

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

Task1:

File Users (6)		
New....	Delete	
Select	User ID	Group List:
<input type="checkbox"/>	ed-none	ED-NONE
<input type="checkbox"/>	ed-guest	ED-APP-GUEST
<input type="checkbox"/>	ed-admin1	ED-APP-ADMIN
<input type="checkbox"/>	ed-admin2	ED-APP-ADMIN
<input type="checkbox"/>	ed-user1	ED-APP-USERS
<input type="checkbox"/>	ed-user2	ED-APP-USERS

Task2:

SECURE Company Ltd

Employee Management System

Login Page

Username

Password

SECURE Company Ltd

Employee Management System

Main Menu

1. [Add a new employee](#)
2. [Change an employee's details](#)
3. [Change an employee's password](#)
4. [Delete an employee](#)
5. [Display employee's details](#)

Click

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

SECURE Company Ltd

Employee Management System

Login Page

Username

Password

SECURE Company Ltd

Employee Management System

Authorization Failure Page

Sorry, you are not authorized to access the resources.

Please discuss this with your manager.

Please retry with another credentials

Task3:

(note: these are only the requirements we will use to see if our coding tasks are ok or not. So picture or ss here)

Task4:

4.1 CRUD operations mean Create, Read, Update and Delete records in database. The “creating” and “deleting” operations should only be done by the admin so as to prevent unauthorized people from creating or deleting records without admin approval and thus cause either mismatch in the stored data or even data loss. Thus it’s better to delegate this task to the admin and instead let the employees only read and update their own data.

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

4.2 – Employees should only be allowed to view their own data in details (except password part) and not anyone else's ones. That's because employees do not have the right to view their colleagues data as doing so would be a breach in privacy.

4.3 It is a good practice. That's because one's password is something that the client already knows by heart. So there is no need to show it. Furthermore, it also prevents the password from being stolen (like maybe "someone was looking over their shoulders while client was viewing their details and saw that info" type scenario can occur)

4.4.1 Other than the employeeID, everything else can be updated by the employee. That's because it is primary key that is used by the database to uniquely identify the employee. Although it is similar to password aspect, it is not the same as that one was done to protect their privacy and the employee is allowed to change his or her password.

4.4.2 employeeID should not be updated by employee. It can be done by making it as a "view only" value in the form present in the employee's update page (so as to not allow any input in the employeeID on that page).

.....

4.5 It should occur in the Employee Facade in the Business tier. This makes sure that the data can be updated in the employee table in the database at the same time.

