Web Application Design Notes

Jeff Offutt
http://www.cs.gmu.edu/~offutt/

SWE 642
Software Engineering for the World Wide Web

N-Tier Architecture

Client-server ... 3-tier ... N-tier ...
Design Goals

• A major design goal of the N-tier architecture is separation of concerns:
  1. Presentation
  2. Logic
  3. Data

• Also to support our seven criteria:
  1. Maintainability
  2. Security
  3. Scalability
  4. ...

Separation of Concerns

• Presentation → HTML & JSP
• Data content → Servlets & beans
• Data representation → Data structures, Beans & java classes
• Data storage → Data base & files, Oracle, sql
Separation of Concerns (2)

- `doGet()` and `doPost()` should be simple and short
  - Shift processing to other methods and classes
- Put complicated logic in non-servlet classes
- Put almost no logic in JSPs
  - JSPs should present data they get from other classes

- Use servlets to process user input
- Use JSPs to present data that is on server

Design Specification—Good Habits

- Software Requirements Baseline
- Information Architecture Specification
  - Site map, Web Page Flows, Compositions, Labeling, Data Element Mappings
- Web Application Design
  - High-Level Software Design
  - Software Architecture and System Architecture Diagram
  - Class Diagrams
  - Sequence Diagrams
  - Class Specifications
Model-View-Controller (MVC)

- The MVC architecture is a common abstraction in web application design
  - Divides the responsibilities of objects
  - Decreases coupling between objects and layers (supports easier maintenance)
  - Helps divide the work – supports development expertise areas
- One of the first web design patterns
- Probably over-used
- If I only have one tool, a hammer, all problems look like nails …

Graphic from Designing Enterprise Applications with the Java 2 Platform, Enterprise Edition, Nicholas Kassem et al., October 2000
Web Application Design Complexity

Common Design Mistakes

- No design specifications and no comments in code
- Overly limiting collaboration amongst the development team — only 1-2 people understanding and owning the design
- Coding for future requirements
  - Don’t code ahead—requirements will always be different
  - Instead, use a flexible design
- Choosing a design framework that is too complicated
  - Then making it worse by only using part of the framework
- Using the “be lazy” parts of agile processes without using the “do it better” parts
**Best Practices**

- Establish a Software Requirements Baseline
- Create design specifications
- Use Java Doc
- Train all of the development team on the design patterns and design constructs selected for the application, especially the connection points between tiers
  - 30 minute rule—every person can explain design in 30 minutes
- Design/develop adaptable systems, using object-oriented principles
- Use meaningful package, class, method, and variable names

**Web Application Design Summary**

- SWE 642 teaches the basics, but only scratches the surface of designing web applications
- SWE 642 gives a solid grounding, but your knowledge is far from complete
- Dozens of design frameworks are available to help build better web applications
  - Be sure to learn the concepts, not just the syntax!
- Practitioners and educators are learning more every day
- The web changes every aspect of software engineering