

## **Systems Development: Requirements Analysis Assignment**

### **Background**

This is the third of three related assignments for the requirements phase of developing a student registration database. We will engage in a practical exercise for publishing a complete system specification.

### **Due Date**

Each group will turn in a written report of the completed assignment due before the start of class on April 16, 2007.

### **General Requirements**

During the requirements specification sub-phase, each team translated subjective and objective murmurs, needs, and wants from needs assessment into requirements using a collection model. Each requirement has specific characteristics and a taxonomy.

We did not try to resolve conflicts among the various requirements. We also did not examine how these requirements might group together into subsystems. That is the main work for this phase.

Develop a system specification to support design. Follow the activities mentioned in the specific requirements below. Include the list of modified requirements along with the other items requested in the final report.

### **Specific Requirements**

Please make sure your group completes **all** of the assigned tasks for the entire assignment.

### Schedule Work

Plan the group's tasks for the entire assignment. Determine a reasonable work schedule for completing all relevant tasks. Note the sample report on the web site includes an example schedule.

### Resolve Conflicts

It is possible (but highly unlikely) that the requirements specification activity produced a list of items with no internal conflicts. Analyze the collected requirements and look for items in opposition. Resolve the conflict with customer input.

If any requirement is changed to resolve a conflict, update the modification date in the collection model. Also write a sentence or two in the system specification documenting how the conflict was corrected.

### Perform Subsystem Decomposition

Examine the requirements list to look for natural groupings. Collect similar requirements together in subsystems. Determine the interface between subsystems, namely the sharing of inputs and outputs. Identify activity responsibilities for all subsystems.

### Input-Activity-Output Diagrams

For each subsystem, draw an input-activity-output diagram. This diagram shows all inputs to the subsystem, all data transformation processes as activities, and outputs from the subsystem.

The diagram for one subsystem should clearly reference other subsystems. For example if Subsystem B takes input from Subsystem A and passes output to Subsystem C, this should be identified explicitly. Describe in natural language all inputs, activities and outputs as well. Note the sample report on the web site contains an example diagram and text descriptions.

### Use Cases

Develop one use case for every transaction suggested by the requirements. Transactions may be initiated by a person, the system, or their mutual cooperation. Follow UML guidelines for diagramming and documenting a use case. Your group may use the simplified notation given during the lecture.

## **Final Report**

Submit a report including the work schedule, all of the items mentioned above, and the modified requirement list. Write a short (one page) executive summary on the major findings from the group during this phase.