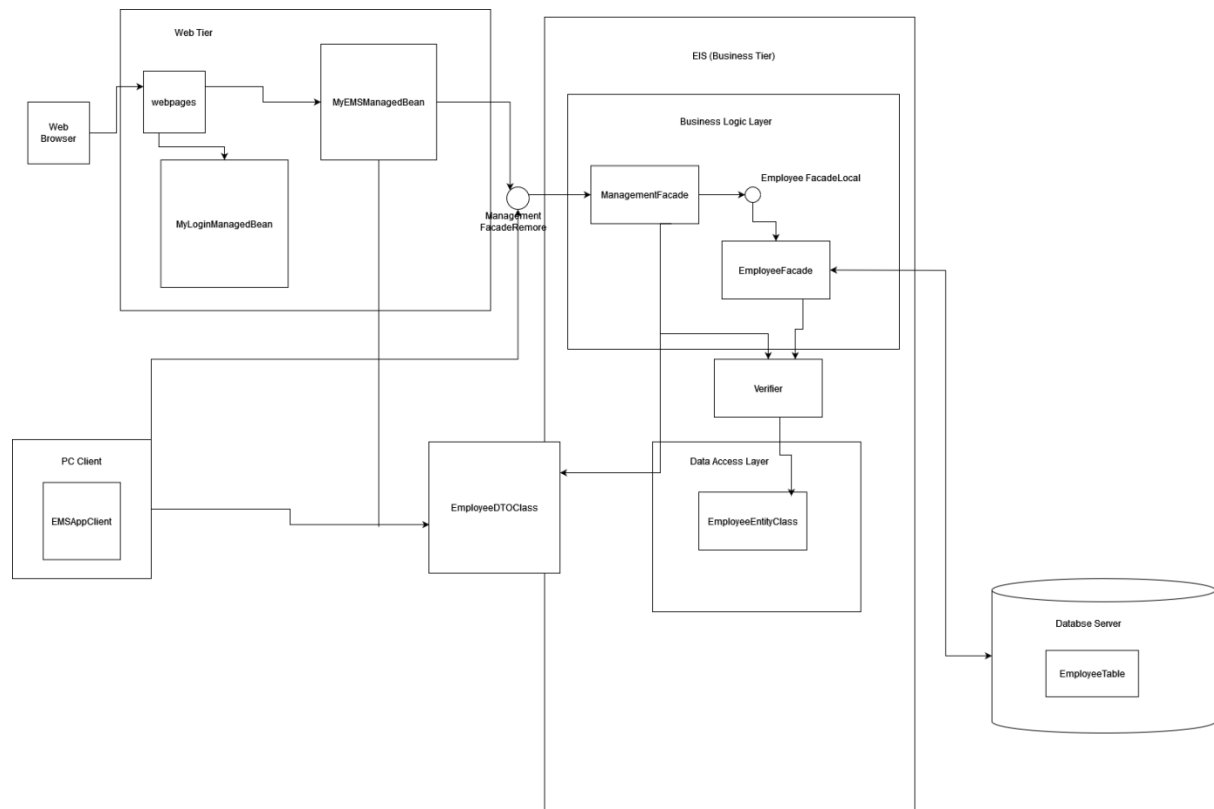
4.6 It will be better to not send it at all. That's because displaying password is a security risk and even if password isn't displayed, as long as it is sent over, there are ways to find that data. Employees should instead be taken to a different page where they would be requested to enter their old password first, before allowing them to change to a new one. This ensures that the client is actually who he says he is and has the right to make changes to his password.

4.7 It is a good practice in my eyes as doing that doesn't actually remove the record in the database and instead keeps an easy way to restore the employee data instead of recreating it from scratch. Thus even if an admin mistakenly deletes the record, he can still restore it back quickly. Furthermore, there may also be the case where the employee may return to the company in the future or the employee details are needed by the company later on (like for tax purposes and stuff). Thus the temporary action of making it "false" can save the company from going through a lot of hassle during those cases.

4.8 I believe the application can provide the features listed there. That's because it is giving admin the right to perform all CRUD operations while letting employees view (except password) and update their details (except their employeeID) whilst maintaining privacy and security. Furthermore, deleting doesn't really remove the record and instead makes it false, making it easy to restore the employee details if need be.

But I also believe that they can provide more security regarding the password. For instance they can set up a two factor authentication or so, or maybe set up an email feature where updating their password sends them an email regarding it. That way even if the password is stolen and the person tries to modify it, the issue can be fixed instantly by the employee before too much damage can occur.

Name: S M Ragib Rezwan
ID: 103172423
Tutor: Wei Lai
Task5:



The user will use his or her browser which will give him access to webpages in web tier. These in turn handle the communication between client and business tier (ass users with different roles login to see different webpages or do different actions). Then in the business tier, ManagementFacade transfers the data to and from the DTO and Employee façade (which in turn updates info in employee table in the Database). The data access layer contains the EmployeeEntityClass which contains the methods to call the data stored in the database and can only be accessed after passing through the verifier (using employeeID and password).

All Codes and Test case outputs for task 6:

