I chose to do question 26.3 and 26.4

Exercise 26.3

Index.xhtml

<?xml version='1.0' encoding='UTF-8' ?>

<!-- index.xhtml -->

<!-- Registration form that demonstrates various JSF components -->

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

      xmlns:h="http://java.sun.com/jsf/html"

      xmlns:f="http://java.sun.com/jsf/core">

    <h:head>

        <title>Sample Registration Form</title>

    </h:head>

    <h:body>

        <h:form>

            <h1>Registration Form</h1>

            <p>Please fill in all fields and click Register</p>

            <h:panelGrid columns="4" style="height: 96px; width:456px;">

                <h:graphicImage name="fname.png" library="images"/>

                <h:inputText id="firstNameInputText" required="true"

                             requiredMessage="Name must be fewer than 30 characters"

                             value="#{validationBean.firstName}"

                             validatorMessage="Name must be less than 30 characters">

                    <f:validateLength maximum="30" />

                </h:inputText>

                <h:graphicImage name="lname.png" library="images"/>

                <h:inputText id="lastNameInputText" required="true"

                             requiredMessage="Name must be fewer than 30 characters"

                             value="#{validationBean.lastName}"

                             validatorMessage="Name must be less than 30 character">

                    <f:validateLength maximum="30"/>

                </h:inputText>

                <h:graphicImage name="email.png" library="images"/>

                <h:inputText id="emailInputText" required="true"

                             requiredMessage="Please enter a valid email number"

                             value="#{validationBean.email}"

                             validatorMessage="Invalid email format">

                    <f:validateRegex pattern=

                                     "\w+([-+.']\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*" />

                </h:inputText>

                <h:graphicImage name="phone.png" library="images"/>

                <h:inputText id="phoneInputText" required="true"

                             requiredMessage="Please enter a valid phone number"

                             value="#{validationBean.phone}"

                             validatorMessage="Invalid phone number format">

                    <f:validateRegex pattern=

                                     "((\(\d{3}\) ?)|(\d{3}-))?\d{3}-\d{4}" />

                </h:inputText>

            </h:panelGrid>

            <p><h:graphicImage name="publications.png" library="images"/>

                <br/>Which book would you like information about?</p>

            <h:selectOneMenu id="booksSelectOneMenu">

                <f:selectItem itemValue="CHTP"

                              itemLabel="C How to Program" />

                <f:selectItem itemValue="CPPHTP"

                              itemLabel="C++ How to Program" />

                <f:selectItem itemValue="IW3HTP"

                              itemLabel="Internet &amp; World Wide Web How to Program" />

                <f:selectItem itemValue="JHTP"

                              itemLabel="Java How to Program" />

                <f:selectItem itemValue="VBHTP"

                              itemLabel="Visual Basic How to Program" />

                <f:selectItem itemValue="VCSHTP"

                              itemLabel="Visual C# How to Program" />

            </h:selectOneMenu>

            <p><h:outputLink value="http://www.deitel.com">

                    Click here to learn more about our books

                </h:outputLink></p>

            <h:graphicImage name="os.png" library="images"/>

            <h:selectOneRadio id="osSelectOneRadio">

                <f:selectItem itemValue="WinVista" itemLabel="Windows Vista"/>

                <f:selectItem itemValue="Win7" itemLabel="Windows 7"/>

                <f:selectItem itemValue="OSX" itemLabel="Mac OS X"/>

                <f:selectItem itemValue="Linux" itemLabel="Linux"/>

                <f:selectItem itemValue="Other" itemLabel="Other"/>

            </h:selectOneRadio>

            <h:commandButton value="Register"/>

            <h:outputText escape="false" value="#{validationBean.result }"/>

        </h:form>

    </h:body>

</html>

Validation.java

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

/\*\*

 \*

 \* @author eg08014

 \*/

package validation;

import java.io.Serializable;

import javax.faces.bean.ManagedBean;

@*ManagedBean*(name= "validationBean")

public class ValidationBean

        implements *Serializable* {

    private *String* firstName;

    private *String* lastName;

    private *String* email;

    private *String* phone;

    // return the name String

    public *String* getFirstName() {

        return firstName;

    } // end method getName

    // set the firstName String

    public *void* setFirstName(*String* *firstName*) {

        this.firstName

                = firstName;

    } // end method setFirstName

    // return the lastName String

    public *String* getLastName() {

        return lastName;

    } // end method getLastName

    // set the lastName String

    public *void* setLastName(*String* *lastName*) {

        this.lastName

                = lastName;

    } // end method setLastName

    // return the email String

    public *String* getEmail() {

        return email;

    } // end method getEmail

    // set the email String

    public *void* setEmail(*String* *email*) {

        this.email

                = email;

