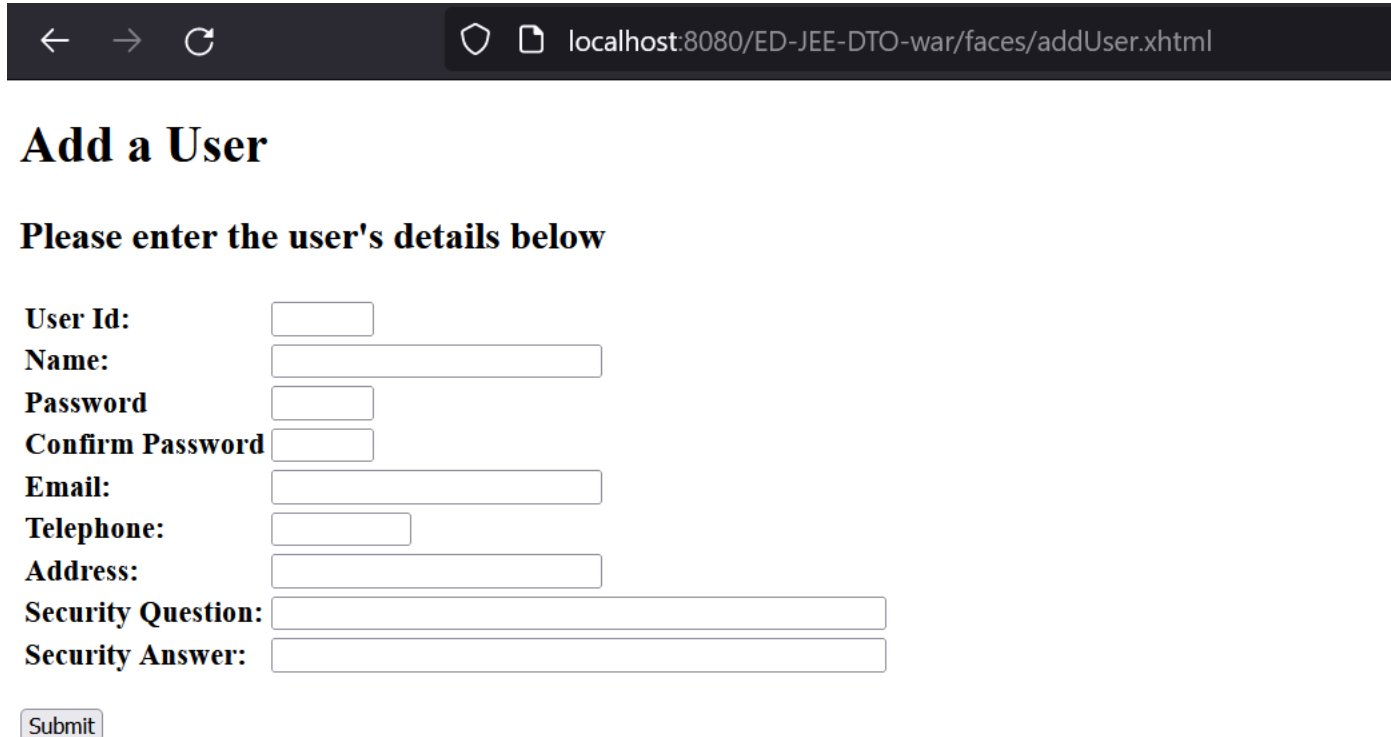


Name: S M Ragib Rezwan  
ID: 103172423  
Tutor: Wei Lai

Step 1 ending part:



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/ED-JEE-DTO-war/faces/addUser.xhtml'. The page title is 'Add a User'. Below the title, it says 'Please enter the user's details below'. The form contains the following fields:

- User Id:
- Name:
- Password:
- Confirm Password:
- Email:
- Telephone:
- Address:
- Security Question:
- Security Answer:

At the bottom left of the form is a 'Submit' button.



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/ED-JEE-DTO-war/faces/addUser.xhtml'. The page title is 'User Added - Failure'. Below the title, it says 'User whose userid is 000001 cannot be added to the system.' and 'Possibly there is an existing user with the same userid.' At the bottom left, there is a 'Back to Main Menu' button.

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai



## User Added - Success

User whose userid is 90 has been added to the system.

Back to [Main Menu](#) !

