**CLC Mini-Project 3**

**Mat Paschall**

**11/26/17**

Question 1

JSF respond to user generated events using classes that have registered themselves to listen for events. These classes can then perform an action or call another method that can perform an action when the event is fired. All event classes extend the javax.faces.event.FacesEvent class. As this process is completed in a web browser which has no server access, the event can be delayed; Therefore, these events are handled in separate phases. As there may be value changes on the client side that the server is unaware of, changes should be queued, and only notify listeners by firing an ActionEvent to call a broadcast() method once all data is queued. The code for listening to an event may look something like the following.

//add UInput as listener

<h:commandButton value="Add"

actionListener="#{UInput.handleAdd}" />

//handler for user input, fired when a button is pressed on the form.

public class UInput extends javax.faces.event.FacesEvent {

private String name;

private String password;

Handle();

//calls the QClass method add, to add variables to a queue to check against server info

Public void Handle(ActionEvent e){

QClass.add(String name);

QClass.add(String password);

}

}

//adds items to the queue in preparation for checking data

Public QClass extends javax.faces.event.FacesEvent {

String data;

Queue<String> q = new ArrayDeque<String>();

//method to add data to queue

Public void add(string data){

q.add(data);

}

//The queue would then be sent to the server to check and verify correct input.

Question 2

WebLogic

WebLogic is a great server for deploying e-commerce web sites. These web sites allow many customers to make purchases, and must always be online. WebLogic features automated fail-over, which automatically moves an application to a different server if a server failure or maintenance causes a server to go down.

WebLogic would be a good choice for a cloud storage website. These store customer files and need to be available at all times, in case a customer needs to access data.

WebLogic could also be used to host a company intranet site, as the information stored can be accessed on premises and in the cloud. This would allow an employee to access work information even when they are at home.

Glassfish

Glassfish can be used to manage Java EE applications. It can launch and run Java applications inside of a web browser.

Glassfish can also be used as a web server that takes HTTP requests. This means that Glassfish can actively take requests from remote users, and serve up web pages in response.

Since Glassfish can act as a web server and application server, it can be user to host web pages for a variety of users when paired with server hardware. This would allow Glassfish to work in tandem with Java EE applications using EJBs and JSP to display websites to users upon request.

References:

[www.onjava.com/pub/a/onjava/excerpt/JSF\_chap8/index.html?page=3](http://www.onjava.com/pub/a/onjava/excerpt/JSF_chap8/index.html?page=3)

<https://www.oracle.com/middleware/weblogic/index.html>

<https://www.oracle.com/middleware/weblogic/enterprise-edition.html>

<http://www.oracle.com/technetwork/middleware/glassfish/overview/index.html>