**Assignment 3 – ITC357**

**Discussion on development:**

Developing the jsp and servlet files was very hard. The logic was very different.

**What I found hard to get my head around:**

Using the request dispatcher forwarding method, it didn’t update the URL in the address and I researched for hours on why this and why that. Also using sendRedirect forgets the all the parameters so it was a no go. I was only towards the end that I realised the jsp shouldn’t be directly accessible, thus, using the servlet only, the URL never had to change. This wasn’t implemented correctly in my assignment and mine was designed badly by making the user go the registration jsp first, otherwise, it produced an error.

**Getting data to stay:**

Where is says in the subject outline for the error to be produced on the same page, I read this as the data already entered must not disappear and remain there. Thus I tried to find a very clean way of displaying data in the jsp. During research I found, I main topics, java beans, which we apparently studying later, so I didn’t use and Expression Language, which I use. I didn’t utilise EL much and but I tried much as possible to use it. For the text boxes it worked very well, but with radio buttons and check boxes, I couldn’t find a way of using EL efficiently. So ended up using a scriptlet which are very bad apparently and I agree, it’s very messy and although, I indented my code, I had to use the search tool to find where I wanted to go. Coding java inside a jsp is bad.

**Default checkbox:**

Besides the java code I used to keep the boxes ticked if there was an error, setting a default box was a quite a challenge. I should have left this, regrettably. Getting a jsp to tell the difference between a fresh call and a submit call on error, was fundamentally impossible. This is due to the way html form work, if a check box isn’t ticked, it’s not submitted and compared to a fresh load of the page, the servlet never gets the checkbox value. Granted, though, if another checkbox was ticked, u could find out by iterating through the array. The problem was when there was no array, none ticked. So what I ended up doing was adding a hidden field below the default value, so there as always a value. If it was received, the form was used, if-not like-wise.

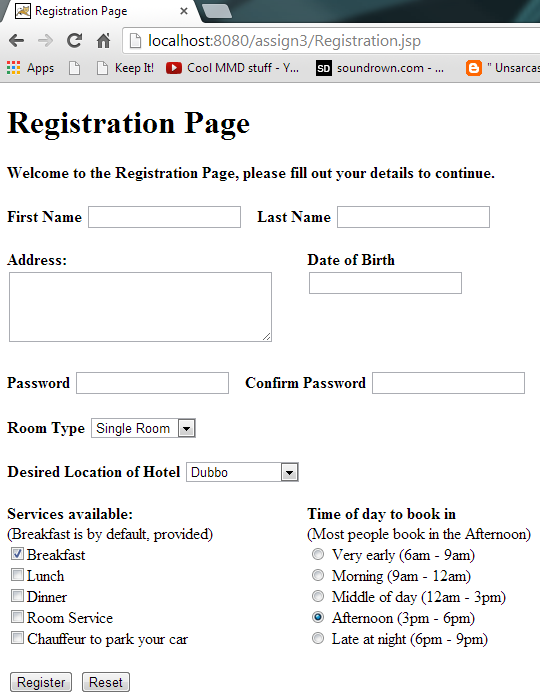
**Extra Notes:**

I didn’t read enough into the subject enough to find out if we should use the web.xml file of just “webservlet” annotations, which I used. However I have managed to get a servlet working with web.xml mapping and it not fun with name and mapping.

Program made and complied in :

* Windows 7 64bit
* IDE for Java EE Developers 2.0.0.20130613-0530
* apache-tomcat-7.0.42-windows-x86

To run project:

1. Open Eclipse IDE for Java EE Developers
2. Open the “file” menu
3. Click “Import”
4. Expand “General”
5. Double click “Existing projects into workspace”
6. Browse for the root directory (the one with .project files etc.)
7. Import
8. On the side, expand “assign3”
9. Expand “ web content”
10. Right click Registration.jsp
11. click “run as”
12. click “Run on server”
13. if prompted, browse to your “apache-tomcat-7.0.42-windows-x86” folder for the jar file requested.
14. Enjoy.