    } // end method setEmail

    // return the phone String

    public *String* getPhone() {

        return phone;

    } // end method getPhone

    // set the phone String

    public *void* setPhone(*String* *phone*) {

        this.phone

                = phone;

    } // end method setPhone

    // returns result for rendering on the client

    public *String* getResult() {

        if (lastName

                != null

                && firstName

                != null

                && email

                != null

                && phone

                != null) {

            return "<p style=\"background-color:yellow;width:200px;"

                    + "padding:5px\">LastName: "

                    + getLastName()

                    + "<br/>FirstName"

                    + getFirstName()

                    + "<br/>E-Mail: "

                    + getEmail()

                    + "<br/>Phone: "

                    + getPhone()

                    + "</p>";

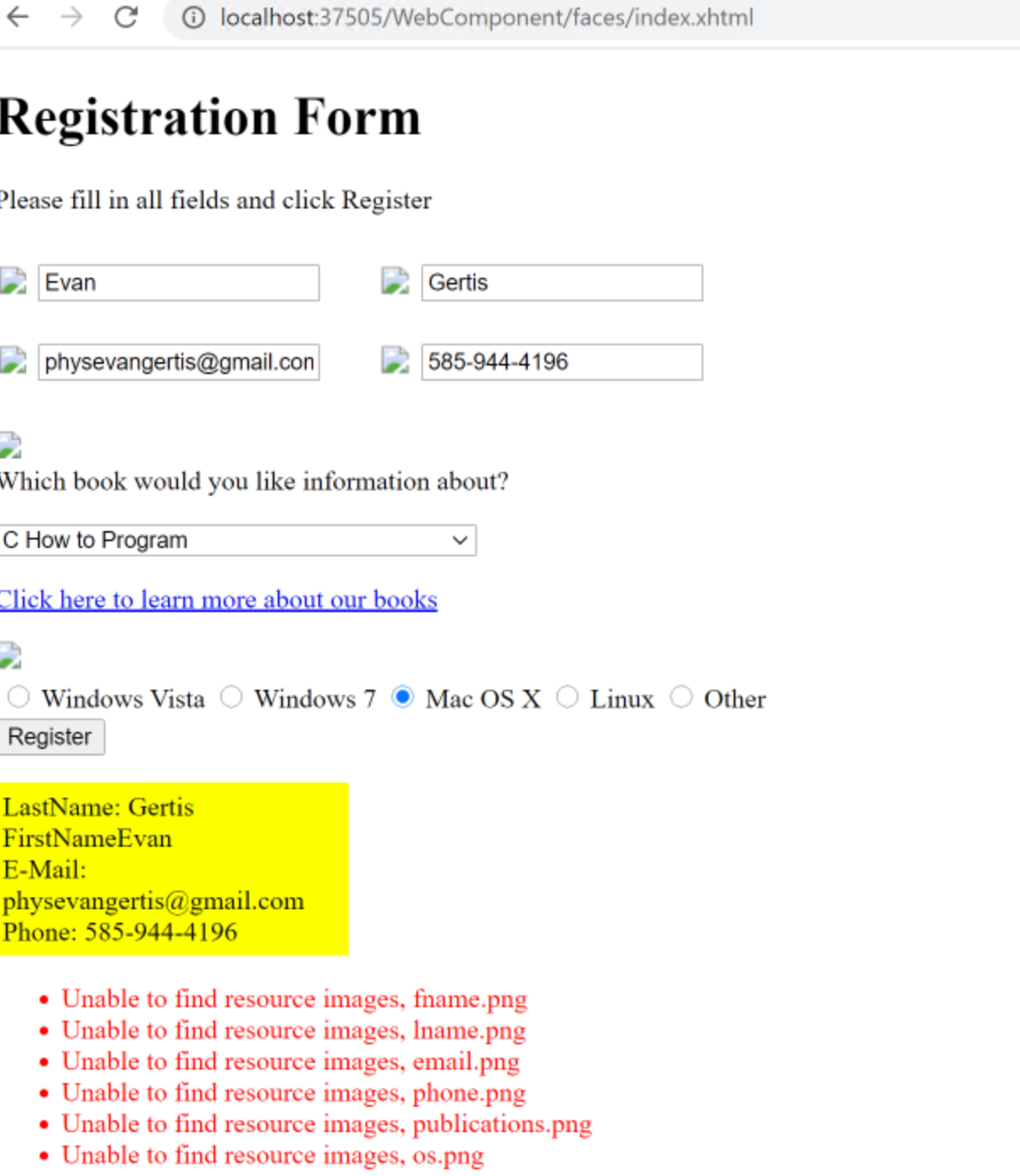
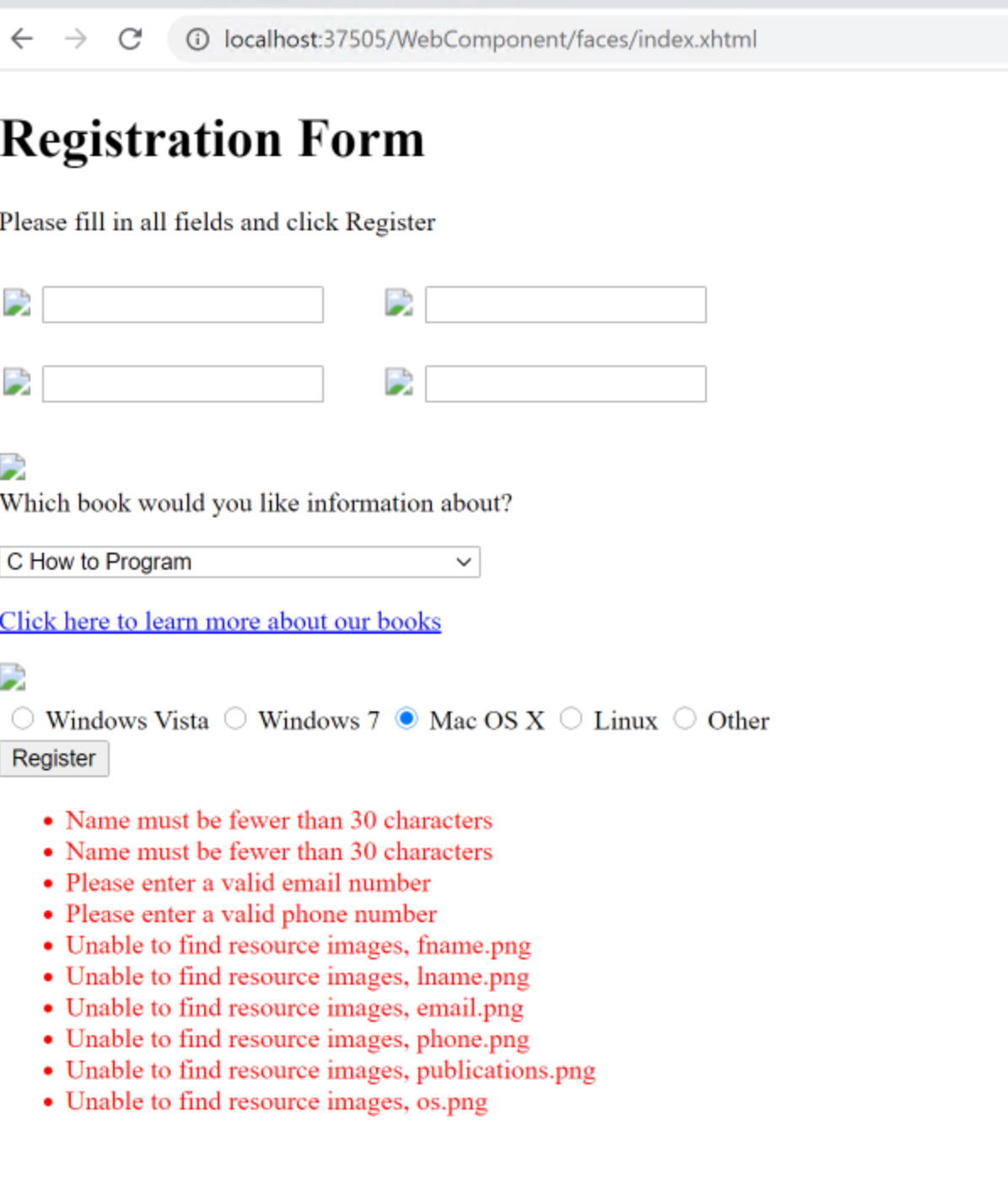
        } else {

