

# CSSE 374 – Software Architecture and Design I

## Project Milestone 3

The first two milestones were designed to help transition from CSSE 371 to CSSE 374 by setting up the course infrastructure (SVN, etc.), and establishing a basis for moving into design (via a Domain Model). Note up to this point, we did not involve tasks for producing any source code.

Milestone 3 introduces the first Iteration for your Junior Project. This involves detailing the problem space with System Sequence Diagrams (SSD) and developing Operation Contracts (OC) for relevant operations. This milestone lays the foundation for producing a Software Architecture for your project. In this iteration, you will also develop a Logical Architecture and Package Diagram(s) showing the packages, layers, and partitions appropriate to your project. These documents will help organize your software implementation.

## Objective

There are four objectives in Milestone 3:

- 1) Complete the junior project analysis models by developing SSDs and OCs.
- 2) Using the analysis models, organize classes into packages and organize packages into layers and partitions, producing a Logical Architecture and Package Diagram(s).
- 3) Develop the design models for Iteration 1 of the project including both static Design Class Diagrams and dynamic (behavioral) Interaction Diagrams.
- 4) Develop an initial working version of the system that has the major infrastructure in place.

## Due Date

11:59 p.m., Friday, Week 4, (January 8<sup>th</sup>, 2010)

## Tasks

Using the techniques discussed in Larman's *Applying UML and Patterns*, build upon your Domain Model from Milestone 2 for your team project 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. Develop SSDs describing the behaviors and events between your junior project system and key actors in the application domain (Chapter 10).

2. Develop OCs detailing key operations more formally in your analysis model (Chapter 11).
3. Using the analysis model elements (DM, SSDs, and OCs), formulate the allocation of classes to packages based on guidelines from the text. These packages should then be allocated to layers and partitions in your logical architecture and depicted in one or more package diagrams (Chapters 12 & 13).
4. Produce a set of Interaction Diagrams (Sequence and/or Communications Diagrams as appropriate) that model the key behaviors for Iteration 1 functionality (Chapter 15).
5. Produce a set of Design Class Diagrams that show the relevant classes and their associations, dependencies, and composition relationships (Chapter 16).
6. Develop an initial working version of your system. You will need to identify the basic elements of your domain layer to implement first. This core implementation should provide the basic infrastructure on which you will build functionality in future iterations.

For example, you may need to have user interface and database technologies in place. Or you may be using some open source components in your design that you will need to analyze and begin programming against.

Demonstrate your software for this iteration at your first project meeting on or after Friday of 4<sup>th</sup> week. You may use your team SVN repository for source code control. You are also free to use some other version control system (e.g., git on github or Mercurial on code.google.com). If you choose another system, you must make sure your instructor is able to access your code.

### **Submitting Your Work**

Please submit your Milestone 3 document with your SSDs, OCs, Logical Architecture, Interaction Diagrams, and Design Class Diagrams as a single document to your team SVN repository. You can commit either a PDF file or a Word document. Name your document LogicalArchitecture, with the appropriate extension (pdf, doc, or docx). If you are in Dr. Clifton's class, please submit this as a PDF file with the pdf extension.