Answer 5.1 Pass task

COS30041 Creating Secure and Scalable Software (Swinburne University of Technology)

Task 2

editUser.xhtml:

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xh
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
    <h:head>
        <title>Edit a User Page</title>
    </h:head>
    <h:body>
        <h1>Edit a User</h1>
        <h2>
            <p>Please enter the user's new details below</p>
        </h2>
        <h3>
            <h:form>
                <h:panelGrid columns="2">
                    <h:outputText value="User Id: "/>
                    <h:inputText id="userid" value="#{myuserManagedBean.userid}"
                                 required="true"
                                 requiredMessage="The userid field cannot be empty!"
                                 size="6" />
                </h:panelGrid>
                <p></p>
                <h:commandButton id="submit" value="Find"
                                 action="#{myuserManagedBean.getUser}" />
            </h:form>
        </h3>
    </h:body>
</html>
```

editUserFailure.xhtml (wrong id):

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
    <h:head>
        <title>User Not Edited</title>
    </h:head>
    <h:body>
        <h:form>
            <h1>User Edited -Failure</h1>
            <h2>
                User whose user id is
                <h:outputText value="#{myuserManagedBean.userid}"/>
                cannot be edited in the system.
            </h2>
            <p></p>
            <h3>Possibly there is no existing user with the that userid.</h3>
            <p></p>
            Back to <h:commandButton value="Main Menu" action="mainmenu.xhtml"/>
        </h:form>
    </h:body>
</html>
```

editUserSuccess (good id):

```
<h:head>
    <title>User Edited</title>
</h:head>
<h:body>

        <h1>User Edited -Success</h1>
        <h2>
            User ID:
            <h:outputText value="#{myuserManagedBean.userid}"/>
        </h2>
        <h:form>
            <h:inputHidden id="userid"
                           pt:placeholder="#{myuserManagedBean.userid}"
                           value="#{myuserManagedBean.userid}"
                           required="true"
                           requiredMessage="The userid field cannot be empty!"
                           />
            <h:panelGrid columns="2">
                <h:outputText value="Name: "/>
                <h:inputText id="name"
                           pt:placeholder="#{myuserManagedBean.name}"
                           value="#{myuserManagedBean.name}"
                           required="true"
                           requiredMessage="The name field cannot be empty!"
                           size="30"/>
                <h:outputText value="Password "/>
                <h:inputText id="password"
                           pt:placeholder="#{myuserManagedBean.password}"
                           value="#{myuserManagedBean.password}"
                           required="true"
                           requiredMessage="The password field cannot be empty!
                           size="6"/>
                <h:outputText value="Email: "/>
                <h:inputText id="email"
                           pt:placeholder="#{myuserManagedBean.email}"
                           value="#{myuserManagedBean.email}"
                           required="true"
                           requiredMessage="The email field cannot be empty!"
                           size="30" />
                <h:outputText value="Telephone: "/>
```

```
                <h:inputText id="phone"
                           pt:placeholder="#{myuserManagedBean.phone}"
                           value="#{myuserManagedBean.phone}"
                           required="true"
                           requiredMessage="The telephone field cannot be empty!"
                           size="10" />
                <h:outputText value="Address: "/>
                <h:inputText id="address"
                           pt:placeholder="#{myuserManagedBean.address}"
                           value="#{myuserManagedBean.address}"
                           required="true"
                           requiredMessage="The email field cannot be empty!"
                           size="30" />
                <h:outputText value="Security Question: "/>
                <h:inputText id="secQn"
                           pt:placeholder="#{myuserManagedBean.secQn}"
                           value="#{myuserManagedBean.secQn}"
                           required="true"
                           requiredMessage="The security question field cannot be empty!"
                           size="60"/>
                <h:outputText value="Security Answer: "/>
                <h:inputText id="secAns"
                           pt:placeholder="#{myuserManagedBean.secAns}"
                           value="#{myuserManagedBean.secAns}"
                           required="true"
                           requiredMessage="The security answer field cannot be empty!"
                           size="60"/>
            </h:panelGrid>
            <p></p>
            <h:commandButton id="submit" value="Submit"
                           action="#{myuserManagedBean.updateUser}" />
        </h:form>
    </h:body>
</html>
```

editUserSuccessSuccess (no problem in editing user info):

