Objectives

• Review Ajax
• What We’ve Done
• In Brief: What else is out there

Review: Ajax

• What technologies does Ajax use?
• How does Ajax work?
• Why is Ajax useful?

• Content does not have to be in XML
  ➢ Can return plain text

Security

• Better not to have JSPs visible to user
• If go to servlet, hides the technology used from the user
  ➢ No explicit .jsp extension
  ➢ Hacker can’t focus attack on known technology

1st Day: What This Course Is About

• Web applications
  ➢ Distributed computing
  ➢ Web application technologies (server and client)
  ➢ How to develop high-quality Web applications
• Software engineering
  ➢ Large development project
  ➢ Software, collaboration tools
  ➢ Emulate real-world experience with actual client
• Life-skills
  ➢ Reading, writing, discussion, presentation

What We’ve Done

• Client-Side
  ➢ XHTML
  ➢ CSS
  ➢ JavaScript
  ➢ Ajax
• Server-Side:

Software Engineering Skills

• Writing requirements documents
  ➢ Anticipate needs, potential problems
• Design skills
  ➢ Interface first
  ➢ Rapid prototyping, static HTML mockup
  ➢ Iteration - improves final product
  ➢ Find problems early
• Collaborating with team members
  ➢ Version control software, wiki
• Handling deadlines
  ➢ Prioritizing functionality, modifying requirements
Tools
- Eclipse Web Tools Platform
- Firefox Plugins
  - WebDeveloper, Firebug
- Version Control
  - Subversion, Subclipse
- Wiki, Blog

Web 2.0 Technology
- Different user experience than before
  - Emphasis on user collaboration, creativity, information sharing
- Ajax: enables some of new experience
- Wikis, Blogs, Social networking sites

Using Your Knowledge for Good?
- “So, You Hacked Our Site!”

Frameworks
- Know the Java-based Web fundamentals
  - Servlets - foundation; JSPs, JSTL
- Frameworks
  - JavaServer Faces (JSF)
    - Simplify UI development using components
  - Apache Struts
  - Spring
    - Newer, lighter-weight MVC Framework
    - Works with languages other than Java too

Alternative Programming Languages
- ASP: Active Server Pages
  - Microsoft, VB Script
- CGI: Common Gateway Interface
  - Typically implemented in C or Perl
- Django: MVC Python-based Web Framework
  - Altered form in Google App Engine
- PHP: Hypertext Preprocessor
  - Easy, runs on Apache, but security holes
  - Our Wiki uses PHP
- Ruby on Rails: Ruby-based MVC Web Framework

Web Services
- Machine to machine communication
  - Rather than human to machine
- WSDL: Web Services Description Language
  - Well-defined interface
- SOAP: XML messages
  - Acronym inaccurate now
**Project: Code Statistics**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Lines of Java Code</td>
<td>1803</td>
</tr>
<tr>
<td>Classes</td>
<td>28</td>
</tr>
<tr>
<td>JSP</td>
<td>56</td>
</tr>
<tr>
<td>HTML, CSS, JS</td>
<td>2/6/0</td>
</tr>
</tbody>
</table>

- Clean up code
  - Need all 6 CSS files? Seems to be a lot of duplicates

---

**Project: Implementation**

- Should be done with bulk of implementation today
- **Testing**
  - Test each other’s code ASAP
  - I will do some testing too and look over the code
  - Fix bugs but NO NEW FUNCTIONALITY!

---

**Project: Documentation**

- Technologies used: Orbital Library
  - http://www.functologic.com/
- Handling of ‘v’ in answers

- Future work ideas?

---

**Project: Demo**

- Application functionality
- Typical situations of how a user would use the application
- Security features
  - What a malicious user can’t do
  - Example: my “auth” Selenium tests
    - Try to be devious!

---

**TODO**

- Project
  - Documentation, Bug Fixes, Demo on Monday