

“Statement of Work” for

CA314 Continuous Assessment Assignment (‘08-‘09)

Table of contents:

1. Introduction.....	1
2. Assignment Deliverables & Due dates/Deadlines	2
3. A few guidelines for activities & deliverables.....	2
4. References.....	3

1. Introduction

The continuous assessment element of CA314 in 2008-2009 will be based on a single project assignment to be carried out and reported on throughout the semester.

The assigned project is a reasonably substantial one and, as such, is to be done (thoroughly) in teams. See October 16th email to class).

NOTE that, to work effectively, teams of this size must be well organised and business-like in their approach. It is required that each team hold regular, frequent meetings, and records of these meetings are to be submitted as part of the “deliverables” for the assignment. Records should at least summarise the current status of the team’s work, any decisions made at the meeting, and plans to the next meeting. It is expected that all team members will contribute equally to the work of the team but if this is not the case the team should inform ltuohey@computing.dcu.ie as to the approximate proportion of work contributed by each personⁱ.

The software requirements for the project, based on /1/, are specified in a separate document /2/.

The tasks making up the assignment are detailed in Section 2, as well as the deliverable items and their deadlines for submissionⁱⁱ. It is expected that any required UML modelling will be done with the aid of a software tool – there are two such tools on the lab machines, *WithClass* and *Rational Rose*, although a different tool may be used if desired. The work is divided into two phases or **iterations**:

Phase	Iteration 1: (Basic system implementation, System tests, ...)	Iteration 2: (Implementation of system refinements, detailed verification, ...)	Total C.A.
Mark	15%	10%	25%

It will be seen from /2/ that a great deal of the requirement analysis phase, including definition of use cases and identification of domain level classes, has already been done. Therefore the project consists primarily of detailed software design and programming, as well as software verification (mainly testing). In addition, there is considerable “project management” work involved in organizing a team, planning the work, tracking progress, coordinating team members’ work, etc.

Each team member should read the project software requirements specification and create a list of questions and ambiguities that arise from it. The statement should be read multiple times until its basic functionality can be discussed without excessive referral to the written statement. After all team members have had the opportunity to familiarize themselves with the requirements, each team should meet to discuss the statement and address items like the following:

- Identify and address questions and ambiguities
- Identify project elements requiring graphic image manipulation
- Identify project elements that lend themselves to audio (if any)
- Identify project elements requiring special algorithm creation
- Identify project elements requiring network programming (for **Iteration 2**)
- Establish areas of interest on which individual team members would like to work

2. Assignment Deliverables & Due dates/Deadlines

Devel. Phase	Deliverable	Due Date
Iteration 1 <i>basic system implementation ...</i> <i>use case testing</i>	1. Reference /2/ identifies the specific elements to be implemented and the mandatory testing to be done in this phase. The deliverables are, essentially, - documentation of software design - source code for implemented parts - test case specification and results	Deliverables to be submitted on Fri. Nov. 21st, 2008 at 17.00
	2. Minutes or notes of any team meetings	
Iteration 2 <i>refined system implementation</i> ... <i>detailed test & verification</i> <i>live demo</i> ...	1. Reference /2/ identifies the specific elements to be implemented and the other tasks to be done in this phase. The deliverables are, essentially, - documentation of software design changes and additions - source code for newly implemented parts as well as any changes - test case specification and results – this will be more extensive than for Iteration 1, including also unit and integration test reporting Note: There is scope for more ambitious teams.	a) System to be demonstrated during week beginning Mon. Dec. 15th 2008 at 17.00 b) Deliverables to be submitted on Fri. Dec. 19th 2008 at 17.00
	2. Minutes or notes of any team meetings	

Figure 2-1: Deliverables and Schedule

3. A few guidelines for activities & deliverables

- (Section 7.9 of /3/) “[Each team should define] programming standards for [their assignment]. Responsibility for each class design should be assigned to individual members of the project team. Ideally, all students should implement the classes they designed but ... [some] adjustments may need to be made. Assign responsibility for

implementing classes in a manner that maximises everyone's ability to code and test classes independently and that reasonably distributes the workload.

- See lecture notes for suggestions for unit, integration and system testing

- Ideas for demonstrating the assignment:

- It should be structured around use cases
- It should illustrate the underlying class structure that supports each area of functionality
- It should highlight challenging algorithms, code re-use, and inter-process communication mechanisms.”

In addition, it would be interesting to have an indication of any problems encountered, lessons learned, etc.

4. References

/1/ Braude, E., “Software Design, from programming to architecture”, Wiley 2004.

/2/ “Software requirements specification: Encounter video game”, CA314 continuous assessment project 2008-2009, file: ContAssProjReqts.

/1/ Stiller, E., LeBlanc, C., “Project-based Software Engineering”, Addison-Wesley 2002

ⁱ So that marks may be adjusted proportionately. At the extreme, a student who does not contribute at all to a team's efforts will get zero for continuous assessment.

ⁱⁱ A project box will be set up in Room L114.