            return ""; // request has not yet been made

        }

    } // end method getResult

} // end class ValidationBean



Exercise 26.4

Index.xhtml

<!-- Allow the user to select a topic -->

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

      xmlns:h="http://java.sun.com/jsf/html"

      xmlns:f="http://java.sun.com/jsf/core"

      xmlns:c="http://xmlns.jcp.org/jsp/jstl/core">

    <h:head>

        <title>Topic Selection Page</title>

    </h:head>

    <h:body>

        <h1>Welcome to Sessions!</h1>

        <p>You have made #{selectionsBean.numberOfSelections} selection(s)</p>

        <h3>Make a Selection and Press Submit</h3>

        <h:form>

            <h:selectOneRadio id="topicSelectOneRadio" required="true"

                              requiredMessage="Please choose a topic, then press Submit"

                              value="#{selectionsBean.selection}">

                <f:selectItem itemValue="java" itemLabel="Java"/>

                <f:selectItem itemValue="cpp" itemLabel="C++"/>

                <f:selectItem itemValue="iphone" itemLabel="iPhone"/>

                <f:selectItem itemValue="android" itemLabel="Android"/>

            </h:selectOneRadio>

            <p><h:commandButton value="Submit"/></p>

        </h:form>

        <p><h:outputLink value="shoppingCart.xhtml">

                Click here to view your shopping cart

            </h:outputLink></p>

    </h:body>

</html>

shoppingCart.xhtml

<?xml version='1.0' encoding='UTF-8' ?>

<!-- index.xhtml -->

<!-- Allow the user to select a topic -->

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

      xmlns:h="http://java.sun.com/jsf/html"

      xmlns:f="http://java.sun.com/jsf/core"

      xmlns:c="http://xmlns.jcp.org/jsp/jstl/core">

    <h:head>

        <title>Topic Selection Page</title>

    </h:head>

    <h:body>

        <h1>Shopping Cart</h1>

        <ul>

            <c:forEach var="item" items="${selectionsBean.selections}">

                <li>

                    <label>Title: #{item.title}</label>

                    <p>Genre: #{item.genre}</p>

                    <p>Price: #{item.price}</p>

                </li>

            </c:forEach>

        </ul>

        <p>Total: #{selectionsBean.selections.stream().map(book -> book.price).sum()}</p>

        <p><h:outputLink value="index.xhtml">

                Click here to go back to selections

            </h:outputLink></p>

    </h:body>

</html>

shoppingCart.xhtml

<?xml version='1.0' encoding='UTF-8' ?>

<!-- index.xhtml -->

<!-- Allow the user to select a topic -->

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

      xmlns:h="http://java.sun.com/jsf/html"

      xmlns:f="http://java.sun.com/jsf/core"

      xmlns:c="http://xmlns.jcp.org/jsp/jstl/core">

    <h:head>

        <title>Topic Selection Page</title>

    </h:head>

    <h:body>

        <h1>Shopping Cart</h1>

        <ul>

            <c:forEach var="item" items="${selectionsBean.selections}">

                <li>

                    <label>Title: #{item.title}</label>

                    <p>Genre: #{item.genre}</p>

                    <p>Price: #{item.price}</p>

                </li>

            </c:forEach>

        </ul>

        <p>Total: #{selectionsBean.selections.stream().map(book -> book.price).sum()}</p>

        <p><h:outputLink value="index.xhtml">

                Click here to go back to selections

            </h:outputLink></p>

    </h:body>

</html>

Selections.java

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package sessiontracking;

/\*\*

 \*

 \* @author eg08014

 \*/

import java.io.Serializable;

import java.util.HashMap;

import java.util.Set;

import java.util.ArrayList;

import java.util.List;

import java.util.HashSet;

import javax.faces.bean.ManagedBean;

import javax.faces.bean.SessionScoped;

@*ManagedBean*(name="selectionsBean")

@*SessionScoped*

public class Selections {

    // map of topic to book titles

    public static class Book {

        public Book(*String* *genre*, *String* *title*, *int* *price*){

            this.genre = genre;

            this.title = title;

            this.price = price;

        }

        public *String* getTitle(){

            return title;

        }

        public *String* getGenre(){

            return genre;

        }

        public *int* getPrice(){

            return price;

        }

        public *String* title;

        public *String* genre;

        public *int* price;

    }

    private static final *HashMap*<*String*,*Book*> booksMap = new *HashMap*<*String*,*Book*>();

    static

    {

        booksMap.put("java",new Book("java","Java How to Program",10));

        booksMap.put("iphone",new Book("iphone","How to program iOS",20));

        booksMap.put("android",new Book("android", "How to program android", 20));

        booksMap.put("cpp",new Book("cpp","C++ How to program", 15));

    }

    private static final *HashMap*<*String*, *Book*> selectionsMap = new *HashMap*<*String*, *Book*>();

    static

    {

        selectionsMap.put("java",new Book("java","Java How to Program",10));

        selectionsMap.put("cpp", new Book("iphone","How to program iOS",20));

        selectionsMap.put("iphone", new Book("android", "How to program android", 20));

        selectionsMap.put("android", new Book("cpp","C++ How to program", 15));

    }

    private *List*<*Book*> selections = new *ArrayList*<*Book*>();

    private *String* selection;

    public *int* getNumberOfSelections(){

        return selections.size();

    }

    public *String* getSelection(){

        return selection;

    }

    public *void* setSelection(*String* *topic*){

*String* genre = selectionsMap.get(topic).genre;

        selection = selectionsMap.get(topic).title;

        selections.add(booksMap.get(genre));

    }

    public *List*<*Book*> getSelections(){

        return selections;

    }

}

