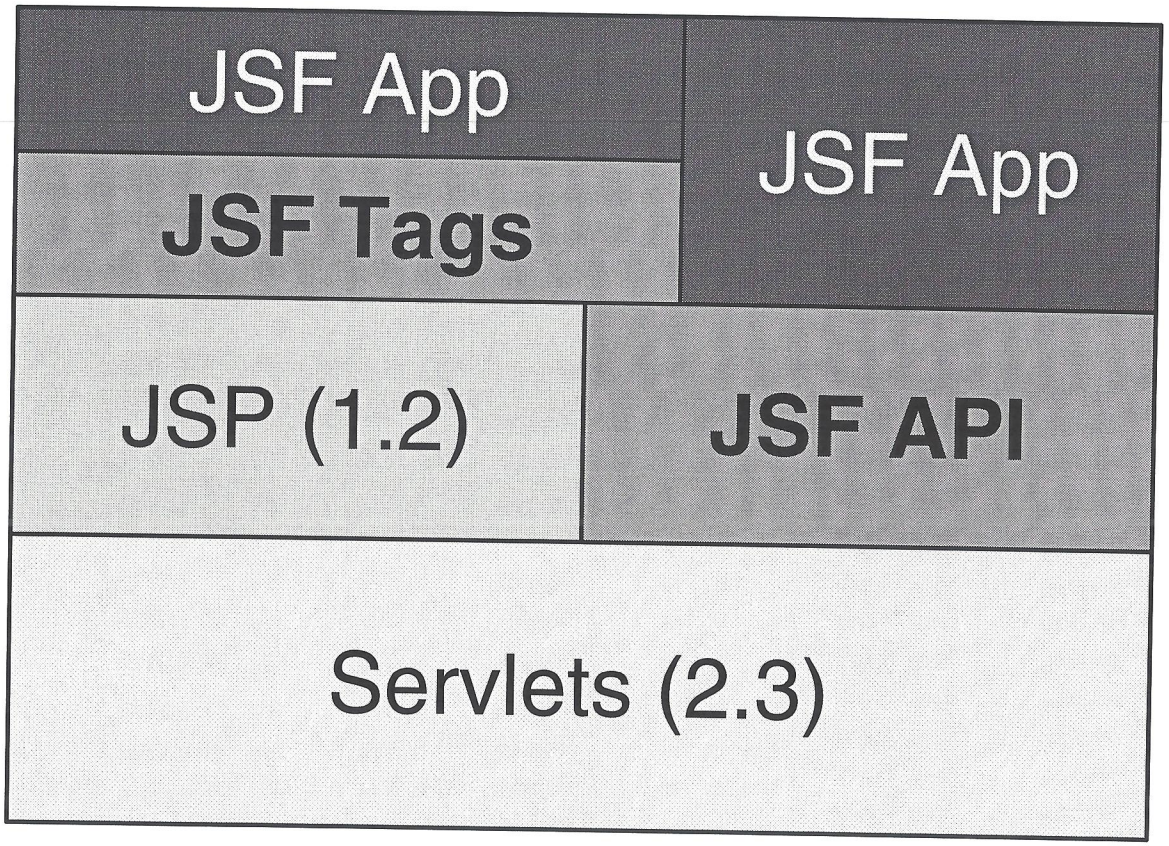
JSF Must Have...

- Extensible UI component model
- Event model
- Validation framework
- Page navigation & Error handling
- I18N & Accessibility

Here is the set of features we've defined to satisfy these requirements.

How JSF Fits In with other Web Technologies



Where Struts?
Any struts relics out there?

So What About Struts?

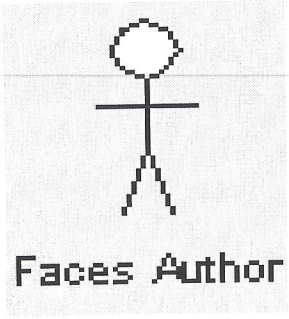
Struts: collection of JSPs, Navigation, some Bean access.

- Different design centers
- Some overlap — but we add components, events, robust Validation
- Possible integration
  - Struts apps use JSF components
- Pick solution best suited to expertise/task

· Mention Craig integration.
· If struts works for you, keep using it.

