

CSSE 374 – Software Architecture and Design I

Project Milestone 4

In Milestone 4 your team will update your analysis and design documents from Milestone 3. To do this, you'll use the feedback from your instructor and the design principles and patterns that we've been discussing and practicing. Additionally, you will be applying the nine GRASP principles to your design.

You will also build on the code infrastructure that you created for Milestone 3. In particular, you'll write code to implement the basic functionality of your project's domain layer. For the most part, we'll defer user interface implementation to Milestone 5.

Objective

We have three main objectives for Milestone 4. First, we want to see that you can apply the GRASP principles to your projects. Second, we want to see that you can make the transition from designs to code. And third, we want to see that you can complete Iteration 2 of your systems, which entails most of the domain layer functionality.

Due Date

11:59 p.m., Friday, Week 7, (January 29th, 2010)

Tasks

Using the techniques discussed in Larman's *Applying UML and Patterns*, extend your work from Milestone 3 by completing the tasks listed below. *Please note that any graphical representations must have accompanying text briefly stating the rationale and assumptions for the organization you have chosen.*

1. Update your Domain Model, System Sequence Diagrams, Operation Contracts, Interaction Diagrams, Logical Architecture, and Design Class Diagrams from Milestone 3 based on the feedback from your instructor.
2. Discuss how you applied the nine GRASP principles (Low Coupling, High Cohesion, Information Expert, Creator, Controller, Polymorphism, Indirection, Pure Fabrication, and Protected Variations) to arrive at your design. Make sure you reference your interaction diagrams. To succeed at this task, you will need to compare your given design to alternatives. After all, the GRASP principles are all about making decisions.

You may find it useful to write separate rationales for each interaction diagram. On the other hand, we aren't looking for a lot of duplication, so (for example) it probably makes sense to discuss Controller (use case vs. façade) in one place.

3. Continue developing the code for your system into Iteration 2. Build upon your work from Milestone 3 by implementing classes for your domain layer. Follow the guidelines from Ch. 20 to turn your designs into code. Note that you'll need to finish your design diagrams well ahead of the code deadline!

We are *not* looking for user interface code in this milestone.

“But wait,” you say. “Without a UI, how will we know that our domain layer code is correct?” You have two options for testing your domain layer code.

- You can use test-driven development to create unit tests for your domain layer code as you develop it, or
- you can implement a rudimentary UI layer to exercise your domain layer code.

We strongly encourage you to choose the first option if it is reasonable for your project. However, we recognize the each project is unique. Work with your instructor to clarify what code you should develop for this milestone.

Demonstrate your software for this iteration at your first project meeting on or after Friday of 7th week.

Submitting Your Work

Please submit your Milestone 4 document with your Domain Model, System Sequence Diagrams, Operation Contracts, Interaction Diagrams, Logical Architecture, Design Class Diagrams, and design rationale as a single PDF document to your team SVN repository. Make sure you have neatly formatted your document, so we can print it.

Name your document *RefinedDesign.pdf*.