Deliverable: #4 - Project Design
Due: March 18 (written); March 19 (oral)
Title: SE2: Software Design and Architecture.
Course ID: CS 446, SE 464, ECE 452, CS 646

WWW: http://www.cs.uwaterloo.ca/~rtholmes/teaching/2013winter/cs446/index.html
Twitter: https://twitter.com/cs446

Lectures: Monday, Wednesday, & Friday: 0830 - 0920, MC 4041

Instructor: Dr. Reid Holmes; DC 3351. Office hours by appointment. rth.se2@gmail.com
TAs: Laura Inozemtseva; DC 3334. Office hours by appointment. lminozem@uwaterloo.ca
Wei Wang; DC 3334. Office hours by appointment. w65wang@cs.uwaterloo.ca

Description:

Document your project's 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 low-level structure of your system.

This deliverable will consist of two independent parts: a low-level design description and an oral presentation where your team will meet with us to discuss the design of your system. The written component should clearly and succinctly capture and justify the design of your system. This description can be at most 5 pages (2 pages of primarily text and 3 pages of primarily diagrams).

Team deliverable:

1. Metadata: Project title, team member names, team member Quest IDs.
2. System design (4 page maximum).
3. Only one team member needs email this document to `rth.se2@gmail.com`. PDF only.
   File naming scheme: `cs446-d4_<project-name>.pdf`
   * (use – instead of space in file names)

Detailed design:

Create a document that describes the design of your system, and its rationalization, such that a junior programmer could implement some subset of the system and integrate it appropriately. Your design should include a clear description of the structure of the components and its externally visible interfaces. 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, and data structures / 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 along with a sequence diagram that captures how your system behaves for each of the scenarios from the initial proposal. Clarify the physical location of where the classes will reside (e.g., on the client, on a server), as well as any external API your system will use.

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

For the oral component, be prepared to defend your system's design. We will also likely ask about how your design could adapt to specific given evolutionary constraints (e.g., 'you must now support XXX, how would you do that?').

Assessment:

This deliverable is worth 20% of your final grade. The written component is worth 60% of the deliverable mark; the oral component constitutes the remaining 40% (split 50/50 between a group score and an individual score).