Step 2 all code parts:

Finduser webpage:

```
1 <title>Find User</title>
2 </h1>
3 <h2>Find User</h2>
4 <p>Please enter the user's ID below</p>
5 <h3>
6   <h:form>
7     <h:panelGrid columns="2">
8       <h:outputText value="User Id: " />
9       <h:inputText id="userid" value="#{myuserManagedBean.userid}"
10         required="true"
11         requiredMessage="The userid field cannot be empty!"
12         size="6" />
13     </h:panelGrid>
14     <p></p>
15     <h:commandButton id="submit" value="Find"
16       action="#{myuserManagedBean.findUser}" />
17   </h:form>
18 </h3>
19 </h:body>
20 </html>
```

Find user fail webpage:

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

```
Source History
<?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>Find User Fail</title>
  </h:head>
  <h:body>
    <h:form>
      <h1>
        User Found - Failure
      </h1>
      <h2>
        User whose userid is
        <h:outputText value="#{myuserManagedBean.userid}"/>
        cannot be Found.
      </h2>
      <p></p>
      <h3>
        Possibly there is no user with that userid.
      </h3>
      <p></p>
      Back to <h:commandButton value="Main Menu" action="mainmenu.xhtml"/>
    </h:form>
  </h:body>
</html>
```

Edit user webpage:

```
Source History
<?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>Edit User</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: #{myuserManagedBean.userid}"/>
          <h:inputHidden id="userid"
            value="#{myuserManagedBean.userid}"
            required="true"
            requiredMessage="The name field cannot be empty!"
          />
          <h:outputText value="Name: "/>
          <h:inputText id="name" value="#{myuserManagedBean.name}"
            required="true"
            requiredMessage="The name field cannot be empty!"
            size="30"/>
          <h:outputText value="Password "/>
          <h:inputText id="password" value="#{myuserManagedBean.password}"
            required="true"
            requiredMessage="The password field cannot be empty!"
            size="6"/>
          <h:outputText value="Confirm Password "/>
          <h:inputText id="cPassword" value="#{myuserManagedBean.cPassword}"
```