Roles in JavaServer Faces

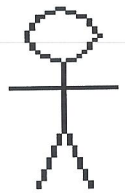
- Scenario 1: The Jack of All Trades



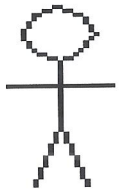
  - Uses pre-built JSF Components to build UI
  - App requires minimal back-end processing.
  - Quick and dirty web apps, ie, little inter-departmental tools.
  - Skills:
    - Good HTML, minimal JSP
    - Some graphic design
    - Simple, procedural Java programming

Roles in JavaServer Faces

- Scenario 2: The Mom and Pop Shop



Faces Author



Java Developer

  - Faces Author uses pre-built JSF Components to build UI
  - Java Developer provides business logic and back end code
  - Skills
    - Faces Author: HTML, minimal JSP
    - Java Developer: mid level Java programming, "model 2" training

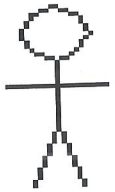
Roles in JavaServer Faces

- Scenario 3: Enterprise

Faces Author

– Faces Author uses pre-built and home grown JSF Components (from Faces Developer) to build UI

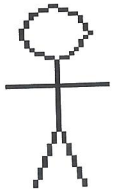
– Java Developer provides business logic and back end code



Java Developer

– Faces Developer creates app specific UI components for use by Faces Author

– Skills

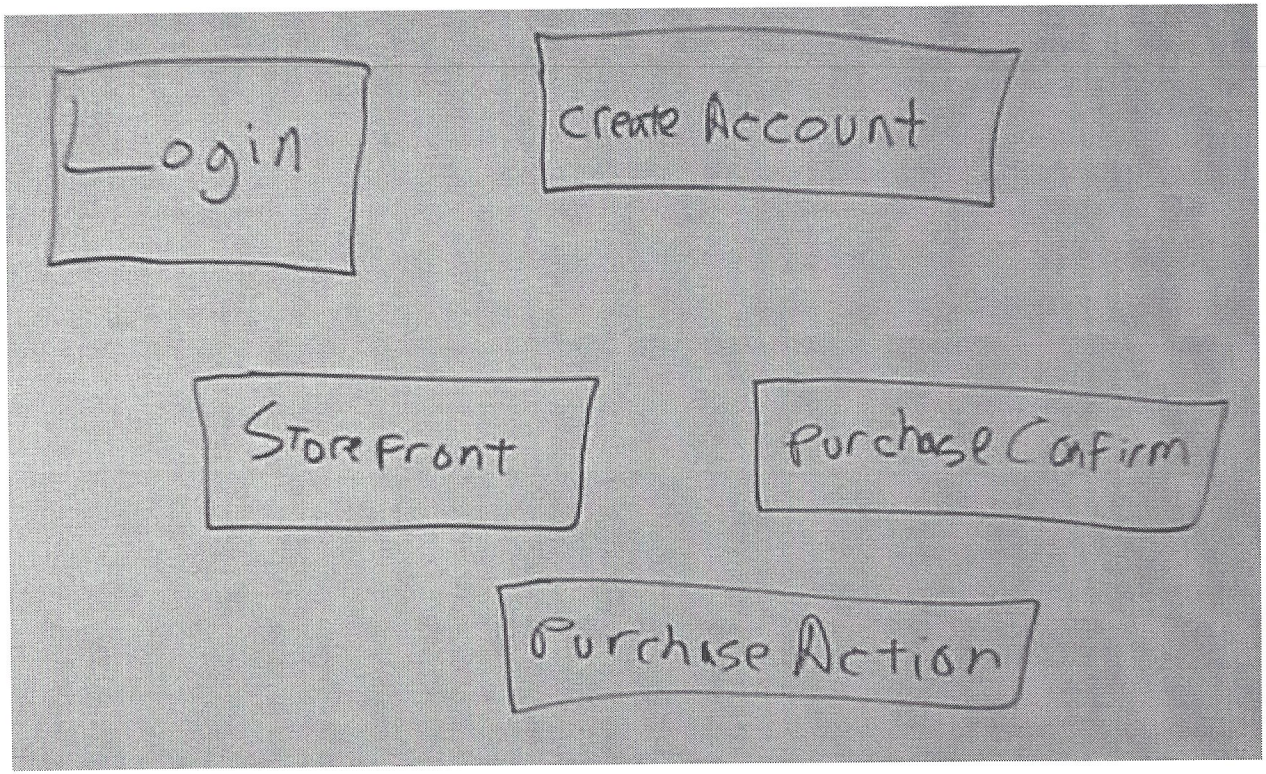


Faces Developer

- Faces Author, Java Developer: same
- Faces Developer: UI Component Design, JavaServer Faces internals

