Multi-tier Architecture and Debugging

FCDB 9.1

Dr. Chris Mayfield

Department of Computer Science
James Madison University

Mar 19, 2013
Three-tier architecture

**Presentation tier**
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

**Logic tier**
This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

**Data tier**
Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

Source: [http://en.wikipedia.org/wiki/Multitier_architecture](http://en.wikipedia.org/wiki/Multitier_architecture)
### Languages and software

**Diagram:**
- SQL (PostgreSQL, MySQL, etc.) 
- Java (Tomcat, JBoss, Glassfish, etc.) 
- HTML (Apache, IIS, GWS, etc.) 
- JavaScript (Chrome, Firefox, etc.) 

### Design principles
- Separation of concerns (data, logic, gui)
- Development: all tiers run on same machine
- Deployment: multiple machines for each tier
HTTP is stateless

When you browse the web:

1. Client asks for URL
2. Server returns HTML
3. Otherwise disconnected

How do we make an application?

- URL parameters (via GET/POST)
- Server-side sessions (cookies)
<form action="query.jsp" method="get">
  <input type="text" name="title">
  <input type="submit" name="ins" value="Insert" />
  <br>
  <a href="query.jsp?del=%7BGo+Dukes!%7D">Delete</a>
</form>

Action
▶ Where to submit the form

Method
▶ GET = parameters visible in URL
▶ POST = parameters in HTTP request
Java Servlets

JSP = Template for Java Servlet
  ▶ See http://en.wikipedia.org/wiki/Java_Servlet
  ▶ Goal: generate dynamic HTML response

Don’t forget: Ctrl-space
Impedence mismatch

Relational

```sql
CREATE TABLE movie (  
    movie_id integer PRIMARY KEY,  
    title text NOT NULL,  
    kind_id integer, -- foreign key  
    year integer,  
    ...  
);
```

Object-oriented

```java
public class Movie {  
    private int movie_id;  
    private String title;  
    private Kind kind; // Java enum  
    private int year;  
    ...  
}
```
Interactive Demo: RealityCheck

svn+ssh://student.cs.jmu.edu/cs474/absent
Today’s concepts

index.jsp
- JSP directives: import packages, error pages
- Including other JSP pages (organize/reuse code)
- Writing comments, CSS styles, javascript, etc.

divsch.js
- Use SQL to generate code in other languages
- Example functions that manipulate HTML DOM

Query.java
- Passing request and out objects to methods
- Using StringBuilder to generate dynamic SQL

output.jsp
- Using session to maintain application state
Debugging techniques

Server-side debugging (NetBeans)
- Set breakpoints in Java/JSP code
- Step through code, inspect variables
- View request headers/etc from Tomcat

Client-side debugging (Chrome)
- Inspect HTML elements (Ctrl-Shift-c)
- Set breakpoints in JavaScript code
- Explore DOM interactively in Console