Faces-config file

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

```

97         <from-outcome>success</from-outcome>
98         <to-view-id>/admin/deleteEmployeeSuccessful.xhtml</to-view-id>
99     </navigation-case>
100     <navigation-case>
101         <from-outcome>failure</from-outcome>
102         <to-view-id>/admin/deleteEmployeeFailure.xhtml</to-view-id>
103     </navigation-case>
104 </navigation-rule>
105
106
107 <navigation-rule>
108     <description>Employee's Main Menu</description>
109     <from-view-id>/user/mainmenu.xhtml</from-view-id>
110     <navigation-case>
111         <from-outcome>logout</from-outcome>
112         <to-view-id>/logout.xhtml</to-view-id>
113     </navigation-case>
114 </navigation-rule>
115 <navigation-rule>
116     <description>Display Employee</description>
117     <from-view-id>/user/displayEmployee.xhtml</from-view-id>
118     <navigation-case>
119         <from-action>#{myEmpManagedBean.EdisplayEmployee()}</from-action>
120         <from-outcome>success</from-outcome>
121         <to-view-id>/user/displayEmployeeDetails.xhtml</to-view-id>
122     </navigation-case>
123     <navigation-case>
124         <from-action>#{myEmpManagedBean.EdisplayEmployee()}</from-action>
125         <from-outcome>failure</from-outcome>
126         <to-view-id>/user/displayEmployeeFailure.xhtml</to-view-id>
127     </navigation-case>
128     <navigation-case>
129         <from-action>#{myEmpManagedBean.EdisplayEmployee()}</from-action>
130         <from-outcome>wrongpassword</from-outcome>
131         <to-view-id>/user/displayEmployeeWrongPassword.xhtml</to-view-id>
132     </navigation-case>
133     <navigation-case>
134         <from-outcome>debug</from-outcome>
135         <to-view-id>/user/debugEmpId.xhtml</to-view-id>
136     </navigation-case>
137 </navigation-rule>
138 <navigation-rule>

```

faces-config navigation-rule navigation-case

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

```
136         </navigation-case>
137     </navigation-rule>
138     <navigation-rule>
139         <description>Change employee</description>
140         <from-view-id>/user/changeEmployee.xhtml</from-view-id>
141         <navigation-case>
142             <from-action>#{myEmpManagedBean.EsetEmployeeDetailsForChange()}</from-action>
143             <from-outcome>success</from-outcome>
144             <to-view-id>/user/changeEmployeeDetails.xhtml</to-view-id>
145         </navigation-case>
146         <navigation-case>
147             <from-action>#{myEmpManagedBean.EsetEmployeeDetailsForChange()}</from-action>
148             <from-outcome>failure</from-outcome>
149             <to-view-id>/user/changeEmployeeFailure.xhtml</to-view-id>
150         </navigation-case>
151         <navigation-case>
152             <from-action>#{myEmpManagedBean.EsetEmployeeDetailsForChange()}</from-action>
153             <from-outcome>wrongpassword</from-outcome>
154             <to-view-id>/user/changeEmployeeWrongPassword.xhtml</to-view-id>
155         </navigation-case>
156         <navigation-case>
157             <from-outcome>debug</from-outcome>
158             <to-view-id>/user/debugEmpId.xhtml</to-view-id>
159         </navigation-case>
160     </navigation-rule>
161
162     <navigation-rule>
163         <description>Change details</description>
164         <from-view-id>/user/changeEmployeeDetails.xhtml</from-view-id>
165         <navigation-case>
166             <from-outcome>success</from-outcome>
167             <to-view-id>/user/changeEmployeeSuccessful.xhtml</to-view-id>
168         </navigation-case>
169         <navigation-case>
170             <from-outcome>failure</from-outcome>
171             <to-view-id>/user/changeEmployeeFailure.xhtml</to-view-id>
172         </navigation-case>
173         <navigation-case>
174             <from-outcome>debug</from-outcome>
175             <to-view-id>/user/debugEmpId.xhtml</to-view-id>
176         </navigation-case>
177     </navigation-rule>
178 </faces-config>
```

