

Deliverable: #4 - Detailed Design.
Due date: Due @ 0800 November 9, 2010.
Title: SE2: Software Design and Architecture.
Course ID: SE 464, CS 446, ECE 452, CS 646

Course www: http://bit.ly/uw_se2 [check frequently for updates]

Lectures: Tuesday & Thursday 1600 - 1720 MC 4063
Tutorials: Friday 1430 - 1520 MC 4063

Instructor: Dr. Reid Holmes; DC 3551. Office hours by appointment. rth.se2@gmail.com
TAs: Sarah Nadi; DC 3334. Office hours by appointment. snadi@cs.uwaterloo.ca
Wei Wang; DC 3334. Office hours by appointment. w65wang2cs.uwaterloo.ca

Description:

Document the detailed design of your proposed software system. Your target audience for this report is a junior programmer who would be responsible for implementing some portion of the project. Using this document, the programmer should be able to correctly implement any subset of the system.

Requirements:

1. Title page, including project name, team name, and each team members name and Quest IDs.
2. Detailed design.
3. Personal component.
4. Parts 1-4 must be compiled in a PDF document and submitted to rth.se2@gmail.com. Please name your file Project-Name_Design.pdf. Only one team member needs to send this document.
5. Part 5 must also be submitted in PDF to rth.se2@gmail.com. Please name your file Project-Name_Your-Name_Design.pdf. Be sure to include your name and project name on your personal component!

Detailed design:

You are to give, for each module, a clear specification for the module, such that a junior programmer can implement that module. The Detailed design will include a clear description of the behaviour of the module and its externally visible interfaces. It will include a description of algorithms and data structures to be used and will describe non-obvious implementation techniques. Rationale must be provided documenting why you selected your design. The applicability of your design compared to alternative designs could also be referenced in this discussion.

You should reference descriptions in your Architecture report of important patterns, classes abstractions, data structures or algorithms that are critical to the successful implementation of your system. Use diagrams as appropriate for this report. At a minimum, include a class diagram that shows all of the classes and public API for your system and how they interact. Clarify the physical location of where the classes will reside (e.g., in the browser, on a server), as well as any external API your system will use.

External Interfaces: Give details of information transmitted to/from the system, such as the Graphical User Interfaces, files, databases, messages or networks. This should include menu design (if appropriate) and a clear specification of any details by which modules in the system transmit information to/from the external interfaces. If appropriate, give language-level interfaces to specify these methods. You do not need to include a discussion of feasibility studies.

This detailed design should be a companion to your previously-submitted system architecture. Ideally, your report should repeat as little of the material as possible from the system architecture while still remaining a standalone document as much as possible.

Personal component:

This is an important aspect of the assignment (we will probably read it before the group component). First, provide a high-level overview of your system's detailed design. A person should be able to read this overview and have a firm understanding of the design of your system, its modules, interfaces, and most important algorithms. Synthesize the key points of the design rather than copying them from the group document. [This would typically be included first in the design document but is included here as an individual component.] The key here is *synthesis*, impart on the reader the main design decisions and rationale in a coherent fashion (1.5 pages or less). The personal component should be succinct and to the point (less than 2 pages in total). This component would be easiest to complete after the detailed design report is done.

Consider your personal component a two-page summary sent to your supervisor at a co-op job outlining the design of your system. What information do they need to understand the design? What rationale would convince them that your design is complete and is sound? These two pages are your opportunity to convince your supervisor that you know what you are doing and are a strong designer.

Assorted points:

Teams are allowed to discuss their designs with one another but must document this collaboration at the end of their report. References to any external resources (books or web sites) should also be included. The personal component **MUST** be completed individually; no collaboration is acceptable.

Assessment:

This deliverable is worth 10% of your final mark. 60% of the assignment grade will correspond to the design document; 40% of the assignment grade will correspond to the individual component. Be sure that each required component of the deliverable is complete and included in the final document and that the documents are submitted in the correct format, with the appropriate file name, by the due date.

For the group component, a design that could be implemented in practice with little additional information, embodies sound design principles, and clearly rationalizes design decisions will receive a high mark (in the A range); if non-major clarifications are required or minor design shortcomings exist a B will be assessed. Major design shortcomings, design ambiguities, or other deficiencies will receive C grades (or lower).

For the individual component, a similar rubric stands, but from the point of view of your supervisor.