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
    <h:head>
        <title>User Edited</title>
    </h:head>
    <h:body>
        <h:form>
            <h1>User Edited -Success</h1>
            <h2>
                User whose userid is
                <h:outputText value="#{myuserManagedBean.userid}"/>
                has been edited in the system.
            </h2>
            <p></p>
            <h3>
                Back to
                <h:commandButton value="Main Menu" action="mainmenu.xhtml"/>
                !
            </h3>
        </h:form>
    </h:body>
</html>
```

editUserSuccessFailure (there is problem in editing user info):

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
    <h:head>
        <title>User Not Edited</title>
    </h:head>
    <h:body>
        <h:form>
            <h1>User Edited -Failure</h1>
            <h2>
                User whose user id is
                <h:outputText value="#{myuserManagedBean.userid}"/>
                cannot be edited in the system.
            </h2>
            <p></p>
            <h3>Possibly there is no existing user with the that userid.</h3>
            <p></p>
            Back to <h:commandButton value="Main Menu" action="mainmenu.xhtml"/>
        </h:form>
    </h:body>
</html>
```

Add sendmail to MyuserFacade:

```
@Override
public void sendemail(String name, String email)
{
    String smtpServer = "smtp.gmail.com";
    String from = "s[redacted]@gmail.com";
    String to = email;
    String subject = "Testing from gmail";
    String body = "Hi "+name+" ,\nThis is a test!\nRegards,\nEdmonds\n";
    String emailUser = from;
    String password = "[redacted]";
    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();
```

Add class autheticator:

```
public class MyAuthenticator extends Authenticator {
    PasswordAuthentication mypa;
    public MyAuthenticator(String username, String password)
    {
        mypa = new PasswordAuthentication(username, password);
    }
    @Override
    public PasswordAuthentication getPasswordAuthentication()
    {
    return mypa;
    }
}
```

updateUser method:

```
public String updateUser() {
    String result = "failure";
    if (isValidUserid(userid) && isValidName(name) && isValidPassword(password) && isValidEmail(email)
            && isValidPhone(phone) && isValidAddress(address) && isValidSecQn(secQn) && isValidSecAns(secAns)) {
        MyuserDTO myuserDTO = new MyuserDTO(userid, name, password, email, phone, address, secQn, secAns);
        if (myuserFacade.updateRecord(myuserDTO)) {
            myuserFacade.sendemail(name, email);
            result = "success";
        }
    }
    return result;
}
```

Task 3

Database Current State:

#	USERID	NAME	PASSWORD	EMAIL	PHONE	ADDRESS	SECQN	SECANS
1	000001	Peter Smith	123456	psmith@swin.edu.au	9876543210	Swinburne EN510f	What is my name?	Peter
2	000002	James T. Kirk	234567	jkirk@swin.edu.au	8765432109	Swinburne EN511a	What is my name?	James
3	000003	Sheldon Cooper	345678	scooper@swin.edu.au	7654321098	Swinburne EN512a	What is my last name?	Cooper
4	000004	Clark Kent	456789	ckent@swin.edu.au	6543210987	SwinburneEN513a	What is my last name?	Kent
5	000007	David Lee	654321	dlee@swin.edu.au	0123456789	Swinburne EN510g	What is my name?	David

Test adduser with empty text field:

Add a User

Please enter the user's details below

User Id:	000005
Name:	Huy
Password	123
Confirm Password	123
Email:	13
Telephone:	
Address:	
Security Question:	
Security Answer:	

Submit

- The telephone field cannot be empty!
- The email field cannot be empty!
- The security question field cannot be empty!
- The security answer field cannot be empty!

Test adduser with existing id:

Add a User

Please enter the user's details below

User Id:	000001
Name:	huy
Password	123
Confirm Password	123
Email:	123
Telephone:	123
Address:	123
Security Question:	123
Security Answer:	213

Submit

User Added -Failure

User whose user id is 000001 cannot be added to the system.

Possibly there is an existing user with the same userid.

Back to Main Menu

Test adduser with non existind id:

Add a User

Please enter the user's details below

User Id:	000005
Name:	huy
Password	123
Confirm Password	123
Email:	123
Telephone:	123
Address:	123
Security Question:	123
Security Answer:	213

Submit

User Added -Success

User whose userid is 000005 has been added to the system.

