

**Deliverable:** #2 - Software Architecture.  
**Due date:** Due @ 0800 October 12, 2010.  
**Title:** SE2: Software Design and Architecture.  
**Course ID:** SE 464, CS 446, ECE 452, CS 646

**Course www:** [http://bit.ly/uw\\_se2](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](mailto:rth.se2@gmail.com)  
**TAs:** Sarah Nadi; DC 3334. Office hours by appointment. [snadi@cs.uwaterloo.ca](mailto:snadi@cs.uwaterloo.ca)  
Wei Wang; DC 3334. Office hours by appointment. [w65wang2cs.uwaterloo.ca](mailto:w65wang2cs.uwaterloo.ca)

### Description:

Document the architecture of your proposed software system. Your target audience for reports and documents is a manager or developer who is somewhat but not intimately familiar with your project. This document would be the first thing you would give to a new employee to understand how your system is structured.

### Requirements:

1. Title page, including project name, team name, and each team members name and Quest IDs.
2. System architecture.
3. Methodology.
4. Future considerations.
5. Personal component.
6. Parts 1-4 must be compiled in a PDF document and submitted to [rth.se2@gmail.com](mailto:rth.se2@gmail.com). Please name your file Team-Name\_Project-Name\_Architecture.pdf. Only one team member needs to send this document.
7. Part 5 must also be submitted in PDF to [rth.se2@gmail.com](mailto:rth.se2@gmail.com). Please name your file Team-Name\_Project-Name\_Your-Name\_Architecture.pdf.

### System architecture:

Give the overall structure of the system that you have proposed 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. Concentrate on the goals, requirements, evolvability, interfaces, and testability of the entities in your architecture. Ensure that all of the external components/services your system will interact are accounted; these must be captured in your architectural description. The diagram should accurately reflect the system you want to build, be internally and externally consistent, and be as unambiguous as possible.

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 different, 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 classes, methods, and fields is too low level.

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).

Include a description of how each of the use cases you provided with your proposal activate or use the various parts of your system's architecture. Also describe how your architecture supports the non-functional qualities you outlined in your proposal.

### **Methodology:**

Provide a plan that describes how the system will be designed and implemented (e.g., what modules / subsystems and in what order, who will do them and how they should be tested), with estimates for how long each module or subsystem will take. Provide a schedule for the remainder of the project with respect to these modules / subsystems, including the prototype demonstration, design deadline, final implementation, and final demonstrations. A Gantt chart is not required but may be helpful. Investigate and report on the feasibility of key or risky parts of your system. You may wish to create small prototypes investigating these risky interactions to ensure they are possible. It is expected that you will have identified and investigated (in code) some of these main risky interactions to ensure that they are possible or to devise a risk mitigation strategy if they are not.

### **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 (e.g., supporting races instead of games, supporting multiple locations and people, supporting different departments, supporting different campuses, supporting room bookings, supporting other IOUs, supporting different kinds of events, supporting new devices or form factors).

### **Personal component:**

This is a very important component of the assignment (we will probably read it before the group component). First, provide a high-level overview of your system's architecture. A person should be able to read this overview and have a firm understanding of the architecture of your system, its components, connectors, and topology. Synthesize the key points of the architecture rather than copying them from the group document. [This would typically be included first in the architecture document but is included here as an individual component.] The key here is *synthesis*, impart on the reader the main architectural decisions and rationale in a coherent fashion (1.5 pages or less). Second, outline the portions of the architecture you will be most involved in designing, building, and testing, as well as any other roles you expect to take in your project (.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 architectural report is done.

### **Assorted points:**

Teams are allowed to discuss their architectures 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 architecture 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.