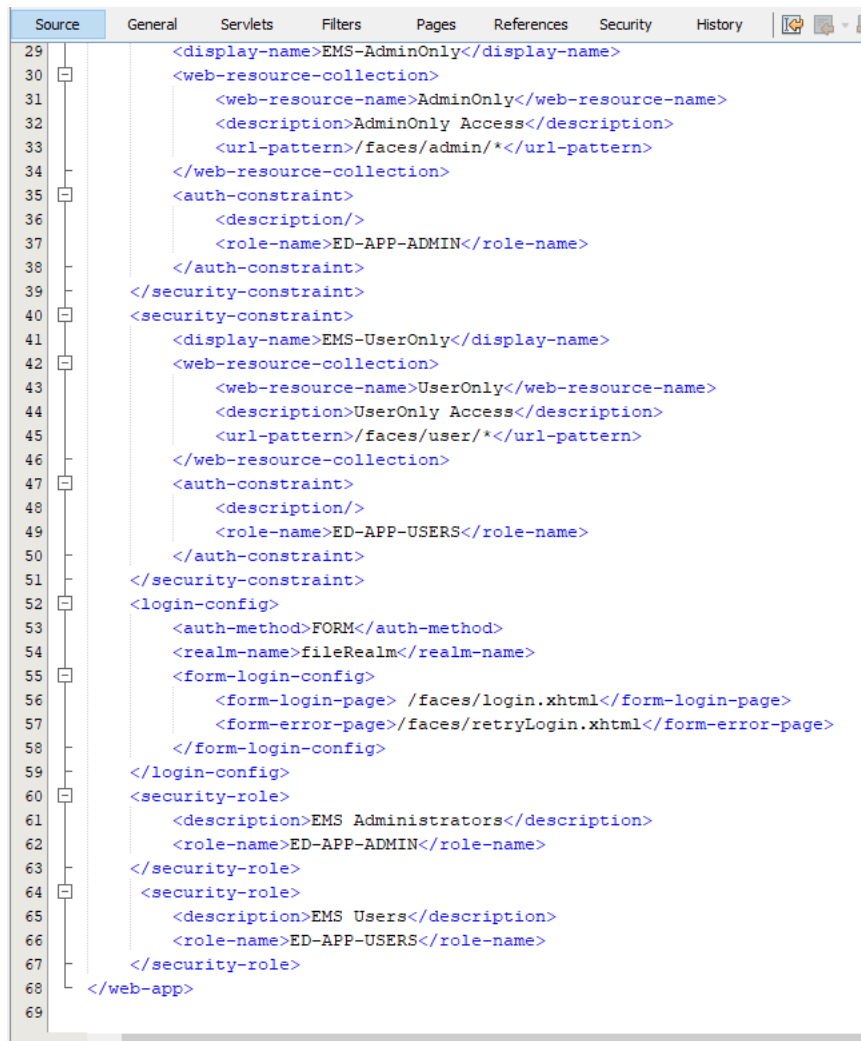
```
171         <to-view-id>/user/changeEmployeeFailure.xhtml</to-view-id>
172     </navigation-case>
173     <navigation-case>
174         <from-outcome>debug</from-outcome>
175         <to-view-id>/user/debugEmpId.xhtml</to-view-id>
176     </navigation-case>
177 </navigation-rule>
178
179     <navigation-rule>
180         <description>Change Password</description>
181         <from-view-id>/user/changeEmployeePassword.xhtml</from-view-id>
182         <navigation-case>
183             <from-action>#{myEmpManagedBean.EchangeEmployeePassword()}</from-action>
184             <from-outcome>success</from-outcome>
185             <to-view-id>/user/changePasswordSuccessful.xhtml</to-view-id>
186         </navigation-case>
187         <navigation-case>
188             <from-action>#{myEmpManagedBean.EchangeEmployeePassword()}</from-action>
189             <from-outcome>failure</from-outcome>
190             <to-view-id>/user/changePasswordFailure.xhtml</to-view-id>
191         </navigation-case>
192     </navigation-rule>
193
194
195
196
197 </faces-config>
198
199
```

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

Web file:



```
29 <display-name>EMS-AdminOnly</display-name>
30 <web-resource-collection>
31 <web-resource-name>AdminOnly</web-resource-name>
32 <description>AdminOnly Access</description>
33 <url-pattern>/faces/admin/*</url-pattern>
34 </web-resource-collection>
35 <auth-constraint>
36 <description/>
37 <role-name>ED-APP-ADMIN</role-name>
38 </auth-constraint>
39 </security-constraint>
40 <security-constraint>
41 <display-name>EMS-UserOnly</display-name>
42 <web-resource-collection>
43 <web-resource-name>UserOnly</web-resource-name>
44 <description>UserOnly Access</description>
45 <url-pattern>/faces/user/*</url-pattern>
46 </web-resource-collection>
47 <auth-constraint>
48 <description/>
49 <role-name>ED-APP-USERS</role-name>
50 </auth-constraint>
51 </security-constraint>
52 <login-config>
53 <auth-method>FORM</auth-method>
54 <realm-name>fileRealm</realm-name>
55 <form-login-config>
56 <form-login-page> /faces/login.xhtml</form-login-page>
57 <form-error-page>/faces/retryLogin.xhtml</form-error-page>
58 </form-login-config>
59 </login-config>
60 <security-role>
61 <description>EMS Administrators</description>
62 <role-name>ED-APP-ADMIN</role-name>
63 </security-role>
64 <security-role>
65 <description>EMS Users</description>
66 <role-name>ED-APP-USERS</role-name>
67 </security-role>
68 </web-app>
69
```