Back to Main Menu **!**

#	USERID	NAME	PASSWORD	EMAIL	PHONE	ADDRESS	SECQN	SECANS
1	000001	Peter Smith	123456	psmith@swin.edu.au	9876543210	Swinburne EN510f	What is my name?	Peter
2	000002	James T. Kirk	234567	jkirk@swin.edu.au	8765432109	Swinburne EN511a	What is my name?	James
3	000003	Sheldon Cooper	345678	scooper@swin.edu.au	7654321098	Swinburne EN512a	What is my last name?	Cooper
4	000004	Clark Kent	456789	ckent@swin.edu.au	6543210987	SwinburneEN513a	What is my last name?	Kent
5	000007	David Lee	654321	dlee@swin.edu.au	0123456789	Swinburne EN510g	What is my name?	David
6	000005	huy	123	123	123	123	123	213

Test getUser with existing id

Display a User

Please enter the user's id below

User Id: 000005

Submit

Display User - Success

User ID: 000005

User Name: huy

User Password: 123

User Email: 123

User Address: 123

User Phone: 123

User Secret Question: 123

User Secret Answer: 213

Back to Main Menu **!**

Test getUser with non existing id

Display a User

Please enter the user's id below

User Id: 000006

Submit

Display User -Failure

User whose user id is 000006 cannot be founded.

Back to Main Menu

Test deleteUser with non existing id

Delete a User

Please enter the user's id below

User Id: 000006

Submit

Delete User -Failure

User whose user id is 000006 cannot be deleted from the system.

Possibly there is no existing user with the that userid.

Back to Main Menu

Test updateUser with existing id

Edit a User

Please enter the user's new details below

User Id: 000007

Find

User Edited -Success

User ID: 000007

Name:	David Lee
Password	654321
Email:	102641744@student.swin.edu.au
Telephone:	0123456789
Address:	Swinburne EN510g
Security Question:	What is my name?
Security Answer:	David

Submit

User Edited -Success

User ID: 000007

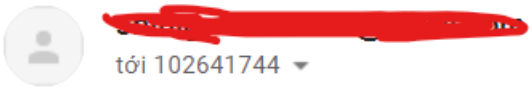
Name:	Peter Lee
Password	654321
Email:	102641744@student.swin.edu.au
Telephone:	0123456789
Address:	Swinburne EN510g
Security Question:	What is my name?
Security Answer:	David

Submit

User Edited -Success

User whose userid is 000007 has been edited in the system.

Back to Main Menu **!**



tới 102641744

Hi Peter Lee ,

•••

This is a test!
Regards,
Edmonds

Test updateUser with non existing id:

Edit a User

Please enter the user's new details below

User Id: 000008

Find

User Edited -Failure

User whose user id is 000008 cannot be edited in the system.

Possibly there is no existing user with the that userid.

