the choice of technologies and techniques
- Evaluate the effectiveness of your submission

Assessment

The presentation and demonstration will form an important part of the assessment and will contribute approximately 50% of the marks available. The remaining marks will be awarded based on the report submitted and the quality of the code written.

Marking Scheme

Criterion	Description	Marks
Working system	Demonstration of working system	50%
Well-designed system	Report on working system	25%
Well-written code	Software quality of code submitted	25%

EJZ 28 September 2015