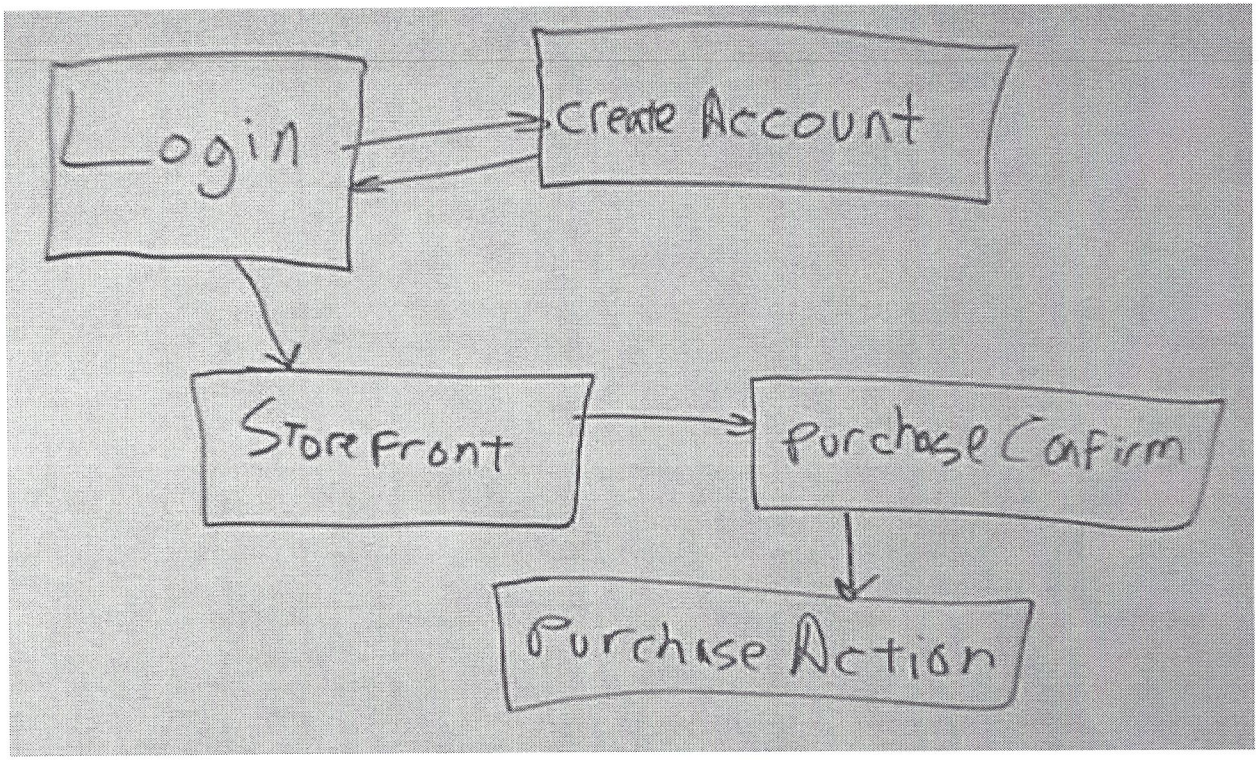
What Does a Faces App Look Like?