myEmpManagedBean file:

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

```
382
383 public String EsetEmployeeDetailsForChange() {
384     // check empId is null
385     if (isNull(empId) || conversation == null) {
386         return "debug";
387     }
388
389     if (!employeeManagement.hasEmployee(empId)) {
390         return "failure";
391     }
392
393     // note the startConversation of the conversation
394     startConversation();
395
396     // get employee details
397     return EsetEmployeeDetails();
398 }
399
400 public String EchangeEmployee() {
401     // check empId is null
402     if (isNull(empId)) {
403         return "debug";
404     }
405
406     EmployeeDTO empDTO = new EmployeeDTO(empId, name, phone,
407         address, email, password, appGroup, bnkAccId, salary, active);
408     boolean result = employeeManagement.updateEmpolyeeDetails(empDTO);
409
410     // note the endConversation of the conversation
411     endConversation();
412
413     //check if this part is reallt needed or not
414     EmployeeDTO e = employeeManagement.getEmployeeDetails(empId);
415     EmployeeDTO p = employeeManagement.getEmployeeDetails(password);
416
417     if (employeeManagement.checkUserIdPassword(empId, password) == true && result) {
418         return "success";
419     } else {
420         return "failure";
421     }
422 }
423
```

web.MyEmpManagedBean > EsetEmployeeDetailsForChange >

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

```
423
424 public String EchangeEmployeePassword() {
425     // check empId is null
426     if (isNull(empId)) {
427         return "debug";
428     }
429
430     if (employeeManagement.checkUserIdPassword(empId, password) == true) {
431
432         // newPassword and confirmPassword are the same - checked by the validator during input to JSF form
433         boolean result = employeeManagement.updateEmployeePassword(empId, newPassword);
434
435         System.out.println("result = " + result);
436
437         if (result) {
438             return "success";
439         } else {
440             return "failure";
441         }
442     } else {
443         System.out.println("something is wrong");
444         return "failure";
445     }
446 }
447
448 public String EdisplayEmployee() {
449     // check empId is null
450     if (isNull(empId) || conversation == null) {
451         return "debug";
452     }
453
454     return EsetEmployeeDetails();
455 }
456
457 private String EsetEmployeeDetails() {
458
459     if (isNull(empId) || conversation == null) {
460         return "debug";
461     }
462     System.out.println("testing before calling get emp details!");
463
464     EmployeeDTO e = employeeManagement.getEmployeeDetails(empId);
465     EmployeeDTO p = employeeManagement.getEmployeeDetails(password);
466 }
467
468 web.MyEmpManagedBean > EchangeEmployee > if (isNull(empId)) >
```

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

```
456
457 private String EsetEmployeeDetails() {
458
459     if (isNull(empId) || conversation == null) {
460         return "debug";
461     }
462     System.out.println("testing before calling get emp details!");
463
464     EmployeeDTO e = employeeManagement.getEmployeeDetails(empId);
465     EmployeeDTO p = employeeManagement.getEmployeeDetails(password);
466
467     System.out.println("testing if this can run!");
468
469     if (e == null) {
470         // no such employee
471         return "failure";
472     }
473     if (employeeManagement.checkUserIdPassword(empId, password)) {
474
475         System.out.println("testing if check password can run!");
476         // found - set details for display
477         this.empId = e.getEmpid();
478         this.name = e.getName();
479         this.phone = e.getPhone();
480         this.address = e.getAddress();
481         this.email = e.getEmail();
482         this.password = e.getPassword();
483         this.appGroup = e.getAppGroup();
484         this.bnkAccId = e.getBnkAccId();
485         this.salary = e.getSalary();
486         this.active = e.isActive();
487         return "success";
488
489     } else {
490         return "wrongpassword";
491     }
492 }
493
494 }
495
```