Back to Main Menu

Test deleteUser with existing id

Delete a User

Please enter the user's id below

User Id: 000005

Submit

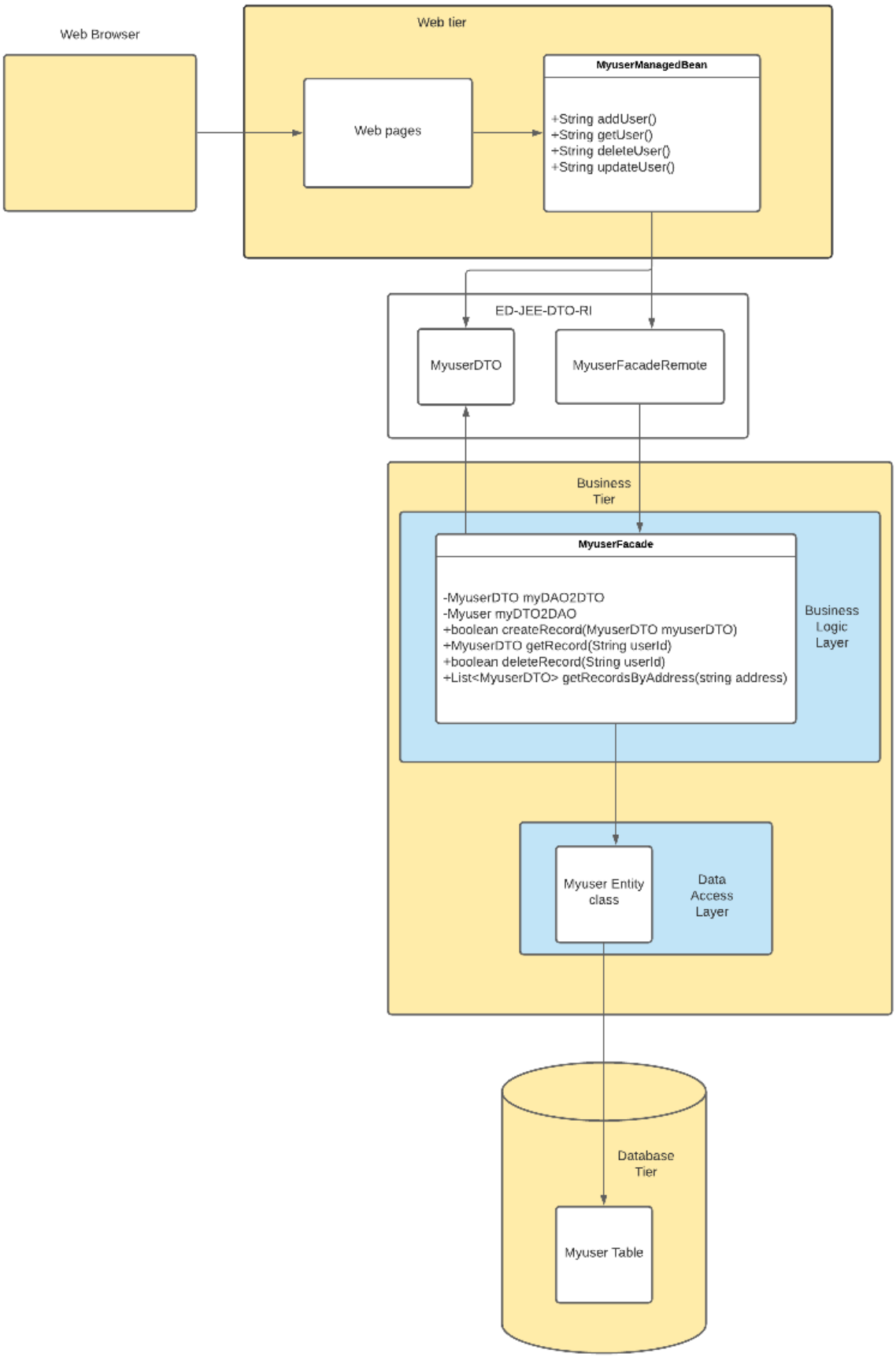
Delete User -Success

User whose userid is 000005 has been deleted from the system.

Back to Main Menu **!**

#	USERID	NAME	PASSWORD	EMAIL	PHONE	ADDRESS	SECQN	SECANS
1	000001	Peter Smith	123456	psmith@swin.edu.au	9876543210	Swinburne EN510f	What is my name?	Peter
2	000002	James T. Kirk	234567	jkirk@swin.edu.au	8765432109	Swinburne EN511a	What is my name?	James
3	000003	Sheldon Cooper	345678	scooper@swin.edu.au	7654321098	Swinburne EN512a	What is my last name?	Cooper
4	000004	Clark Kent	456789	ckent@swin.edu.au	6543210987	SwinburneEN513a	What is my last name?	Kent
5	000007	David Lee	654321	dlee@swin.edu.au	0123456789	Swinburne EN510g	What is my name?	David

Task 4



- Web pages is the main UI for the user
- MyuserManagedBean is a managed bean contains getters, setters, business logic and a backing bean to store data from the html form
- MyuserDTO is a DTO class use to securely transfer data from web tier to business tier
- MyuserFacadeRemote is a Remote Facade class used to combine several common methods in the "unfacaded" interface to reduce latency and network traffic.
- MyuserFacade is a Facade class that served as in interface which simplifies a complicated interface into a simpler one.
- Myuser is an Entity class which help we save and retrieve data without using SQL directly.

Task 5

5.1

If the user changed their info including email address, we should alert the old one.

If the user wanted to change their info, we should first confirm their password and secret question and answer. If email is changed send a confirmation email to their new email and send a email contain a link to change their email back to their old email, this will be active only for 72 hours.

This will ensure the person changing the email know the password, the new email is correct and a back-up link if there is any problem with the new email or it was not really the owner of the account who change the email.

5.2

Company Policies: Customer have a 3-day window to change back to their old email with only their email account.

Business Logic: When changing details, we will ask them for password and secret question. If email is changed, send a confirmation email to new email and a change back email to old email

Database: Add a last email column and a last date email changed column.

With this new policy, if a client's computer is compromised and they changed the email, the owner will have a reliable way to take the account back using email then using that email to change back password if needed.

For this to work, we needed to know the old email address and the date it was changed to know if the time window is over or not.