COS30041 Creating Secure and Scalable Software [Java EE]

61 Credit Task 5.2 JSF Technology stack – User Input Validation in JSF

Time Frame: Weeks 5 – 8

Suggested to start in Week 5 and complete in Week 8
Submission Due: Week 9, Fri, 6:30pm

Overview

In this task, you are required to extend a Web application using related technologies. You are also required to demonstrate your work is of good quality.

Purpose	To demonstrate your ability to use relevant technologies to develop quality Web application
Tasks	1. Research into how to validate user inputs in a web page (using JSF's standard validator tag and validation method in ManagedBean) 2. Extend the Web application so that it can handle all CRUD operations 3. Prepare your test cases and test your application thoroughly by using appropriate input values and database contents 4. Describe (with justification) and document your design 5. Answer questions related to the design of the project
Pre-req Task[1]	Pass Task 5.1
Follow-up Task[2]	Pass Task 6.1
Suggested Time	1 – 2 hours if you know the stuff well 4 – 6 hours if you need to read the concepts and learn how to program the Web UI using Java EE
Resources	Lecture 05 Web UI Java EE – JSF + related technologies
Feedback	Ask your tutor for feedback
Next task	Pass Task 6.1

Credit Task 5.2 Submission Details and Assessment Criteria

You must create your own document (pdf) in **portrait** mode[3], which you will upload to Doubtfire, with the following details:

- Your name and student id
- Your tutor's name
- Your own responses to the tasks according to the corresponding instructions (see below)

[1]You need to complete the pre-requisite task before doing this task.
[2]You need to complete this task in order to do the follow-up task because the follow-up task depends on your answer in this one.
[3]Landscape mode pdf does not work properly in Doubtfire.

Background

Assumption: **You have completed all Pass Tasks up to and including Pass Task 5.1**

In this Portfolio task, we extend the capability of the "addUser.xhtml" page in Pass Task 5.1 so that it can validate the following input fields:

(a) userid – It must be exactly six characters long. If not, display an appropriate message (your choice) to alert the user.
(b) password and cPassword – Each of them **must be exactly 6 characters long**, have at least one Uppercase letter, one Lowercase letter, one digit [0 – 9] and one "+ | - | *".
(c) password and cPassword – The pair must be exactly the same. You may trim any spaces but the string comparison must be case sensitive.

Tasks and Instructions

Task 1. Analysis and Design Programming [Utilizing Validator in JSFs]
Now, there are at least two different ways of doing this in JSF technologies[4]:

(1) Using standard validator tag
(2) Using a user-defined validation method in a ManagedBean class

You need to read about and understand how to do these. You also need to choose wisely based on your needs. Justify your answers (Explaining why you choose, say, (2) for (a) and (1) for (b))

Task 2. Programming [Assume you have completed Task 1 above]
Modify your web page and the ManagedBean component to implement the "design" in Task 1 above

Task 3. Testing
Write your test cases (including the database content and input values) and test your work thoroughly via your Web client. Remember to collect your screen dump to show me the evidences.

Task 4. Document your learning journey / research journey
Write about your learning experience in this portfolio task.

[Optional, in case you want some challenges[5]]
Task 5. Answer the following question

5.1. Is it possible to mix and match these two validation methods in one input field? Show me your work to justify your answer.

Submission Task
Once completed, you need to submit a pdf file that contains all your work (e.g. selected code segments – show me the key stuff and some screen dumps of your testing)

Demonstration
You may be asked to demonstrate your assignment in the lab. You should be able to do this and explain your code when asked in the lab session.

[4] A third way is to develop a custom validator (aka program your own validator). But I won't encourage you to do that in such a short time.
[5] To be honest, I do not know the answer to this. I will try it when I have time, but probably not in this semester. However, my guess is YES. But, I may be wrong. If you can show me the "correct" answer, you may get some extra marks in your C results.]