

Software Engineering with Objects and Components 1

Group Tutorial Project: Deliverable 2

Detailed Design and Preliminary Construction

Your teams have submitted requirements documents for your facets of the Secure Submission System. The second project deliverable takes this forward to begin the construction process for your proposed system. The main elements for the submission for deliverable 2 are:

1. A prioritisation of the use cases you generated in Deliverable 1 of the project selecting those you want to concentrate on in the first iteration of the software construction process. You should include a short justification for your decisions.
2. Collaboration diagrams that demonstrate how these use cases could be realised in your class diagram (suitably augmented with appropriate attributes and methods). You should document the mapping from use cases to collaboration diagrams
3. A class diagram augmented with appropriate attributes and methods.
4. Appropriate tests for at least two non-trivial classes in your design (this should be agreed with your tutor).
5. Java implementations of your two selected classes that is intended to pass your supplied classes.
6. An outline of how you would test the integration of your two selected classes and a plan of how you would propose to test the integration of your design with the designs of the rest of your tutorial group.

Deadline for Deliverable: 4pm Mon 8 Dec 2003

Documents to submit

Your team should submit: an assessment proforma that will be available on the module web page, and the documentation specified above. As for the first deliverable, the assessment proforma records your tutorial group, team details and contribution given by each team member. The requirements on each element of the deliverable is given in more detail below.

Prioritised Use Case Model

You should provide a (potentially) revised version of the use case model from your first deliverable. You should take account of any feedback from your tutor and modify the use cases appropriately. Then in discussion with your tutor you should identify a small subset of the use cases that you think makes a sensible basis for constructing a first iteration of the implementation. If possible this set should be coordinated with the other teams in your tutorial group.

Collaboration Diagrams and Augmented Class Diagram

For those use cases you have identified as a priority you should carry out more detailed design involving the enrichment of your class diagram with attributes and methods and a collection of collaboration diagrams that correspond to the use cases. The collaboration diagrams should illustrate in detail how a particular use case can be realised by a system structured according to your class diagram.

Tests

Given your prioritised set of requirements, choose at least two (non-trivial) classes in your class diagrams as candidates for implementation. *You should confirm your choice of classes with your tutor.* You should choose two classes that are linked in the sense that in at least one of the use cases the classes are required to collaborate to generate the correct result.

Having determined the classes you intend to construct, define unit tests for these classes and implement them in an appropriate context (e.g. you may want to use the JUnit framework). The tests should be traceable back to a particular use case and should also be traceable to the appropriate collaboration diagram. The tests should be generated before generating any code and should attempt to be reasonably comprehensive – if an implementation passes these tests it should be close to acceptable implementation of the class. The documentation for the tests should specify what the code is testing and what the expected outcome is for each test. The test suite should comprise a collection of Java programs that exercise a given test class.

The justification for the adequacy of the test suite is an important part of this deliverable. You should write a short narrative justifying the design of your tests and how they relate to the appropriate collaboration diagrams.

You should email your tutor your tests and the Java code you have developed at or before the deadline for deliverable 2.

Java Prototype

You must provide Java classes for at least two of the classes from your first deliverable's class model. These classes should be involved in realising the use cases you have identified as a priority. Your goal is to produce a working prototype for the classes that can be analysed as the basis for a next iteration of the design/construction process

The Java classes you define should pass the tests you have defined as part of this practical.

Integration Plan

You should include a plan on what testing is required to assure the integration of your classes into the overall design. This should specify what the integration test should be directed towards and how it should be tested.

Assessment

Marks will be awarded by team, with individual team members receiving a proportion of marks in relation to their contribution as for deliverable 1. Marks for this deliverable constitute 50% of the practical marks for this course. The marks for this part of the practical constitute 12.5% of the overall assessment of this module.