Consider an auction web application that provides a page like this and allows the user to bid on an item, specified by an item ID that the user somehow (we do not care how) obtained.

[Image: Page A]

Validate the ID is a legit product ID, assuming `model.validID()` and similarly validate that the dollar amount is indeed a dollar amount (as opposed to, say, a non-numeric string).

Upon clicking “Place Bid”, the business logic of the web application uses a given `Integer function model.currentHighestBid(String id)` to find out what is the highest current bid placed earlier.

- If the current bid exceeds the highest earlier bid the user is sent to the Page B, a sample of which is displayed below.
- If it does not (exceed), the user is sent back to Page A, whereas a message like the one shown below also appears.
Upon clicking the “Buy” the business logic invokes a boolean function
model.buyWithCard(String CreditCard), which if it succeeds the customer is sent
back to Page A and otherwise he is sent back to Page B and a message “Provide another
card” is displayed. Ignore validation of the credit card form.
Write a struts program that implements the above. You may use any of the given model functions above. Provide the following pieces:

- `struts-config.xml` (we provide a template in the next page so that you do not write tags by hand)
- form beans, including validation (if you decide to use validation in form beans – you do not have to)
- action beans
- html code of page A – you may break it down into more pages if it facilitates the solution.

You do not have to write the html code of Page B, just indicate which form bean it sets.
Solution

Here is the html of Page A1, ignoring most format and presentation details.

```html
<%@ taglib uri="http://struts.apache.org/tags-html" prefix="html" %>
<html>
<title>Page A1: First visit to the bid page</title>
<body>
<h1>Here you place bids for items specified by ID</h1>
<p>
<html:errors/>
<html:form action="/PlaceBid">
<table>
<tr>
<td>Item ID:</td>
<td> <html:text property="itemID" /></td>
</tr>
<tr>
<td>Bid:</td>
<td> <html:text property="bid" /></td>
</tr>
<tr>
<td>.</td> 
<td> <html:submit value="Place Bid"/></td></tr>
</table>
</html:form>
</body>
</html>
```

Here is the html of Page A2, also ignoring most format and presentation details.

```html
<%@ taglib uri="http://struts.apache.org/tags-html" prefix="html" %>
<html>
<title>Page A2: Visit to the bid page upon failure of previous bid</title>
<body>
<h1>Here you place bids for items specified by ID</h1>
<p>
The highest current bid is 
<%= (Integer) request.getAttribute("highestCurrentBid") %>.  
Provide higher bid.
</p>
<html:form action="/PlaceBid">
<table>
<tr>
<td>Item ID:</td>
<td> <html:text property="itemID"  
value="<%= (Integer)request.getAttribute("itemID")%>" /></td>
</tr>
<tr>
<td>Bid:</td>
<td> <html:text property="bid" /></td>
</tr>
<tr>
<td>.</td> 
<td> <html:submit value="Place Bid"/></td></tr>
</table>
</html:form>
</body>
</html>
```
Form Beans.

public class BidForm extends ActionForm {
    private String itemID = null;
    private String bid = null;

    public ActionErrors validate(
        ActionMapping mapping,
        HttpServletRequest request) {
        ActionErrors errors = new ActionErrors();
        if ((bid == null) || (bid.isEmpty())) {
            errors.add("bid",
                new ActionMessage("errors.required", "Bid"));
        } else {
            try {
                Integer.parseInt(bid);
            } catch (NumberFormatException e) {
                errors.add("bid",
                    new ActionMessage("errors.notanumber", "Bid"));
            }
        }
        if ((itemID == null) || (itemID.isEmpty())) {
            errors.add("itemID",
                new ActionMessage("errors.required", "Item ID"));
        } else if (!model.validID(itemID) {
            errors.add("itemID",
                new ActionMessage("errors.invalid", "Item ID"));
        }
        return errors;
    }
}

Action Classes.

public class PlaceBidAction extends Action {

    public ActionForward execute(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) {
        BidForm bForm = (BidForm) form;
        Integer highestCurrentBid = 
            model.currentHighestBid(bForm.getItemID()) {
            if (Integer.parseInt(bForm.getBid()) > highestCurrentBid) {
                return null;
            }
        }
request.setAttribute("itemID", bForm.getItemID());
request.setAttribute("yourBid", bForm.getBid());

    return mapping.findForward("highestBid");
} else {
    request.setAttribute("itemID", bForm.getItemID());
    request.setAttribute("highestCurrentBid", highestCurrentBid);

    return mapping.findForward("notHighestBid");
}

}
}

public class BuyAction extends Action {

    public ActionForward execute(
            ActionMapping mapping,
            ActionForm form,
            HttpServletRequest request,
            HttpServletResponse response) {

        CreditCardForm cForm = (CreditCardForm) form;
        boolean success = model.buyWithCard(cForm.getCreditCard());

        if (success) {
            return mapping.findForward("success");
        } else {
            request.setAttribute("itemID", cForm.getItemID());

            return mapping.findForward("failure");
        }

    }
}
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE ...ignore...>
<struts-config>
<!-- ---------------------Form Bean Definitions -->
<!-- You may have more or less than the 3 form beans for which we have made space -->
<form-beans>
  <form-bean
    name="BidForm"
    type="forms.BidForm" />
  <form-bean
    name="CreditCardForm"
    type="forms.CreditCardForm"/>
</form-beans>

<!-- ---------------- Action Mapping Definitions -->
<action-mappings>
  <action
    path="/FirstBid"
    type=org.apache.struts.actions.ForwardAction
    parameter="Bid1.jsp" />
  <action
    path="/PlaceBid"
    type="actions.PlaceBidAction"
    name="BidForm"
    scope="request"
    validate=true
    input="Bid1.jsp" >
    <forward
      name="highestBid"
      path="Buy.jsp" />
    <forward
      name="notHighestBid"
      path="Bid2.jsp" />
  </action>
  <action
    path="/Buy.do"
    type="actions.BuyAction"
    name="CreditCardForm"
    scope="request"
    validate=true
    input="Buy.jsp" >
    <forward
      name="success"
      path="Bid.jsp" />
    <forward
      name="" path="/" />
  </action>
</action-mappings>
name="failure"
  path="Buy.jsp" />
</action>

</action-mappings>

</struts-config>
Design with separate jsp’s for the first bid and second bid => Simpler jsp, richer workflow
Still same jsp for first attempt to buy with credit and consequent attempts