employeeManagementRemote

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

```
Source History | [Icons]
1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5   package session;
6
7   import javax.ejb.Remote;
8   import entity.EmployeeDTO;
9
10  @Remote
11  public interface EmployeeManagementRemote {
12
13      boolean hasEmployee(String empId);
14
15      boolean addEmployee(EmployeeDTO empDTO);
16
17      boolean updateEmployeeDetails(EmployeeDTO empDTO);
18
19      boolean updateEmployeePassword(String empId, String newPassword);
20
21      EmployeeDTO getEmployeeDetails(String empId);
22
23      boolean deleteEmployee(String empId);
24
25      boolean removeEmployee(String empId);
26
27      boolean checkUserIdPassword(String empId, String password);
28
29      EmployeeDTO getEmpDetails(String empId, String password);
30  }
31
```

employeeManagement:

```
Source History | [Icons]
3   * and open the template in the editor.
4   */
5   package session;
6
7   import javax.ejb.EJB;
8   import javax.ejb.Stateless;
9   import entity.Employee;
10  import entity.EmployeeDTO;
11  import javax.annotation.security.DeclareRoles;
12  import javax.annotation.security.RolesAllowed;
13
14  @DeclareRoles({"ED-APP-ADMIN", "ED-APP-USER"})
15  @Stateless
16  public class EmployeeManagement implements EmployeeManagementRemote {
17
18      @EJB
19      private EmployeeFacadeLocal employeeFacade;
20
21      private Employee employeeDTO2Entity(EmployeeDTO empDTO) {
```

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

```
73  */
74  @Override
75  @RolesAllowed({"ED-APP-ADMIN", "ED-APP-USERS"})
76  public boolean hasEmployee(String empId) {
77      return employeeFacade.hasEmployee(empId);
78  }
79
80  /**
81   * add an employee to the system
82   *
83   * @param empDTO
84   * @return true if addition is successful, false otherwise
85   */
86  @Override
87  @RolesAllowed({"ED-APP-ADMIN"})
88  public boolean addEmployee(EmployeeDTO empDTO) {
89
90      if (empDTO == null) {
91          // just in case
92          return false;
93      }
94
95      // check employee exist?
96      if (hasEmployee(empDTO.getEmpid())) {
97          // employee exists, cannot add one
98          return false;
99      }
100
101      // employee not exist
102      // convert to entity
103      Employee emp = this.employeeDTO2Entity(empDTO);
104      // add one
105      return employeeFacade.addEmployee(emp);
106  }
107
108  /**
109   * update employee details without updating password
110   *
111   * @param empDTO
112   * @return true if update is successful, false otherwise
113   */
114  @Override
115  @RolesAllowed({"ED-APP-ADMIN", "ED-APP-USERS"})
116  session.EmployeeManagement >
```

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai



```
114     @Override
115     @RolesAllowed({"ED-APP-ADMIN", "ED-APP-USERS"})
116     public boolean updateEmployeeDetails(EmployeeDTO empDTO) {
117         // check employee exist?
118         if (!hasEmployee(empDTO.getEmpid())) {
119             return false;
120         }
121
122         // employee exist, update details
123         // convert to entity class
124         Employee employee = this.employeeDTO2Entity(empDTO);
125         // update details
126         return employeeFacade.updateEmployeeDetails(employee);
127     }
128
129     /**
130     * update employee's password
131     *
132     * @param empId
133     * @param newPassword
134     * @return true if update successful, false otherwise
135     */
136     @Override
137     @RolesAllowed({"ED-APP-ADMIN", "ED-APP-USERS"})
138     public boolean updateEmployeePassword(String empId, String newPassword) {
139         return employeeFacade.updatePassword(empId, newPassword);
140     }
141
142     /**
143     * get employee details and use a DTO to transmit the details
144     *
145     * @param empId
146     * @return a DTO containing the information of the employee if exists, null
147     * otherwise
148     */
149     @Override
150     @RolesAllowed({"ED-APP-ADMIN", "ED-APP-USERS"})
151     public EmployeeDTO getEmployeeDetails(String empId) {
152         // get the employee
153         Employee employee = employeeFacade.find(empId);
154
155         if (employee == null) {
```

