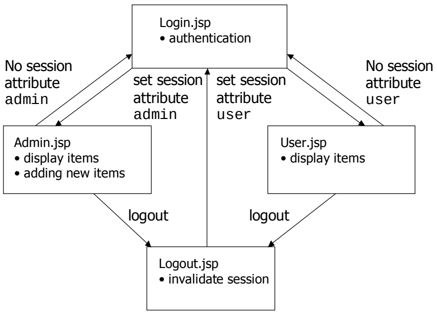
CS520 Web Programming

Introduction to MVC and Web Application Frameworks

Chengyu Sun

California State University, Los Angeles

Example – Items (JSP version)

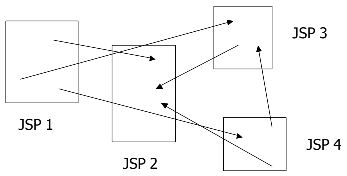


About Items (JSP version)

- Advantages
  - Easy to code
- Problems
  - ??

Model 1 Architecture

- JSPs + Java beans
  - JSPs for presentation
  - beans for business logic



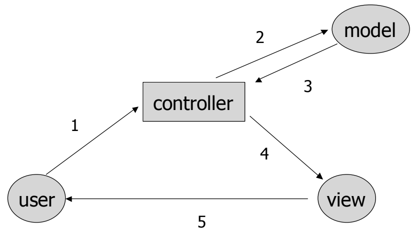
Items (Model 1 version)

- What would it look like??
- Compared to the JSP version, what problems are addressed and what still remain??

Model 2 Architecture

- Also know as Model-View-Controler (MVC) architecture
  - JSPs + beans + servlet
  - Beans for business logic – Model
  - JSPs for presentations – View
  - servlet for web logic – Controller
    - HTTP related processing, e.g. request, response, sessions etc.
    - *Request dispatching*

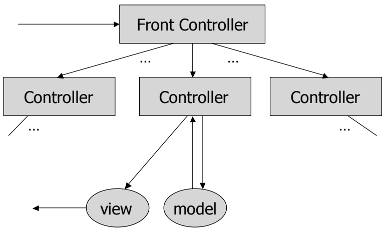
MVC Control Flow ...



... MVC Control Flow

1. Process request
2. Populate beans
3. Store results in request, session, or servlet context
4. Forward request to JSP page
5. Extract bean data from beans and display

Multiple Controllers



Design Pattern: Front Controller

◈Centralized point for request handling
- Avoid duplicate control logic
- Separate system control code from view creation code
- Single point of access control

Items (MVC version)

◈FrontController
- Login handler
- Logout handler
- User handler
- Admin handler

◈Models: Item and User

◈Views: Login.jsp, Admin.jsp, User.jsp

Admin Control Flow

1. Process request

2. Populate beans

```
List<Item> items = itemDao.getAllItems();
```

3. Store results in request, session, or servlet context

```
request.setAttribute( "items", items );
```

4. Forward request to JSP page

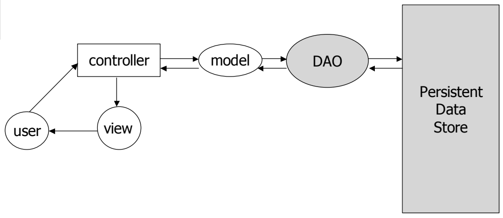
```
request.getRequestDispatcher( "WEB-INF/jsp/Admin.jsp" ).forward(
        request, response );
```

5. Extract bean data from beans and display

```
<c:forEach items="${items}" var="item">
…
```

More About The Example

- web.xml
- Data Access Object (DAO)

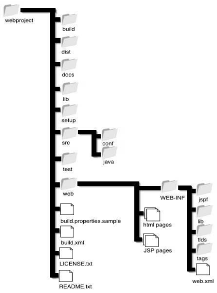


Design Pattern: Data Access Object (DAO)

- Encapsulate data access and manipulation in a separate layer
  - Provide a uniform data access API that is
    - Independent of persistent storage types e.g. RDBMS, OODB, XML, flat files etc.
    - Independent of persistent storage implementation, e.g. PostgreSQL, MySQL, Oracle etc.

Directory Structure …

- Recommended by SUN: http://java.sun.com/blueprints/code/projectconventions.html



… Directory Structure

- Recommended by This SUN:

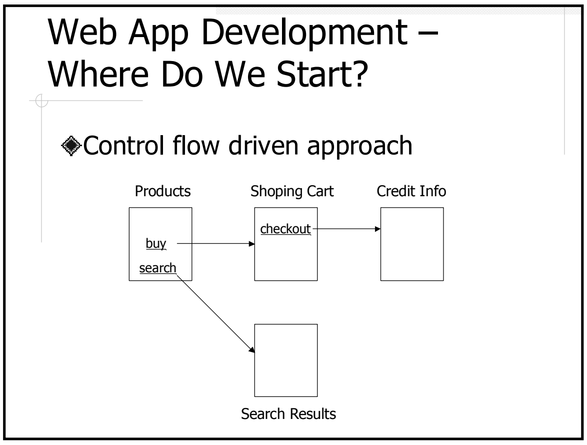
```
\webapp
        \build
        \dist
        \conf
        \doc
        \db
        \lib
                \compile
                \runtime
        \src
        \web
                \WEB-INF
                        \jsp
                        \tld
        build.xml
        build.properties.sample
```

Need for Web Application Frameworks

- Front controller
- Simplify creation of controllers
- Input validation
- Error and exception handling
- Transaction support
- Integration of common libraries
- …

Some Web Application Frameworks

- Struts
  - http://struts.apache.org/
- Spring
  - http://www.springframework.org/
  - More than a MVC framework
- Ruby on Rails
  - http://www.rubyonrails.org/



Web App Development – Where Do We Start?

◈Control flow driven approach

Products

buy

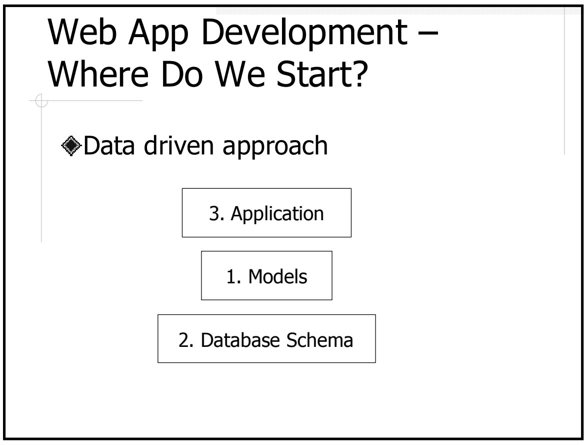
search

Shoping Cart

checkout

Credit Info

Search Results



Web App Development – Where Do We Start?

◈Data driven approach

3. Application

1. Models

2. Database Schema