Deliverable: #2/#4 - Architectural Styles & Design Patterns Presentations
Due: Noon on the day of your presentation.
Title: SE2: Software Design and Architecture (CS 446, ECE 452, CS 646)

Description:

We will cover 9 architectural styles and 9 design patterns in this course. Each team will be given a 20 minute slot to help the class learn about a style or pattern, what its strengths and weaknesses are, and to see how can be used in practice. You should also walk the class through at least one comprehensive example demonstrating how the style or pattern could be used and specifically how it is beneficial to the overall design of the system. All styles and patterns work to address coupling in a specific way; describe how is coupling reduced and what kinds of future changes are enabled by your style/pattern. Each team will choose one in class through a random draw on Jan 17 (or I will randomly draw for any teams that are missing from in class).

The graduate students will have a choice between MVC/MVP design patterns or Cloud/REST architecture styles. The presentation time for this would be 30 mins.

You can use any material you can find from books or online in creating your presentation, but you cannot use the projector. Memorable presentations are encouraged as they will better help everyone remember the details of the architectural style you present.

Presentation Details:

- 20 minute maximum (30 min for graduate student teams).
- Two primary components:
    - Static description of the style / pattern
    - Dynamic description of how the style / pattern is useful over time
- Comprehensive example / tutorial that demonstrates the style / pattern
- No slides; be creative. Make it memorable.

If you are presenting an architectural style; does the style:

- Have its own vocabulary for its components and connectors? (define)
- Impose specific topological constraints? (diagram)
- Most applicable to specific kinds of problems?
- Engender specific kinds of change resilience?
- Have any specific negative behaviours?
- Support/inhibit specific NFPs?

If you are presenting a design pattern; does the pattern:

- Have a purpose / motivation?
- Intended use case?
- Have its own vocabulary? (define)
- Specific structure or runtime behaviour?
- Have known consequences (both positive and negative)?
- Improve / degrade NFPs discussed during architectural discussions?

Deliverables:

While the primary deliverable is the presentation itself, by noon on the day of your presentation you must email me and Has a description of the example you will present to the class. The focus point of this description should be a visual representation of the components / classes associated with your style / pattern. This should be accompanied by a short description of the positive properties your style / pattern provides to the example.