Assignment 1 – CS 618 Software Design

Due at start of class on February 9, 2016

You are asked to build use cases as part of an overall domain model. We suggest that you model the secondary education domain, but you may select a different domain. The domain has students, professors, staff members…..and requires work such as: managing employees (hire them, pay them), buying things (accounts payable), managing students (allowing students to register for classes), etc. You may narrow the domain as you see fit (Is it for registration?  Class scheduling?)

We ask that you use Alistair Cockburn's approach to writing use cases.  The guidelines from his "Writing Effective Use Cases" book are summarized here: <http://faculty.washington.edu/jtenenbg/courses/360/f02/project/usecaseguidelines.html>

More info here too:

<http://www.slideshare.net/harshjegadeesan/lecture-2-writing-effective-use-cases-presentation>

Cockburn slides on our course webpage

Please do the following individually:

1. Name the system scope.
2. Brainstorm and list the primary actors.
*Find every human and non-human primary actor, over the life of the system.*
3. Brainstorm and exhaustively list user goals for the system.

Then, in groups (of three), you should get together and review the individual lists.  Prepare a short write up that describes what you learned in the review that would have been missed in the individual lists (e.g., which actors and/or use cases would have been missed without the peer review?)

For each of the user goals, provide the main success scenario:

1. For each user goal - one at a time - select a use case to expand.
*Consider writing a narrative to learn the material.*
2. Write the main success scenario (MSS).
*Use 3 to 9 steps to meet all interests and guarantees.*

*Determine which of the use cases are central to the analysis (high priority and/or complex), for these use cases:*

1. Brainstorm and exhaustively list the extension conditions.
2. *Include all that the system can detect and must handle.*
3. Write the extension-handling steps.
*Each will end back in the MSS, at a separate success exit or in failure.*
4. Extract complex flows to sub use cases; merge trivial sub use cases.
*Extracting a sub use case is easy, but it adds cost to the project.*
5. Readjust the set: add, subtract, merge, as needed.

Deliverables:

1) A statement of system scope

2) A list of the primary actors

3) A list of user goals (candidate use cases)

4) A couple of paragraphs about the peer review

5) Main Success Scenarios for each use case (there will probably be 10-12 of these)

6) "Fully Dressed" Use cases for high priority use cases (there will probably be 4-5 of these)