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

```
Source History
34 <h:outputText value="Confirm Password " />
35 <h:inputText id="cPassword" value="#{myuserManagedBean.cPassword}"
36         required="true"
37         requiredMessage="The confirm password field cannot be empty!"
38         size="6" />
39 <h:outputText value="Email: " />
40 <h:inputText id="email" value="#{myuserManagedBean.email}"
41         required="true"
42         requiredMessage="The email field cannot be empty!"
43         size="30" />
44 <h:outputText value="Telephone: " />
45 <h:inputText id="phone" value="#{myuserManagedBean.phone}"
46         required="true"
47         requiredMessage="The telephone field cannot be empty!"
48         size="10" />
49 <h:outputText value="Address: " />
50 <h:inputText id="address" value="#{myuserManagedBean.address}"
51         required="true"
52         requiredMessage="The email field cannot be empty!"
53         size="30" />
54 <h:outputText value="Security Question: " />
55 <h:inputText id="secQn" value="#{myuserManagedBean.secQn}"
56         required="true"
57         requiredMessage="The security question field cannot be empty!"
58         size="60" />
59 <h:outputText value="Security Answer: " />
60 <h:inputText id="secAns" value="#{myuserManagedBean.secAns}"
61         required="true"
62         requiredMessage="The security answer field cannot be empty!"
63         size="60" />
64 </h:panelGrid>
65 <p></p>
66 <h:commandButton id="submit" value="Submit"
67         action="#{myuserManagedBean.editUser}" />
68 </h:form>
```

```

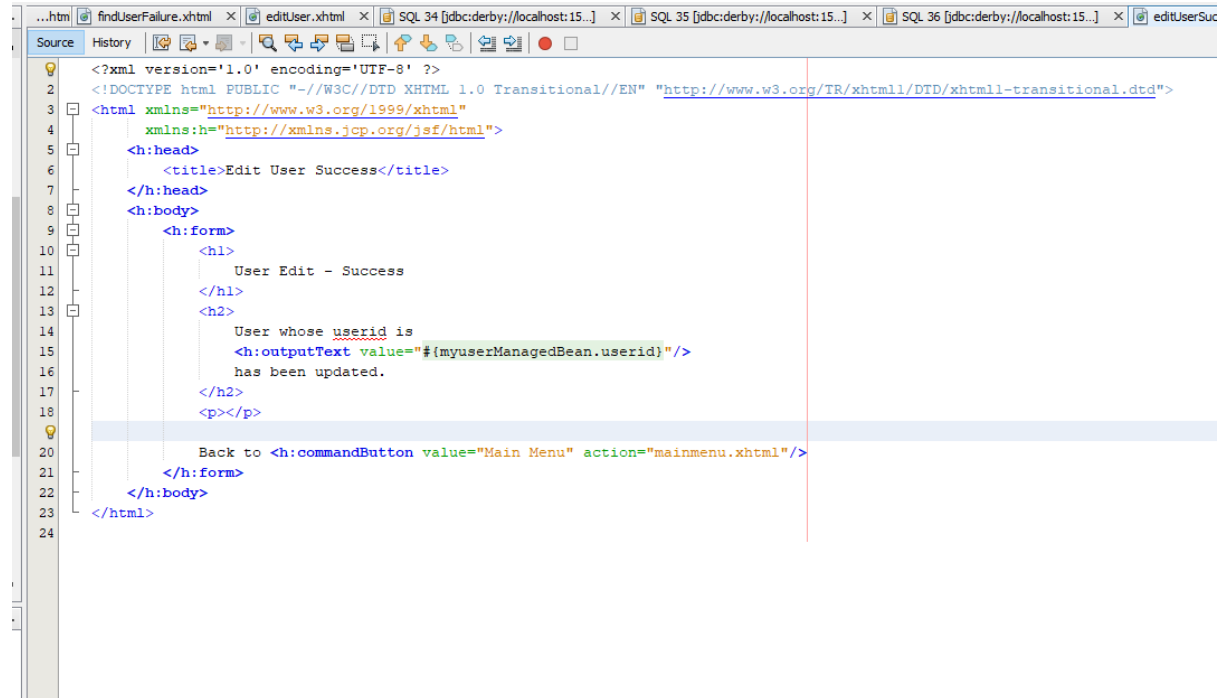
        required="true"
        requiredMessage="The security answer field cannot be empty!"
        size="60" />
</h:panelGrid>
<p></p>
<h:commandButton id="submit" value="Submit"
        action="#{myuserManagedBean.editUser}" />
</h:form>
</h3>
</h:body>
</html>
```

Edit user success webpage:

Name: S M Ragib Rezwan

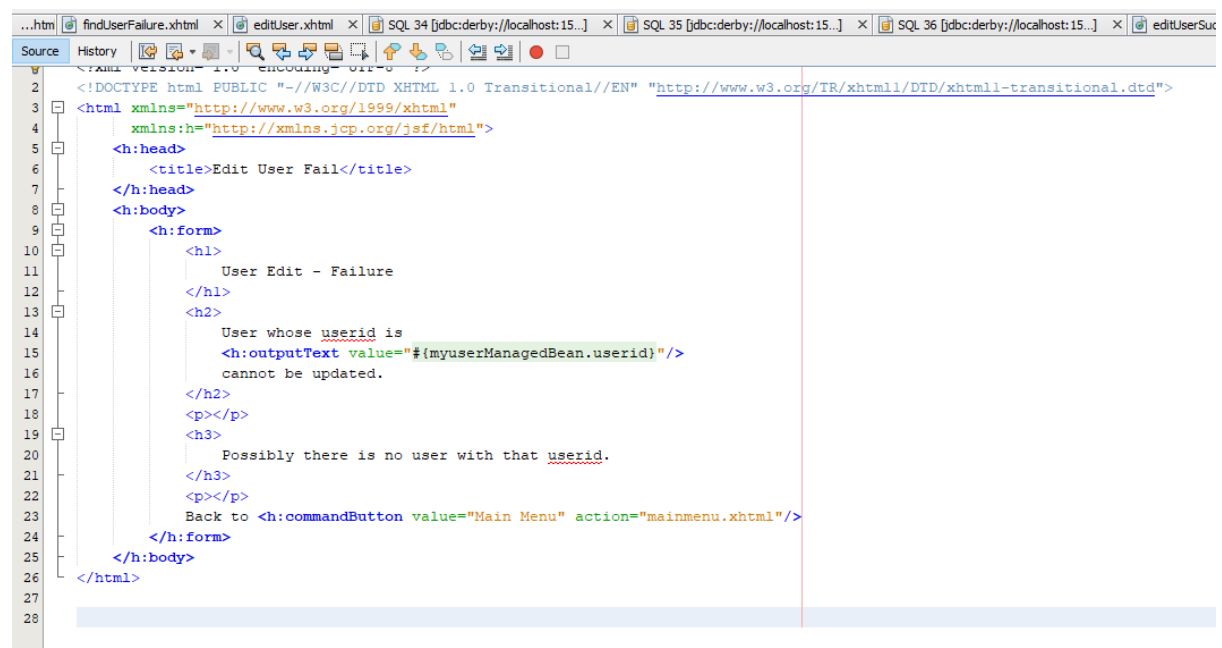
ID: 103172423

Tutor: Wei Lai



```
<?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>Edit User Success</title>
  </h:head>
  <h:body>
    <h:form>
      <h1>
        User Edit - Success
      </h1>
      <h2>
        User whose userid is
        <h:outputText value="#{myuserManagedBean.userid}"/>
        has been updated.
      </h2>
    </h:form>
    <p></p>
    Back to <h:commandButton value="Main Menu" action="mainmenu.xhtml"/>
  </h:body>
</html>
```

Edit user fail:



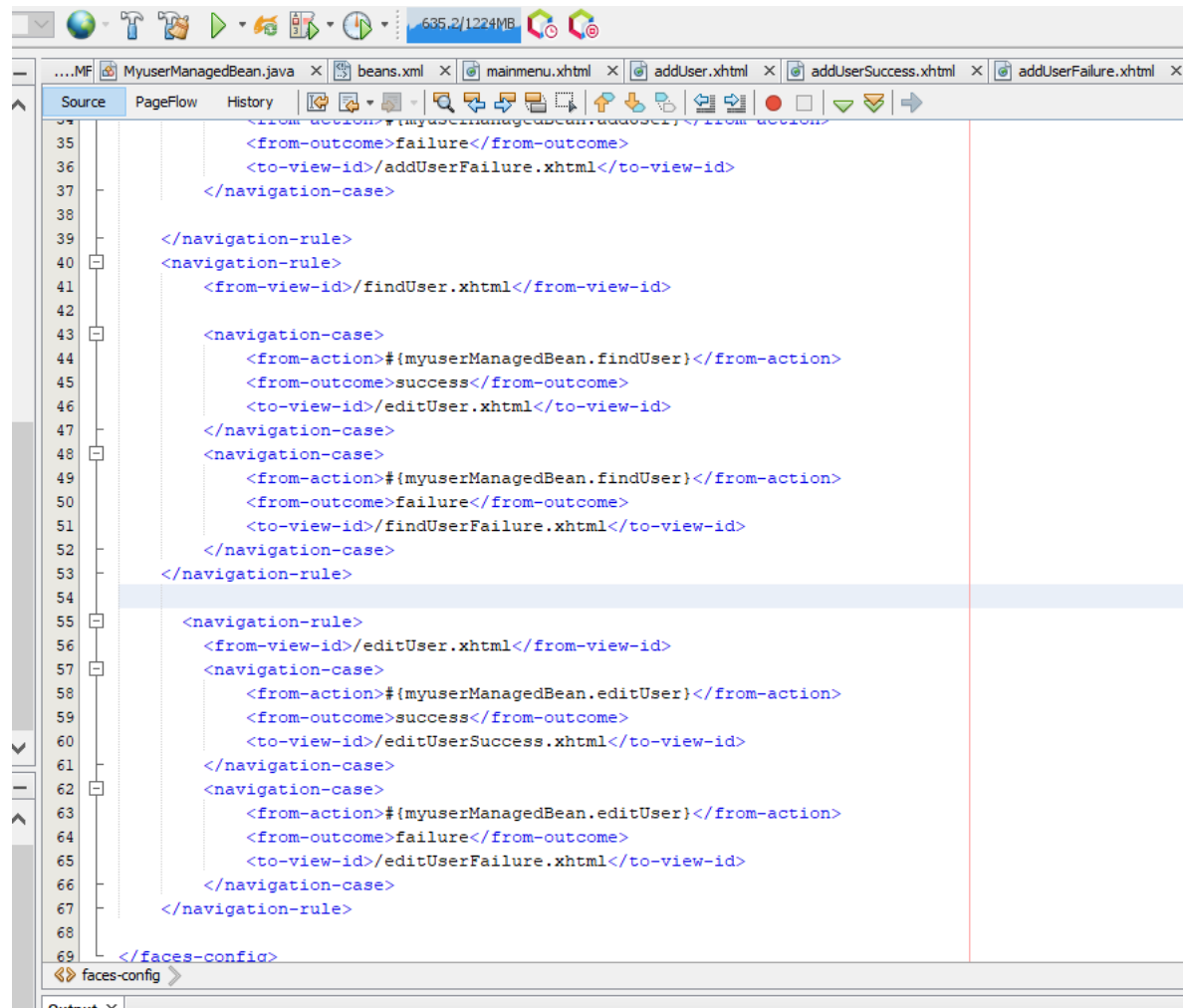
```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
  <h:head>
    <title>Edit User Fail</title>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

Navigation rules from face-config for find user and edit user all cases:

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai



```
35         <from-outcome>failure</from-outcome>
36         <to-view-id>/addUserFailure.xhtml</to-view-id>
37     </navigation-case>
38
39 </navigation-rule>
40 <navigation-rule>
41     <from-view-id>/findUser.xhtml</from-view-id>
42
43     <navigation-case>
44         <from-action>#{myuserManagedBean.findUser}</from-action>
45         <from-outcome>success</from-outcome>
46         <to-view-id>/editUser.xhtml</to-view-id>
47     </navigation-case>
48     <navigation-case>
49         <from-action>#{myuserManagedBean.findUser}</from-action>
50         <from-outcome>failure</from-outcome>
51         <to-view-id>/findUserFailure.xhtml</to-view-id>
52     </navigation-case>
53 </navigation-rule>
54
55 <navigation-rule>
56     <from-view-id>/editUser.xhtml</from-view-id>
57     <navigation-case>
58         <from-action>#{myuserManagedBean.editUser}</from-action>
59         <from-outcome>success</from-outcome>
60         <to-view-id>/editUserSuccess.xhtml</to-view-id>
61     </navigation-case>
62     <navigation-case>
63         <from-action>#{myuserManagedBean.editUser}</from-action>
64         <from-outcome>failure</from-outcome>
65         <to-view-id>/editUserFailure.xhtml</to-view-id>
66     </navigation-case>
67 </navigation-rule>
68
69 </faces-config>
```

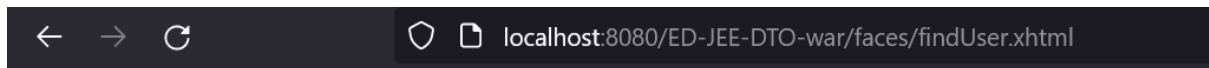
(email code is showing error, so kept it below at end of task 3 screenshots)

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

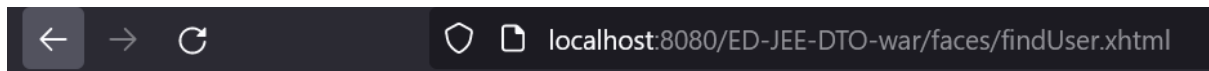
Outputs for find user, find user fail, edit user, edit user success and edit user fail:



## Find User

**Please enter the user's ID below**

User Id:

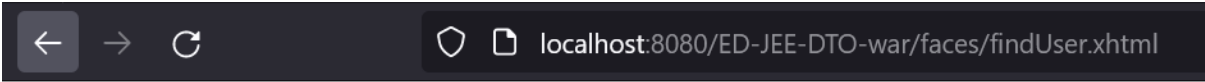


## User Found - Failure

**User whose userid is 1 cannot be Found.**

**Possibly there is no user with that userid.**

Back to



# Edit a User

Please enter the user's new details below

User Id: 000001

Name:

Password:

Confirm Password:

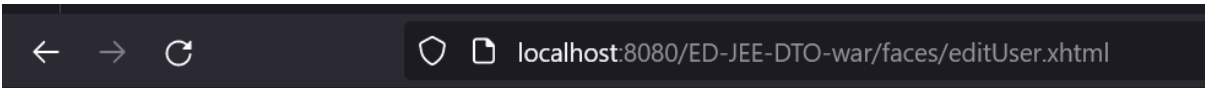
Email:

Telephone:

Address:

Security Question:

Security Answer:



# User Edit - Success

User whose userid is 000001 has been updated.

Back to

#	USERID	NAME	PASSWORD	EMAIL	PHONE	ADDRESS	SECQN	SECANS
1	000001	s	s	s	s	s	s	s
2	000002	James T. Kirk	234567	jrk@swin.edu.au	8765432109	Swinburne ENS11a	What is my name?	James
3	000003	Sheldon Cooper	345678	scooper@swin.edu.au	7654321098	Swinburne ENS12a	What is my last name?	... Cooper
4	000004	Clark Kent	456789	ckent@swin.edu.au	6543210987	Swinburne ENS13a	What is my last name?	... Kent
5	000005	Harry Potter	567890	hpotter@swin.edu.au	6543210987	Swinburne ENS14a	What is my last name?	... Potter



Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

(problem: after adding the email part, the edit user success is no longer working and instead an error shows up. Check that part with him before adding the codes for email and outputs)

Thus sending only the email codes for that one:

```
public boolean sendemail() {///boolean
    boolean result = false;
    String smtpServer = "smtp.gmail.com";
    String from = "bigarnewzar@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 Spassword = "RagibTester11366";
    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, Spassword); // 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, Spassword);
        System.out.println("Message sent OK.");

        result = true;
    } catch (Exception ex) {
        ex.printStackTrace();
    }
    return result;
}
```

```
Source History
1  /**
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package web;
7
8   import javax.mail.Authenticator;
9   import javax.mail.PasswordAuthentication;
10
11  /**
12   *
13   * @author User
14   */
15  public class MyAuthenticator extends Authenticator {
16
17      PasswordAuthentication mypa;
18
19      public MyAuthenticator(String username, String password) {
20          mypa = new PasswordAuthentication(username, password);
21      }
22
23      @Override
24      public PasswordAuthentication getPasswordAuthentication() {
25          return mypa;
26      }
27
28  }
29
```

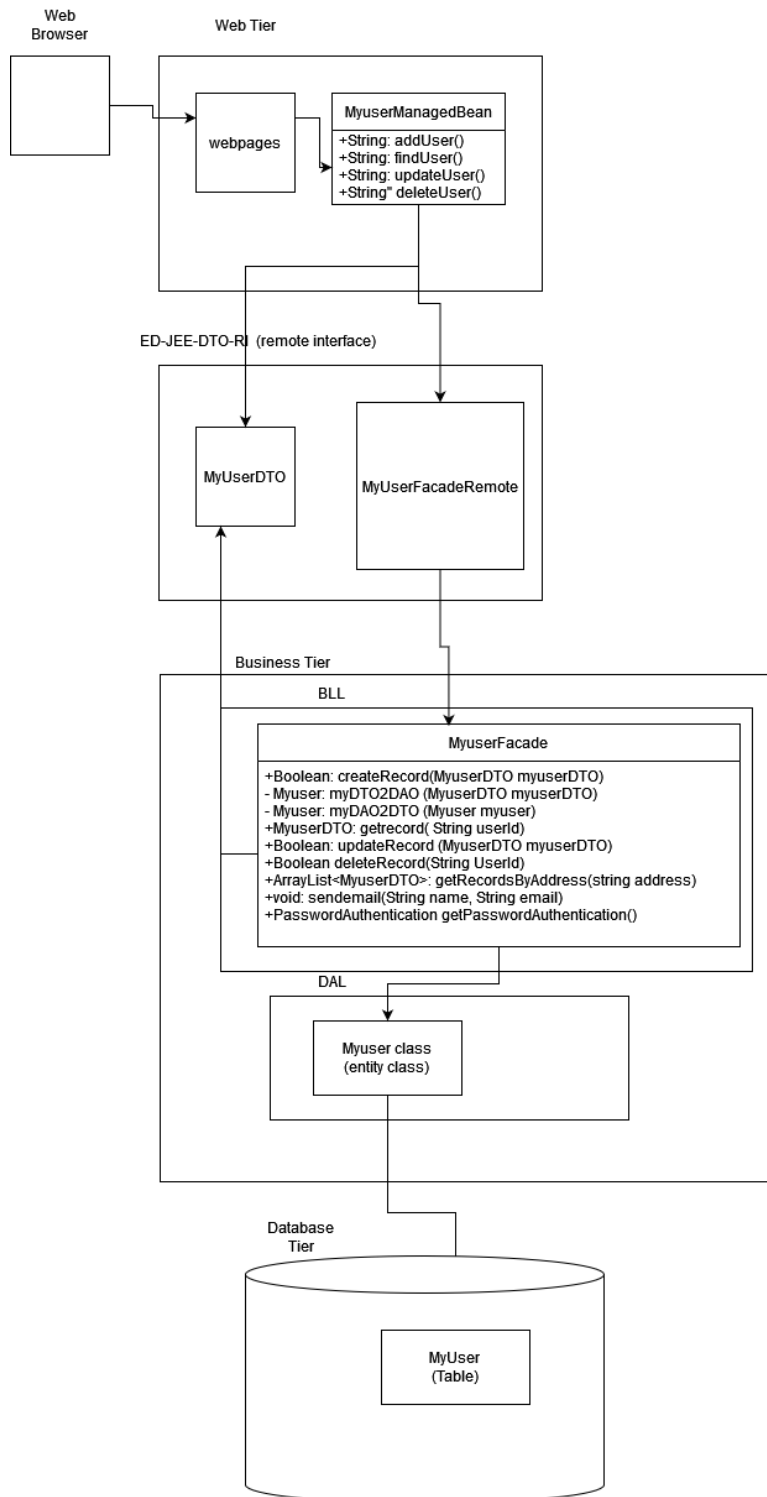
```

