Used Libraries :

Hibernate :

1- Installed the JBoss from Eclipse market.

2- Downloaded the "hibernate-release-5.2.13.Final" and used all .jar files inside the

"lib>required"

3- Added the libraries to the classpath.

Oracle DB Connection:

Downloaded the "ojdbc6.jar" from oracle web page that is for Oracle 11g 2.0.2.0 version.

\*\*Placed the jar under WEB-INF>lib to make it work. (It is about TomCat.)

And added the jar to build path. It is done by referring :

[Reference for ClassNotFoundException.](http://www.java67.com/2015/06/how-to-fix-javalangclassnotfoundexception-oracle.jdbc.driver.oracledriver.html)

Primefaces:

1-The primefaces-6.1.jar need to be placed under "WebContent\WEB-INF\lib"

2-Need to place these references inside the page (index.xhtml) :

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

*xmlns:p="http://primefaces.org/ui"*

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

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

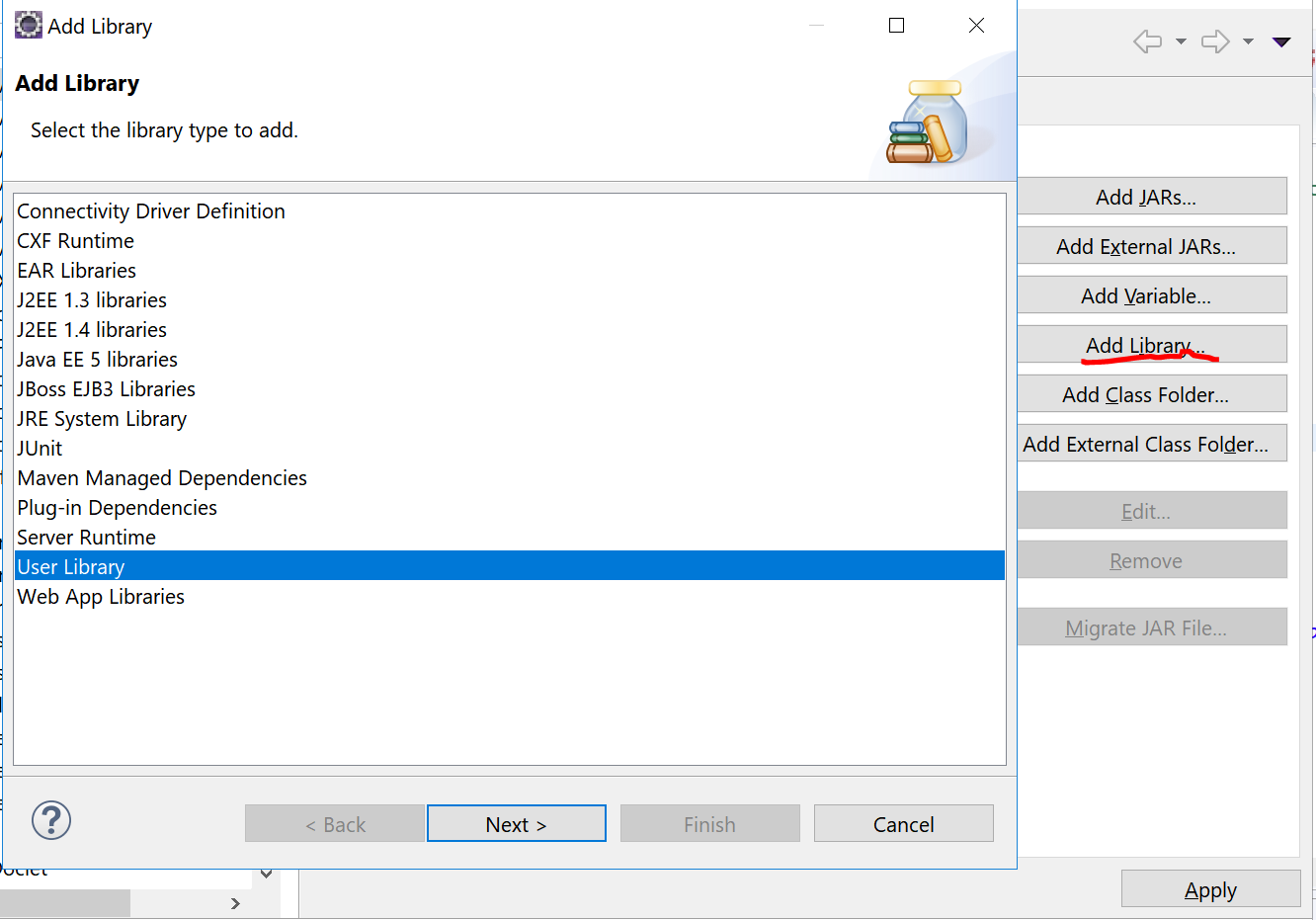
JSF:

Used the mojarra libraries :

[jsf-impl-2.2.9.jar](http://central.maven.org/maven2/com/sun/faces/jsf-impl/2.2.9/jsf-impl-2.2.9.jar)

[jsf-api-2.2.9.jar](http://central.maven.org/maven2/com/sun/faces/jsf-api/2.2.9/jsf-api-2.2.9.jar)

In Java Build Path :



And create the new user library.

Faced Problems and Solutions

1-No mime type could be found for file dynamiccontent. To resolve this, add a mime-type mapping to the applications web.xml

Added the following part to "*Web.xml" :*

*<mime-mapping>*

*<extension>jsp</extension>*

*<mime-type>application/xml</mime-type>*

*</mime-mapping>*

JSF Experiences

JavaServer Faces Lifecycle

JSF Component Elements

*<h: commandButton action = "page1" value = "Navigate Page 1"/>*

Action : Is the value, when used as an existing page name, the JSF will find the related "page1.xhtml" and navigate to it. In case of a function call, will take the return value instead page1.

*<h: commandButton action = "#{className.getPageName}" value = "Navigate"/>*

*<f:param name = "paramName" value = "paramValue"/>*

Value : Is the name that will appear on the button

*<p:commandButton value="Submit" action="#{reservation.getReservation}" update="resID,resNum" />*

Update: Here id of the elements given are going to be updated without post (ajax) with the action of the including element. Multiple updating elements can be placed by a comma.

*<p:commandButton value= "Page1" action="#{reservation.returnPageName(1)}">*

Immediate : Attribute for <p:commandbutton immediate = "true" />

When applied the button will not be effected from the validation issues on the page and will execute it's duty.

Updating An element of template.xhtml from particular xhtml

Placing the element that desired to be update inside the tags :

<h:form id=*"form\_template"*> </h:form>

ViewScoped Bean and SessionScoped Bean

The SessionScoped Managed Bean's instance is able to be used on all other pages in the same HTML Session. So the all page components that related with the members of this managed bean will share the same values.

The ViewScoped Managed Bean's instance is valid while on the same jsp view postbacks without any navigation/redirect. So each element that related with this managed bean's members, will share the same values.

Composition (Template xhtml)

Individual content pages :

*<ui:composition template = "./template.xhtml>*

*<ui:define name="title">...</ui:define>*

*<ui:define name="top">...</ui:define>*

*<ui:define name="left">...</ui:define>*

*<ui:define name="content">...</ui:define>*

Template page template.xhtml :

Has the div's and related css classes here.

<title><ui:insert name = "title"></ui:insert></title>

<ui:insert name = "top">Top Section</ui:insert>

<ui:insert name = "left">Left Section</ui:insert>

<ui:insert name = "content">Main Content</ui:insert>

Dependency Injection (Reaching Another Bean)

*public class OrderBean {*

*@ManagedProperty(value="#{loginBean}")*

*private LoginBean loginBean; // +setter*

*}*

Cannot define cyclic references.

*- Managed bean reservation contains cyclic references. Evaluation path: reservation -> newreservation -> reservation.*

Git Experiences

Creating New Repository

* touch README (can be created manually on windows etc.)
* git add README
* git commit -m 'first commit'
* git remote add origin https://github.com/username/repo.git
* git push -u origin master

Making Commits

* git add -A : stages **All**
* git add . : stages new and modified, **without deleted**
* git add -u : stages modified and deleted, **without new**
* **git push origin master**

Other Commands

git diff xxx : Show the differences between local and repository

CSS Experiences

* Using a '#' before name, means that applies to an ID
* Using a '.' before name, means that is a class, and can be used for any of items.