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

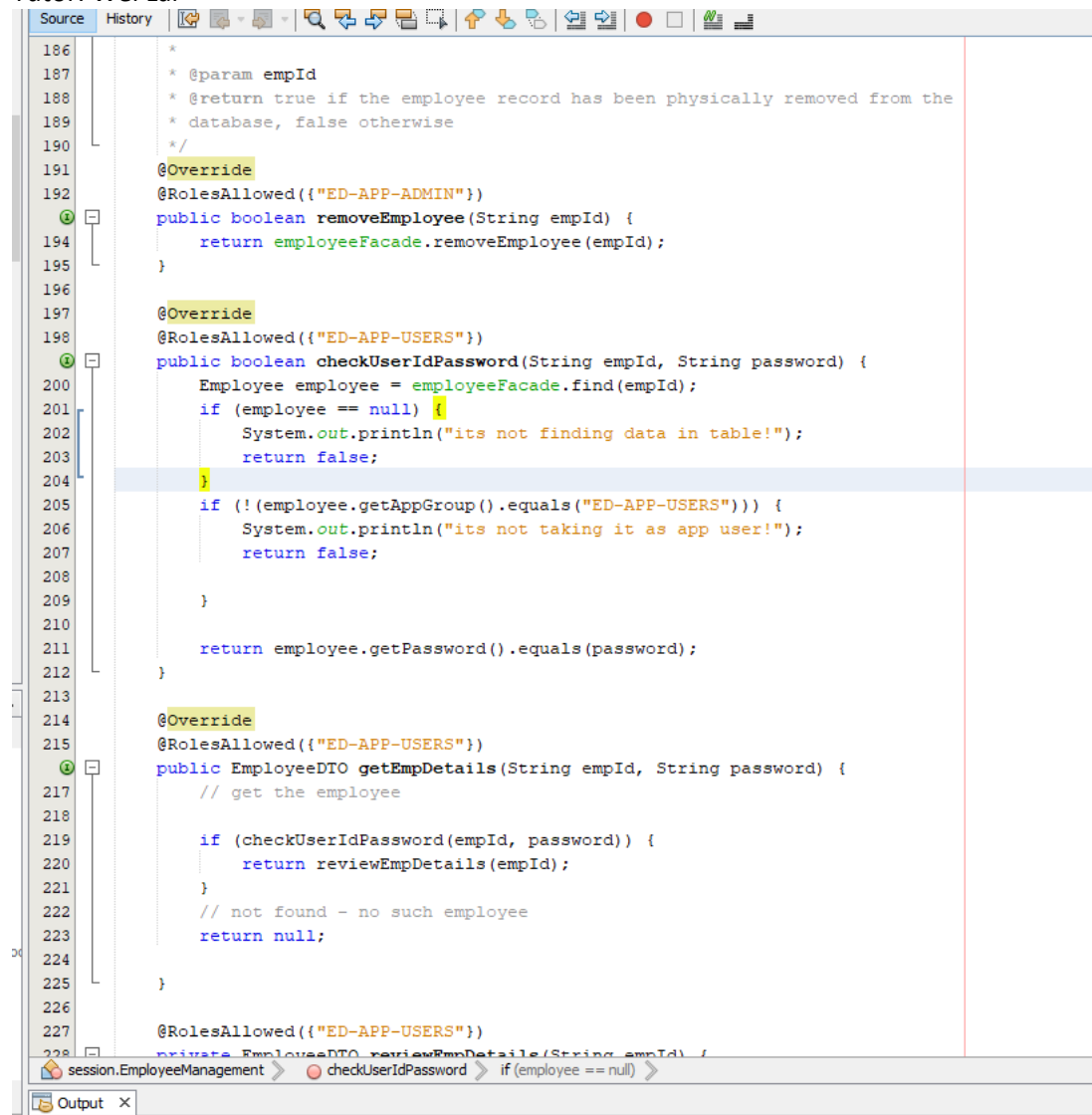
```
148  */
149  @Override
150  @RolesAllowed({"ED-APP-ADMIN","ED-APP-USERS"})
151  public EmployeeDTO getEmployeeDetails(String empId) {
152      // get the employee
153      Employee employee = employeeFacade.find(empId);
154
155      if (employee == null) {
156          // not found - no such employee
157          return null;
158      } else {
159          // found - prepare details
160          EmployeeDTO empDTO = new EmployeeDTO(employee.getEmpid(),
161              employee.getName(), employee.getPhone(), employee.getAddress(),
162              employee.getEmail(), employee.getPassword(), employee.getAppGroup(),
163              employee.getBnkAccId(), employee.getSalary(), employee.isActive());
164
165          return empDTO;
166      }
167  }
168
169  /**
170   * set the employee's active status to false
171   *
172   * @param empId
173   * @return true if it can be set to inactive and have set to inactive; false
174   * otherwise
175   */
176  @Override
177  @RolesAllowed({"ED-APP-ADMIN"})
178  public boolean deleteEmployee(String empId) {
179      return employeeFacade.deleteEmployee(empId);
180  }
181
182  /**
183   * physically remove an employee's record from database
184   *
185   * This is for lab purposes - never ever do this in real world applications
186   *
187   * @param empId
188   * @return true if the employee record has been physically removed from the
189   * database, false otherwise
190   */
191  public boolean deleteEmployeeRecord(String empId) {
192      // TODO: Implement this method
193      return false;
194  }
195  }
```

