

CS872: Software Engineering

Project Description

Dr. Samira Sadaoui

The goal of this course is the direct experience with a large-scale project (more than trivial information systems) in which the student plays the role of the customer and software engineer. In this course, students will gain experience in developing and implementing real-world applications. Students will also develop friendly graphical user interfaces for their software systems.

The project is not limited to the coding part. Indeed, students will analyze, design and implement a software following the phases of the software life cycle: feasibility study, requirements analysis and specification, high-level design, detailed design, coding and module testing, integration and system testing. The products produced in the software process are the following: feasibility study document, requirements specification document, design specification document, programs, system test plan. The evaluation of the project depends on the quality of these products.

1. Timeline

1. Each student should submit to the instructor the project title and the general requirements by September 19th, 2011.
2. The project report, programs and presentation slides should be recorded on a CD. On November 21st, you should submit to the instructor the CD and also a hard copy of the whole report and presentation slides.
3. Orally present and demonstrate the software. Presentations will start on November 21st. The ordering of presentations is determined by a random draw during the lectures. A presentation takes 15 minutes, demo 10 and questions 5.

2. Project Components

The final project report should include all the following products covering the whole software development process.

1. Problem definition (1 page).
2. Feasibility study document (2 pages). Clearly state if you are developing a new system, extending or improving an existing one.
3. Software requirements specification document. This document should include:

- (a) Consistent and unambiguous functional requirements. Give the complete use case diagram with all the use cases and actors. Define in detail five use cases (the most complex ones).
 - (b) Software qualities (correctness, efficiency, robustness and user friendliness, etc).
 - (c) System constraints (programming languages, operating system, hardware, databases, Web services, software tools, etc).
4. Design specification document, including:
- (a) Software architecture.
 - (b) Five sequence diagrams (for the five defined use cases).
 - (c) Complete class diagram with all the classes and their relationships.
 - (d) Two examples of object diagrams.
 - (e) Component diagram including the algorithm(s) behind your implementation.
 - (f) Deployment diagram (optional).
5. A document that comments in detail on the UML tool used for your project (such as Rational Rose, starUML) (1 page).
6. Technical documentation, including:
- (a) Programming languages.
 - (b) Reused algorithms and programs.
 - (c) Tools and environments.
 - (d) Database system. List all your tables.
 - (e) Text files. Indicate the formats of the text files used in your code.
7. User documentation including several screen shots of the GUI (2 pages).
8. Software test plan including:
- (a) Correctness testing with some data tests.
 - (b) Performance testing with some benchmarks.
 - (c) User interface testing with some screen shots.
 - (d) Robustness testing with some incorrect data.
9. Code by including all the necessary libraries. The code is constructed based on the specification and design documents.

Important Notes:

1. All the UML diagrams must be legible and produced using a UML tool.

2. **Rational Rose** is available on several platforms: Windows, Linux and Solaris. I encourage all the students to use the Solaris network in CL 136 lab. Rational Rose does not provide object diagrams. You can use Microsoft Visio that you can download from the CS website: www.cs.uregina.ca/MSDNAA.