- A Collection of JSP Pages



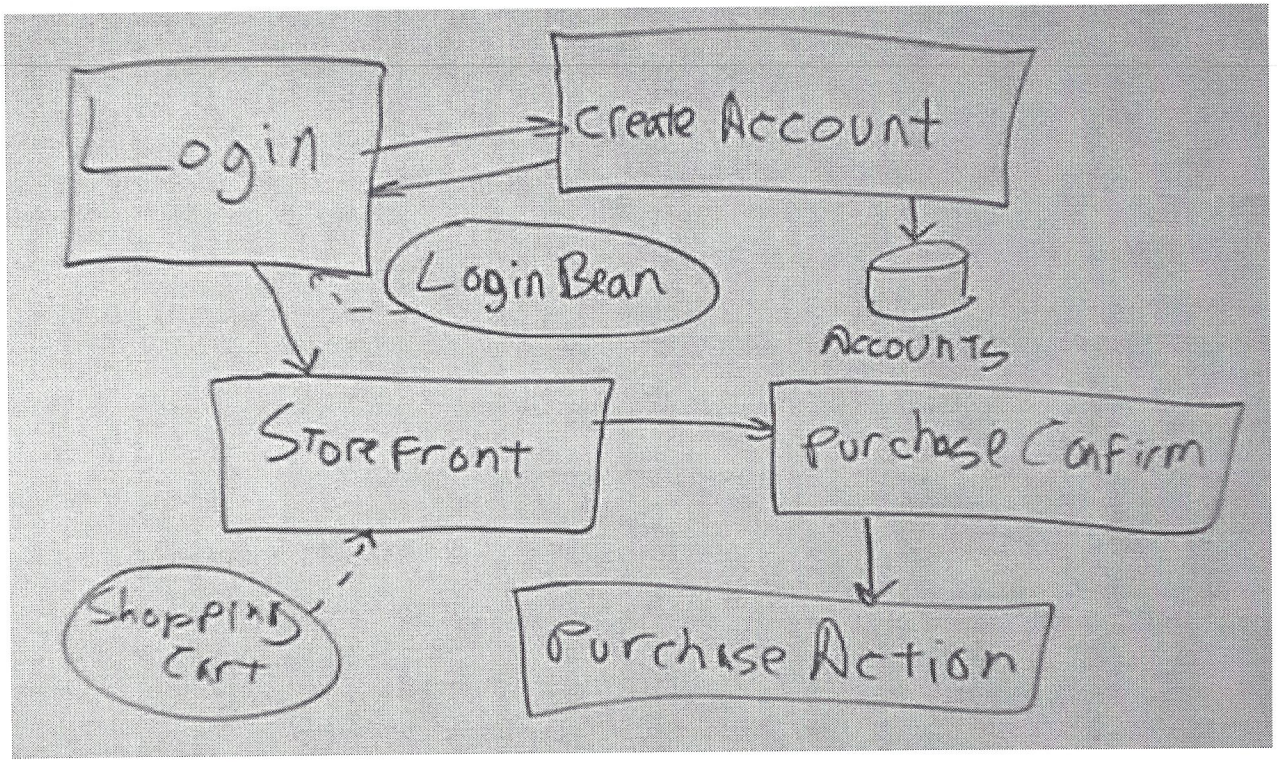
What Does a Faces App Look Like?

- The relationship between the pages



What Does a Faces App Look Like?

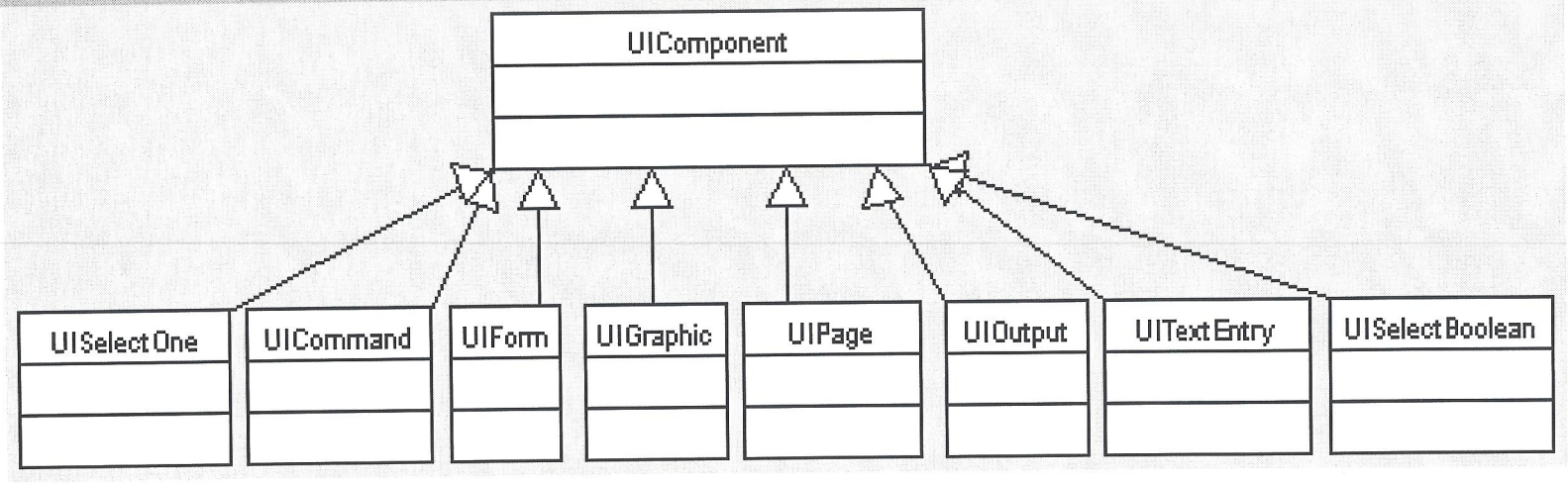
- Back end integration: Java, Database, etc.