public String editUser() {
    String result = "failure";
    /* * if id valid then call facade's userid */
    if (isValidUserId(userid) && isValidName(name)
        && isValidPassword(password) && isValidPassword(cPassword)
        && isValidEmail(email) && isValidPhone(phone)
        && isValidAddress(address) && isValidSecQn(secQn)
        && isValidSecAns(secAns) && password.equals(cPassword)) {
        MyuserDTO myuserDTO = new MyuserDTO(userid, name,
            password, email, phone, address, secQn, secAns);
        if (myuserFacade.updateRecord(myuserDTO)) {
            this.sendemail();
            //had to shut it down for now. The rest part works, but adding this part cause the browser to show wierd errors. ask the tutor for help
            //also ask him if we have to do for delete and view records too although it is only saying to do edit user for P
            //and to just set regex protection for the add user part in C level task
            result = "success";
        }

        // this.sendemail();
    }
    /* {
        if (!sendemail()) {
            result = "failure";
            System.out.println("email cant be sent");
        } else {
            System.out.println("email can be sent");
        }
    } */
}
return result;
}

```

#### Task 4:



Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

Web pages is the UI (basically everything that the user sees)

- MyuserManagedBean is a managed bean. It contains all the getters, setters, and all the logic needed by the system stored in it
- MyuserDTO is a DTO class. It mainly transfers data from web UI to BLL
- MyuserFacadeRemote is a Remote Interface class. It is used to call the methods from the MyuserFacade to the MyuserManagedBean.
- MyuserFacade is a Facade class. It serves as an interface to simplify the codes and contains most of the basic logic being used inside the methods in MyuserManagedBean (except the very basic SQL ones)
- Myuser is an Entity class. It contains the most basic logic needed for us to transfer data to and from the database without using SQL directly.

## Part5

5.1) It would be better to alert the user's old email. But that time, we should at first prompt the user to reenter their password and sec question and answer. This ensures that the user is genuine and allow the update to take place. Also, as another safety precaution we can also provide a way for the user to change the email back to their old one via a link on their old email (this is the final precaution just in case the fraudulent user knows password and also sec question and answer)

5.2) Company policy wise, maybe give the users a certain time period (like a week or so) to revert their email back to old account using the provided link (that is sent to them when they update their email).

Business Logic wise, we will ask them for password, sec question and answer when they try to update their email and also send an email to their previous email asking whether the user is legit and providing a link to revert back to their old email.

Database wise, we can just keep everything same and simple and instead let business logic handle everything. But we can also set up a column for a new email alongside another column with time stamp and set a condition that if time stamp and current time difference exceeds a certain value (like a week for instance), then move the data in the new email to the old email column and make old email column null once more.