

# CSE293 CSE Design Laboratory – Project 3

## Revised Specification and Design

**DUE: Given on Part-by-Part Basis**

Each team is to submit **ONE** assignment with the team name and the names of all team members. Clearly identify which team member did what portions of the submitted assignment.

### **Part I - Due at the fourth class.**

The first part of this assignment is to modify and expand your specification in light of the discussions of Part I and any comments that were provided on your handed in specification. Coincident with this part is Part III given below that begins to add the detail for your project solution. You can add up to three more pages of text (10 total) to your specification.

### **Part II - Due at the fifth class**

The next step in completing the specifications is to more clearly define the control and data organization of the project. You must use a chief-programmer team approach, where a different team member serves in the role of chief-programmer for Tasks A through E given below. For each part, a second team member should provide backup and input. All team members are expected to be familiar with all four tasks and be able to converse about each task during the weekly meeting sessions. All teams must do Tasks A, B, C, D, and F. Task E is to be done only if deemed relevant by a consensus decision by the team. Tasks A through E are intended to evenly distribute the work required by the team so that each team member can demonstrate an individual contribution as part of this assignment. For tasks A to E, each team member must do two diagrams, and each diagram must be accompanied by at least one-half page of written documentation (12 pt, 1 inch margins, single spaced). Task F should be a joint team effort by all members. Note that each design component should be accompanied by detailed written documentation. Examples from prior semesters are on the course web page.

#### **Task A**

Construct Entity-Relationship (ER) diagrams for the data showing the conceptual structure of the data and its interrelations (see Ghezzi, section 5.6.1, or another software engineering textbook). You must use the ER specification capabilities of Together in constructing your diagram. Utilize the documentation generation capabilities of Together to generate HTML documentation for your ER diagrams. Make sure that you describe the attributes of each entity including types and acceptable ranges. Create a twofpage explanation of the selection of your entities and relationships. Include a discussion of any design alternatives and their evaluation. Integrate this documentation with the Together html documentation.

## Task B

For Task B, you have the two following alternatives:

- Construct Data-Flow Diagrams (DFD) for the system (see Section 5.5.1, Ghezzi or another software engineering textbook). Your DFD diagrams should follow the graphical conventions in the book. DFDs may be both high-level and detailed, depending on your focus and approach, i.e., a hierarchy of DFDs. Create a one page explanation of the functions and data stores used for each DFD, and include a discussion of any design alternatives (different function and data store alternatives) and their evaluation.
- Construct UML Sequence Diagrams (SDs) to show the sequential flow between your various system objects and components. Unlike DFDs, SDs concentrate more on the flow that occurs over time, whereas DFDs are more discrete in nature. See Chapter 18 of the UML textbook for a discussion and explanation of SDs.

Regardless of your choice, you must fully document your output, and integrate your documentation with the documentation produced by Together.

## Task C

Construct Finite State Machines (FSM) to show the general control structure of the system (see Section 5.5.2, Ghezzi, or another software engineering textbook). Clearly identify each state and the transitions between states. You may want to create a hierarchy of FSMs to help manage the complexity. Each FSM should include an explanation of the states and the arcs between the states and include a discussion of any design alternatives and their evaluation. FSMs are supported in Together using statechart diagrams (state machines in UML, see Chapter 21 of UML textbook). You must use the statechart diagram capabilities of Together in constructing your FMSs. Utilize the documentation generation capabilities of Together to generate HTML documentation for your FSMs.

## Task D

Construct a UML Class/Object Diagrammatic representation/design for the system. You must include both public, private, and protected methods, and private and protected attributes. Organize your classes into packages as appropriate, and use the various UML references and relationships to indicate which classes and/or packages collaborate with one another. You must include the user and database interfaces in your design by describing the methods available for use and what they do. You should divide the packages and classes between group members, with each group member responsible for a set of packages and classes and a back-up member that has secondary responsibility for those packages and classes. This allows group members to be graded on an individual, as well as a team basis.

## Task E

For Task E, you have the two following alternatives:

- Construct Petri Nets (PNs) to show timing related actions that compete for resources of the system (see Section 5.5.3, Ghezzi or another software engineering textbook). Clearly identify the places and transitions, with arrows indicating the flow. Each PN should include an explanation of its structure.

- Construct UML Activity Diagrams (ADs) to show structural components and flow between the various states. Like PNs, ADs support both forms and joins to indicate actions that can occur in parallel.

Regardless of your choice, you must fully document your output, and integrate your documentation with the documentation produced by Together.

### **Task F**

Create a three page document discussing and explaining the relationship between Tasks A through E. You should discuss how ERs, DFDs or SDs, FSM, class diagram, and PNs or ADs, relate to one another and you should discuss the combined contribution of these four techniques in terms of correctness and robustness.

### **Part III - Due at the sixth class**

Based on comments provided with Part II, you are to revise your design (both diagrams and written portion) for additional comments.