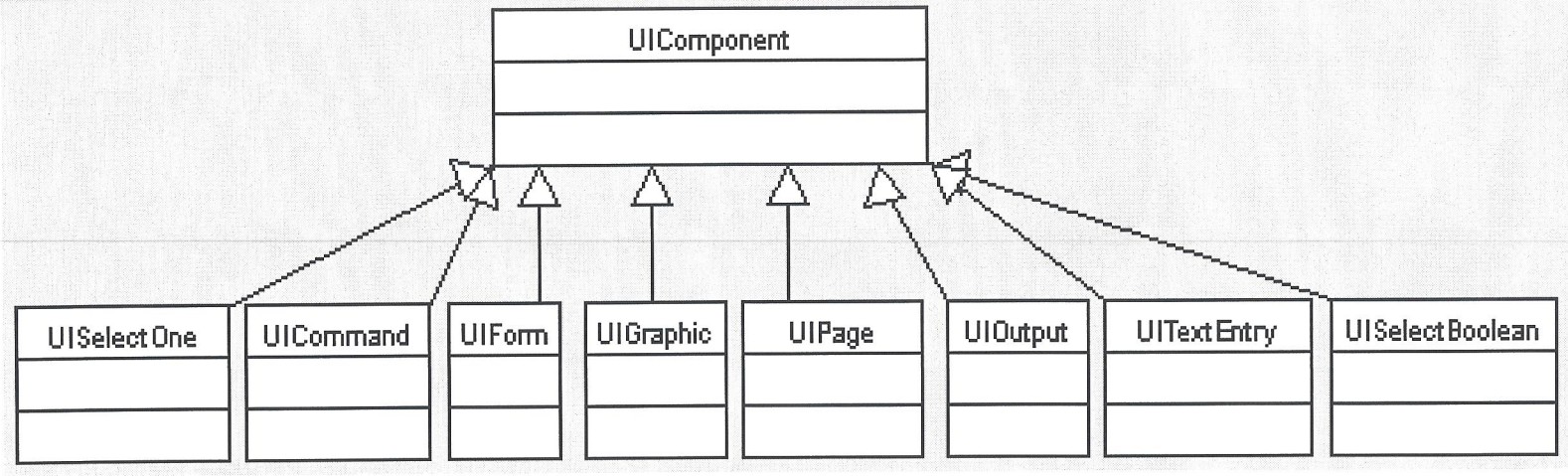
Key Value Add: UI Components

- A well defined, familiar idiom for UI design
- Extensible through composition, adding new components
- Accessible via HTML like JSP tags.

Key Value Add: UI Components



Key Value Add: UI Components



+

Faces JSP Custom Tag Library

Form, TextField, TextArea, CheckBox, Radio Button, Option List, Button, Label, Tree, etc.

