

Assignment Two (A2): GUI Prototyping and Lifecycles (Individual Work)

Liam O'Reilly and Alena Denisova

Number of Credits: 30% of a 15 credit module

Recommended Hours: approximately 30 hours

Electronic and Submission Deadline:

Wednesday November 22nd, 2017 @ 11am

Learning Outcome(s): To practice some requirements analysis and GUI design.

1 Development Models

a) Briefly discuss the primary advantages and disadvantages of the Waterfall model and agile models in software engineering. As you do so, describe two specific agile methods (**at most 100 words (3 marks)**).

b) Describe situations where the Waterfall model and agile methods are appropriate. These situations should draw on real application areas of software engineering (i.e., any area that requires software) that *have not been covered in class or this assignment*. Application areas covered in class or in this assignment will not receive credit (**at most 300 words (7 marks)**).

Note: After your answer, please write your word count into the document. Answers above the specified word counts will have marks deducted for being too long.

2 Use Case Diagram

Consider the following system:

Whiteboard is an electronic student marking system used by academic staff to manage students' grades. Each student has a student number, first name and last name. Academic staff can see a list of students enrolled on a specific module. Academic staff can also search for a student and view what modules that student is enrolled upon. Each module has a member of academic staff listed as a module coordinator, and a number of weighted components. Each component has a name and a weight (e.g., Coursework 1, 30%). The sum of all components, for a given module, should be 100%. Each module also has a module code and a module name.

Students can view and enrol on available modules. When a student enrolls on a module, the member of academic staff coordinating the module is notified. Academic staff can also view each students' grades for a module, which are calculated as a weighted average from all components of the module. Academic staff can also enter (or edit) grades for each student and component.

Draw a use case diagram for the system described above. The diagram can be neatly drawn by hand or using various computer tools. **(10 marks)**

Note: Submission is one use case diagram that fits on a single A4 page. Answers more complex than this will have marks deducted.

3 Requirements

Whiteboard Inc. (the company behind *Whiteboard*) is really interested in seeing a few prototypes for the interface that *academic staff* see when navigating the system. Upon meeting with one of the executives of the company, you discover that the following is also required of the system.

- *Strings entered through text boxes in the interface should not break the system (e.g., string that contain quotes, or apostrophes, or even SQL strings that could be interpreted as commands).*
- *The system needs to be responsive. Using a standard broadband Internet connection, the system should respond in no more than 5 seconds.*
- *The company's colours are grey and yellow. If you can work that into the page, that would be nice.*

Using the above requirements and the requirements described in Section 2 do the following:

1. Name and describe a user requirement and a system requirement. Justify both with a short explanation.
2. Name and describe a functional requirement and

a non-functional requirement. Justify both with a short explanation.

Your answer should be no more than **300 words (10 marks)**.

Note: After your answer, please write your word count into the document. Answers above the specified word count will have marks deducted for being too long.

4 Prototype

Now that you have studied the problem a bit, Whiteboard executives would like you to build some low fidelity prototypes of the interface *academic staff* will use as specified in Section 2.

In the design of this prototype, Whiteboard Inc. asks you to follow the following procedure and provide the following documentation.

4.1 Scenario

Write down in natural language a scenario of how the client interface could be used following one path through the use case diagram. This scenario should provide details on a real use of the system as discussed in class (**at most 100 words (4 marks)**).

Note: After your answer, please write your word count into the document. Answers above the specified word count will have marks deducted for being too long.

4.2 Design

Create **three distinct** GUI designs that support the **client** interface that *academic staff* will use. Your GUI designs **must** consist of a maximum of **two** windows each. With a few lines of text (**no more than 50 words for each design**), you should specify how each design supports the requirements. Your three GUI designs can be neatly drawn storyboards, wireframes (hand drawn or using computer tools), or mockups (using computer tools). You can use any program you wish. Extra initiative earns higher marks.

The prototype should be as visually realistic as possible to achieve a first class or high first class mark. In this section, you should be creative (**21 marks**).

Note: Your answer to this question consists of at least 1 to 2 diagrams for each design and text supporting each diagram. After each design, please include the word count for the text supporting it in the document. Answers above the specified word counts will have marks deducted for being too long.

4.3 Prototype

Given the above description, there is an interface for the electronic student marking system and an underlying subsystem that manages data such as students' grades and module enrolments. In this section, do the following:

a) Create UML class diagrams and collaboration diagrams that model the electronic student marking system and the classes that accept commands from the academic staff persona. You can ignore all other personas in the system in this design. (**10 marks**).

b) Given your design, it is clear that your system must support browsing functionality that allows viewing of all students in numerical order (student number, first name, last name). By selecting a student one can view all modules they are enrolled upon (student number, module code, module name). By selecting a module one can view the student's grades for each component and the overall grade for the module. The system should also allow the user (i.e., academic staff) to add grades to each component and make amendments to the existing grades, which should result in recalculated weighted average mark for the whole module. These operations must be supported by the underlying subsystem.

Do not implement any of the GUI. However, provide a Java implementation of all the classes in this subsystem and select an appropriate data structure. Then, provide an implementation of the above described methods. Your implementation should follow your design closely. Correctness of your java code will be assessed. Unfocused solutions that contain operations or data unrelated to this specific problem will lose marks (**25 marks**).

Submission: Provide UML class diagrams of your design of the above problem and correct java code for the required implementations. Your design must fit onto a single two sided sheet of A4. Your implementation should be a number of .java files.

5 Submission

Below, you will find more detailed submission instructions.

5.1 Submission Instructions

Submission is to Blackboard. Submission via Blackboard follows the normal procedure. For parts of the

submission that are parts of the report, the required format is a single .pdf document. Word document format .doc is **not** acceptable.

For source code, .zip is acceptable. Do not use rar (.rar) format as this is not platform independent. If you submit a rar you will lose marks.

5.2 Submission Summary

Each individual is required to submit the following:

1. **Written Report (65% of A2 (55% content, plus 10% style)):** An electronic submission to Blackboard of your written report is required. The report consists of your answers to Sections 1, 2, and 3. It will also contain parts of the design in Section 4, but not the implementation. The file must be named “*StudentNumber-LastName-A2-report.pdf*”, where *StudentNumber* is replaced by your student number and *LastName* is replaced by your last name. PDF format is required. Word document format (.doc) is not acceptable. *Word count limits are strictly enforced as described in each section above.* Attention: you will be assessed on your conformity to the instructions. Don’t forget to write your name and student number on the report.
2. **Software Design and Prototype (35%) of A2:** The prototype itself must also be submitted to Blackboard electronically. The file should be named “*StudentNumber-LastName-A2-prototype.zip*”, where *StudentNumber* is replaced by your student number and *LastName* is replaced by your last name. We recommend zipping all java files together and submitting them as one file (only include the java source files). The source code is located in the project directory of eclipse. You can simply zip up all the .java and project files of your prototype’s directory.