COS30041 Creating Secure and Scalable Software [Java EE]

Pass Task 5.1 Web UI

Time Frame: Weeks 5 – 7

Suggested to start in Week 5 and complete by the end the Week of mid-semester break (Apr 8)

Submission Due: Week 7, Fri, 6:30pm

Overview

In this task, you are required to program 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. Learn to develop a Web application that requests the services provide by the business object (stateless session bean) developed in Pass Task 4.1 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 4.1
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

Pass Task 5.1 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.

Tasks and Instructions

Assumption: **You have completed all Pass Tasks up to and including Pass Task 4.1**

Task 1. Complete Lab_05_Web_App_Myuser

Task 2. Programming [Assume you have completed Task 1 above]
In this Lab task, we extend the Web application developed in Lab_05_Web_App_Myuser so that it allows a user to update **all**[4] their information except their primary key (and store those information in the database server). The business logic is defined as follows:

(1) a user first enters their userid to search their own record
(2) if the record can be found,
  (a) the information of the user should be displayed for the user to update their information
  (b) the user updates their information [Note: user cannot update their primary key!]
  (c) the user submits the information to update to the system
  (d) the user will be prompted whether the update is successful or not
  ~~**[Part of your Credit Task 5.3 - Research into how to use JavaMail API]**~~
  (e) Send an email to the user's email address[5] (even though it may have just been changed by the user) telling them that their information has been changed (and say something like "in case this is not done by you, please contact us immediately at xyz@swin.com")

Note: Swinburne's wifi blocks "sending email via any non-Swinburne email server". You have two options. First, if you want to use Swinburne's exchange server, you need to dig out the information for connecting to Swinburne exchange server and test this task. Second, you use your own personal hotspot to connect to your own non-Swinburne email server (e.g. Gmail). After the connection, you can then send email using your own non-Swinburne email server. For example, Gmail's settings are as follows:

```
mail.smtp.host = smtp.gmail.com
mail.smtp.port = 587
mail.smtp.auth = true
mail.smtp.starttls.enable = true
```

Note: See the code example in Appendix A on how to use JavaMail API.

  (f) otherwise, display an appropriate message and provide a link for user to go back to (1)

Note: To program this lab task, you need to use your ManagedBean component to send a request to the "updateRecord(MyuserDTO myuserDTO)" method in your Stateless Session Bean developed in your previous portfolio task "04_Pass_Task_4.1".

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. Design Justification

4.1. Draw an architecture diagram, using UML notations, of your entire set of programs. In your diagram, you need to annotate the names and types of each software component, put them in the right tiers, and indicate how they interact with each other.
4.2. Describe and discuss the role of each of the components in your diagram.

[4] Even though I say "all" information here, it is possible that the user only updates their phone number. The meaning of "all" here is that the user can change any information they want.
[5] Most web sites on Internet use user's email address as primary key. Ours is not. However, if we allow user to change their email addresses and want it to be "secure" by notifying the user about this change, the business logic for Pass task in 2(e) will be more complicated than what I want you to focus on in this portfolio task.

4.3. When designing your web pages, what decisions have been made by you (e.g. providing all information in one web page for user to key in the changes or providing only just one information per page)? What assumptions have been made in making such decisions? Explain and justify why you think this is a reasonable choice.

Task 5. Answer the following question [addressing some "security" concerns]
Assume that you now have concerns about your "alert" message in Task 2(e) [What if the account has been hacked and the email address was changed to the "wrong person"?] How can you be sure that your alert message is going to the right person? What precautions you are going to make to address this concern? The following is something for you to think about:

5.1. Which email address(es) should you use to alert the user about their information has just been changed? The old one? Or the new one? What procedures / steps would you use to make sure the "update" is genuine? Remember to justify your answer.

5.2. What are the things that you will change in your program to make this happen? In terms of your "company policies" [probably too broad from the "technical perspective" – which is the focus of our unit]? In terms of business logic? In terms of the database? Remember to justify your answer. [You do not need to write the "new" program, just "talk" about it. Present your idea in a clear and concise manner.]