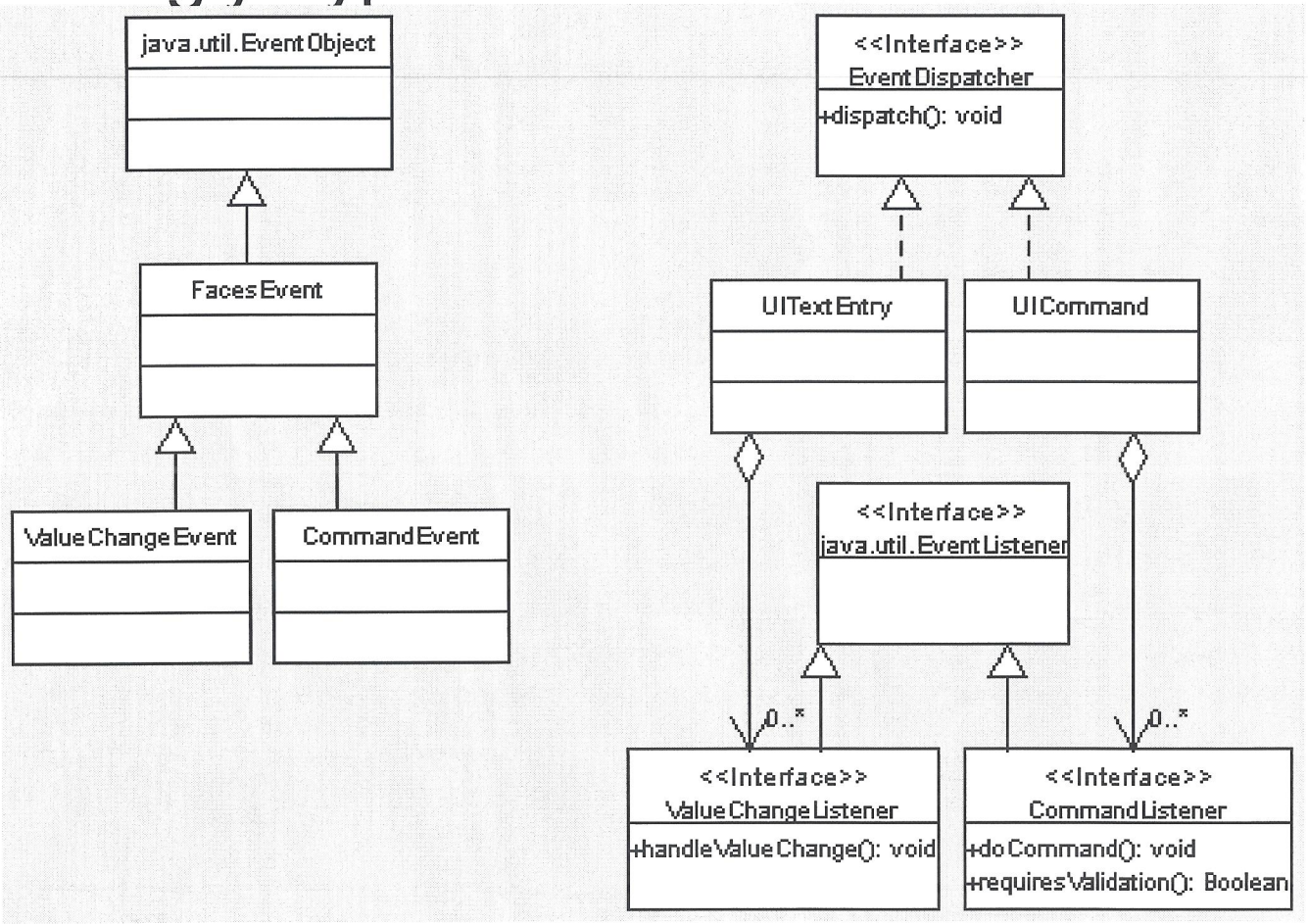
=

Key Value Add: UI Components

```
<faces:Form id="LoginForm" model="UserBean"
                                navigationMapId="LoginNavMap" >
<table> <tr>
  <td><faces:Output_Text id="name_label"  value="Name:" /></td>
  <td> <faces:TextEntry_Input id="userName"
                                model="$UserBean.userName" />
  </td></tr>

  <tr><td><faces:Output_Text id="passwd_label"
                                value="Password:" /></td>

  <td> <faces:TextEntry_Secret id="password"
                        model="$UserBean.password" size="10"/></td>
  </tr></table>
<p> <faces:Command_Button id="Login" label="login"
        commandListener="handleLogin" />

</faces:Form>
```

Key Value Add: Event Handling

- Strongly Typed Event and Listener Classes



Key Value Add: Event Handling

```
<faces:Listener id="loginListener" scope="session"
                    className="basic.EventHandler" />

<faces:TextEntry_Input id="userName" value="default"
                    valueChangeListener="loginListener" />

<faces:Command_Button id="login" label="login"
                    commandListener="loginListener" />
```

```
public class basic.EventHandler implements
        CommandListener,ValueChangeListener {

  public void handleValueChange(ValueChangeEvent event) { ... }

  public void doCommand(CommandEvent event, NavigationHandler nh) {...}

  public boolean requiresValidation(CommandEvent event) {...}
}
```

Key Value Add: Event Handling

- Don't have to deal with request params
- Events are strongly typed
- Familiar paradigm, easily links to other Java code.