**Software Engineering with Objects and Components**
**Group Tutorial Project: Deliverable 1**
**September 2006**

# Requirements - *Secure Coursework Submission System*

Your software company (tutorial group) has now received initial requirements documentation from *The School of Informatics.* You have separated into three teams, each dealing with different facets of the overall system tendered by the *School of Informatics.* In this phase you will have had the opportunity to gather and clarify the requirements for the system. Your first project deliverable is a report to the *School of Informatics* detailing your initial requirements specification (incorporating Use Case Models), a preliminary outline of your proposed implementation (incorporating Class Models) and supporting evidence validating that specification (e.g., using CRC cards).

## Deadline

The deadline for the deliverable is:

**Deliverable 1: 3pm Monday, 30th October**

### Submit your work to the ITO, JCMB, Room 1502.

# Documents to submit

Your team should submit a complete deliverable 1. Deliverable 1 consists of:

1. Requirements Specification

2. Use Case Model

3. Class Model

4. Validation of Class Model

5. Deliverable Assessment

The remainder of this section describes each part forming deliverable 1.

## Requirements Specification

Your overall document should be structured according to a requirements specification template (a requirements specification template will be available online). The agreed requirements are documented at an appropiate level of detail. In general, there needs to be a requirements document which is understandable by all system stakeholders. You need to tailor a requirements specification template (e.g., the Volere template) to your project. The requirement specification should clearly identify the overall system objectives, distinguish between functional and non-functional requirements and highlight potential future requirements or open issues. The requirements specification forms the basis for your UML design and Java implementation. The UML components should conform to the identified requirements.

## Use Case Model

You should provide a collection use case diagrams where the primary actors are in your aspect of the system. Try to be as comprehensive as possible, it is likely that there will be some overlap in use cases considered by each team. Include actors from all the relevant aspects of the system where they interact with the main actor drawn from your facet of the system. Your use case diagrams should be accompanied by informal descriptions of the tasks they specify. These should be brief (typically a few sentences) and informative.

## Class Model

You may provide more than one class diagram, but ensure that all classes and associations in your class model are included. Class attributes and methods should be provided, but bear in mind that you do not as yet have sufficient information to define all of these precisely. It is your responsibility to clearly identify any *speculative* assumptions and design decisions you may have made.

## Validation of Class Model

You must justify that your proposed class model can realise your use case model. You may use any suitable technique to identify which classes (and their methods) collaborate in realising each of your use cases, although the use of CRC cards is the recommended technique. Furthermore, you must clearly set out how your chosen technique validates the class model. Again, you must clearly identify the impact of any speculative assumptions on these realisations.

## Deliverable Assessment

The deliverable assessment form records the software company (tutorial group) to which you belong, and your team details (team responsibility and membership). In addition there is a *team-contribution form*, which allows the team to assess the level of contribution given by each team member. This should be arrived at in the team by consensus. Details of this form, and how it affects individual marks, are given on a separate handout (blank deliverable assessment forms will be available from the SEOC practical web page). Marks will be awarded by team. Each part of the deliverable contributes to the final mark. Allocation of marks to individual team members will be as described in the separate handout. While your reports are assessed individually, it is expected that all groups' reports will be consistent (e.g., wrt terminology, shared use cases, shared classes, etc.). Marks for this deliverable constitute 50% of the practical marks for this course. That is, 12.5% of the overall course mark.

<div align="right">

Massimo Felici
September 29, 2006

</div>