

Deliverable: #4 - Project Architecture and Design
Due Date: Nov 17 @ 0800 via e-mail
Title: SE2: Software Design and Architecture
Course ID: CS 446, SE 464, ECE 452, CS 646

WWW: <http://www.cs.uwaterloo.ca/~rtholmes/teaching/2011fall/cs446/index.html>
Twitter: <https://twitter.com/cs446>

Lectures: Tuesday & Thursday 1600 - 1720 MC 1056
Tutorials: Friday 1430 - 1520 MC 4060

Instructor: Dr. Reid Holmes; DC 3351. Office hours by appointment. rth.se2@gmail.com
TAs: Kevin Shelley; DC 3334. Office hours by appointment. karshell@cs.uwaterloo.ca
Wei Wang; DC 3334. Office hours by appointment. w65wang2cs.uwaterloo.ca

Description:

Document your project's architecture and design. Your target audience for this documentation is a junior programmer who would be responsible for implementing some portion of the project. This is the first thing you would give a new employee to get them up to speed on the structure of your system.

This deliverable will consist of three independent parts: a high-level architectural description, a low-level design description, and an individual component. As the class projects are modest in scope, there will naturally be some overlap between the architecture of the system and its design but this should be kept to a minimum.

Requirements:

1. Title page, including project name, team name, and each team members name and Quest IDs.
2. System architecture.
3. System design.
4. Personal component.
5. Parts 1-3 must be compiled in a PDF document. Part 4 must be in a separate PDF document.
6. Only one team member needs email the team component to rth.se2@gmail.com by 0800 on Nov 17.
File naming scheme: `cs446-d4_<project-name>.pdf`
* (use - instead of space in file names)
7. Each team member needs email their personal document to rth.se2@gmail.com by 0800 on Nov 17.
File naming scheme: `cs446-d4_<project-name>_<your-name>.pdf`
* (use - instead of space in file names)

System architecture:

Describe the overall architecture of your project with descriptions of each major component, the connectors, and the topology that binds them. The components could be packages, subsystems, modules, frameworks, or external interfaces as is appropriate for your project. Ensure that all of the external components/services your system will interact are accounted; these must be captured in your architectural description. Create a diagram that accurately reflects your system that is as unambiguous as possible. As Deliverable 2 focused on the architecture and design goals of your system, this deliverable will focus more on the components and their interrelationships.

Your architecture should be easy to understand with simple interfaces and modest interaction among components. Clarify the architectural style(s) of your system and justify why you chose the style(s) you did. Include a discussion of at least two other styles you considered and justify why you decided not to use them in the end.

While finding the right balance between architecture and design may be difficult, be careful not to make your architecture so high-level that it misses important information (e.g., if you have a client/server system, elaborate on the architectures of both

the client and server components). A good rule of thumb is that discussing individual methods and fields is too low level; inevitably though, you may end up talking about class-level details since these projects modest in scope.

Your architectural document should contain some clarifying diagrams that clearly illustrate the structure of your system. The most important of these is the component diagram; be sure it is clear what the requires and provides interfaces are for each component. You should also include a simple state diagram and sequence diagrams for each of the use cases you specified in your original proposal. You may use software tools to draw the architecture or draw it by hand and embed a photo/scan of the document (NOTE: ensure your diagrams are easy to read when printed or viewed on the screen).

Future considerations:

An analysis of how your architecture would accommodate changing requirements is required. Think critically about how you could envision your system being altered and discuss how your architecture would support or inhibit evolving to meet those changed requirements. Identify two ways you think your system may need to evolve in the future and describe how your project's architecture would support these changes.

Detailed design:

You are to provide, for each component, a clear specification such that a junior programmer can implement and integrate any given component. Your design should include a clear description of the behaviour of the module and its externally visible interfaces. It should 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 should 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.

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 20% 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.