Warning: To answer these questions well, you need to have some "deep" thinking, probably you need to make some assumptions and policy changes (business decisions). It is not a simple fix from the technology point of view. This may be a bit tricky as well. I do not expect a "perfect" answer here. I am expecting a reasonable one with not many unanswered issues. Probably I am asking you a question which I cannot answer it well. That is probably why most web sites did not border with this "email alert feature." Well they do by just sending an email to the "new" email address by saying "bad luck" if the account's been hacked. To me, this question is open end in some sense. So, don't spend too much time on it to have a "perfect" answer. A reasonable answer will be sufficient (of course, I do not want you to say – "send it to the new email address and bad luck …" – the one that I gave you here! ☺)

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). Remember to replace your "email credentials" by dummy data before submission.

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.

Using JavaMail API (via SMTP)

Qn: What do you need in sending an email?
Ans: A lot, the following is just the minimal information that you need.

Receipent's email address (aka "To" field)
Sender's email address (aka your email address)
Subject of the email (aka the subject line)
Body of the email (aka the text in your email)

Qn: What information do I need to authenticate with my own email server?
Ans: Depending on the actual protocol and your server. Usually, all these are done by your email client. May it be your Outlook, Mail, Thunderbird (Anyone using this email client?), or your Web Browser [after your login to your "web mail frontend" e.g. Outlook 365, GMail, Hotmail (still exist?)]

Here is the information from "GMail" using SMTP (Simple Mail Transfer Protocol?)

mail.smtp.host = smtp.gmail.com
mail.smtp.port = 587
mail.smtp.auth = true
mail.smtp.starttls.enable = true

Also, you need to temporarily "turn on" the "Access for less secure apps" in your GMail account when you test this "sending email feature". See https://support.google.com/accounts/answer/6010255?hl=en-GB for detailed instructions. Remember to "turn off" the "Access for less secure apps" after you have done your testing and collected your evidence for submission.

Also, remember to replace your "email address" and "password" with dummy data before putting your key code segments in the submission document. I do not want to know your email credentials.

Qn: How do I authenticate with my email server and send my email message out?
Ans: See the following code segments for sending email using Gmail's server via SMTP

Code segment for sending email (put in your "class") – need to change the information to suit your needs

```
String smtpServer = "smtp.gmail.com";
String from = "yyyy@gmail.com";
String to = "elau@swin.edu.au";
String subject = "Testing from gmail";
String body = "Hi Edmonds,\nThis is a test!\nRegards,\nEdmonds\n";
String emailUser = from;
String password = "XXXXXX";

try {
    Properties props = System.getProperties();

    // -- Attaching to default Session, or we could start a new one --
    props.put("mail.smtp.host", smtpServer);
    props.put("mail.smtp.port", 587);
    props.put("mail.smtp.auth", true);
    props.put("mail.smtp.starttls.enable", true);

    // -- prepare a password authenticator --
    MyAuthenticator myPA = new MyAuthenticator(emailUser, password); // see MyAuthenticator class

    // get a session
    Session session = Session.getInstance(props, myPA);

    // -- Create a new message --
    Message msg = new MimeMessage(session);
```

```
    // -- Set the FROM and TO fields --
    msg.setFrom(new InternetAddress(from));
    msg.setRecipients(Message.RecipientType.TO, InternetAddress.parse(to, false));

    // -- Set the subject and body text --
    msg.setSubject(subject);
    msg.setText(body);

    // -- Set some other header information --
    msg.setHeader("X-Mailer", "Gmail");
    msg.setSentDate(new Date());

    // -- Send the message --
    Transport.send(msg);
    Transport.send(msg, emailUser, password);

    System.out.println("Message sent OK.");
} catch (Exception ex) {
    ex.printStackTrace();
}
```

Code segment for the MyAuthenticator class

```
public class MyAuthenticator extends Authenticator {

    PasswordAuthentication mypa;

    public MyAuthenticator(String username, String password) {
        mypa = new PasswordAuthentication(username, password);
    }

    @Override
    public PasswordAuthentication getPasswordAuthentication() {
        return mypa;
    }

}
```

Remember to fix imports.

If you know how to use an anonymous inner class (some “Advanced” Java technique) to replace the code for “MyAuthenticator.java” class, you are welcome to use it.

Note: Glassfish server has all the support for JavaMail. So, deploying your app with these code to GlassFish server will work. However, if you do this on just NetBeans IDE or normal Java SDK via command line, you need to download “javax.mail.jar” (from Oracle – use the latest version) and include it as a “library” in your NetBeans project.