



## 605.601— Foundations of Software Engineering: Fall 2020

### Assignment 03: Team Project – From Analysis to Design and Architecture

Due Date: **Week 14: November 30, 2020 Midnight**

#### Reviewing/revising your analysis:

Please take the next steps from the analysis. Draw one or two activity diagrams.

1. Identify **ONE** Use case from EACH package you have been working on during your last assignment (Assignment # 2).
2. Revisit and study the flow within the Use case, including alternative flows and exceptions.
3. Draw an Activity diagram for the Use case based on the flow of the Use case appearing in its documentation (if you have one).
4. Ensure each Activity diagram has carefully created partitions including the System partition (objects and components).
5. Work as a team to identify Classes from the descriptions of the Use cases you created in previous modules. This will require you to READ and ANALYZE both the Use cases and the problem statement for your system.
6. Create and enter all your Entity classes in your modeling tool. *[Note: It is highly likely that you have not identified all the classes at this stage. When you conduct abstraction and create additional hierarchies of your classes, you will discover and invent many more entity and other classes.]*
7. Each entity class during analysis is expected to contain AT LEAST one or two attributes and a similar number of operations.

#### Moving on to Design:

1. Identify and document additional design-level classes (such as database classes and classes from your possible language of implementation) that extend your original class design. If you like, please feel free to keep it at a Component-level.
2. **(Optional)** Fully DEFINE all the Entity classes in design by identifying and defining their Attributes and Operations as discussed in object-orientation module. Enter and update the details in your modeling CASE tool.
3. Ensure that ALL attributes are defined.
4. Ensure ALL Operations have Signatures (containing parameters and return values) and Visibility.

5. Generate pseudo-code for your above Class diagram (if possible, in the modeling tool you are using. Use a language of your choice if necessary, to enable code generation).
6. Consider some of the advanced design features discussed in the class (polymorphism, multiple inheritance) and enhance your class diagram with those features.
7. Consider a design pattern like Observer, Singleton or Façade. Apply that design pattern in any one of your class diagrams.

### **Preparing Software / System Architecture:**

1. Create a System Architecture diagram for your project.
2. Create ONE UI-Flow diagram corresponding to an Actor using the interfaces identified.
3. Identify and specify ONE device for your case study (device interfaces may include printer, card readers for security ID cards, barcode readers for inventory, etc.)
4. Identify and draw a detailed component diagram for your case study. [At this stage you are required to produce only ONE diagram that shows the major components of your solution together; hence the team will have to work together in creating this diagram.]
5. For this exercise, focus only on the components, and not on their relationships.
6. Map a cohesive set of Classes to each of the components. [This mapping of classes to components (also called "realization") will be achieved in two to three attempts (i.e., it won't be achieved in the first instance). Note that in practice, the language of implementation will provide some ready-made components, which will become part of your solution design.]

**Deliverable 1:** A brief description and presentation of your activity diagram(s) that you and your team is preparing. This may include your reasoning behind choosing the 'actor' and 'use cases', any dependencies or potential connections – and why you believe these choices are important. You don't need to be perfect. Please note that I am looking for a way to assess how well you understand the concept.

**Deliverable 2:** A set of design diagrams with a basic use of pattern, design features, and the generation of pseudo-code. Again, for this assignment, you may save your diagrams and the package in PDF format if you like.

**Deliverable 3:** A set of architecture diagram(s) with basic components and user interfaces. All I am looking for is a conceptual architecture model with fundamental layers and tiers as we discussed in the class (Please see Module 06A and 06B).

Please contact me with any questions or suggestions you may have.