session.EmployeeManagement

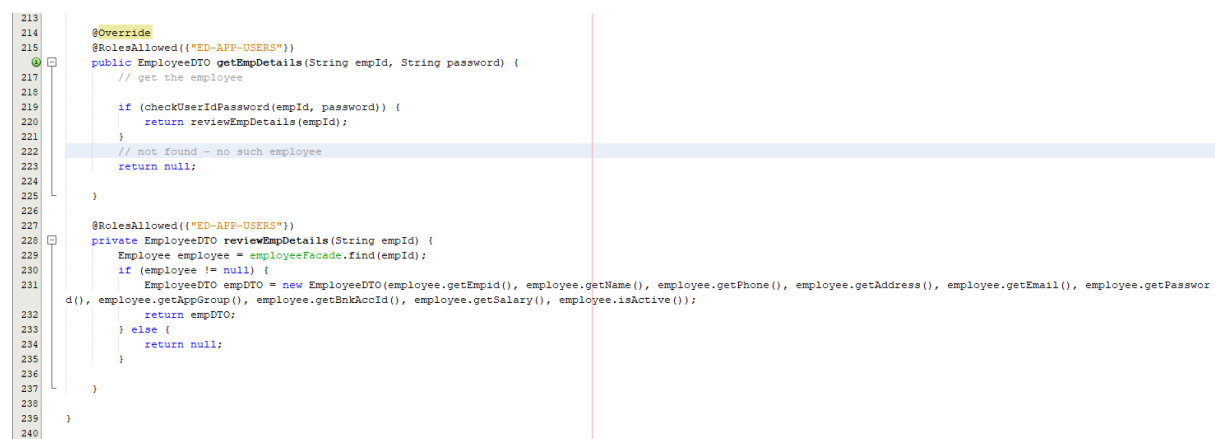
Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai



```
186 *
187 * @param empId
188 * @return true if the employee record has been physically removed from the
189 * database, false otherwise
190 */
191 @Override
192 @RolesAllowed({"ED-APP-ADMIN"})
193 public boolean removeEmployee(String empId) {
194     return employeeFacade.removeEmployee(empId);
195 }
196
197 @Override
198 @RolesAllowed({"ED-APP-USERS"})
199 public boolean checkUserIdPassword(String empId, String password) {
200     Employee employee = employeeFacade.find(empId);
201     if (employee == null) {
202         System.out.println("its not finding data in table!");
203         return false;
204     }
205     if (!(employee.getAppGroup().equals("ED-APP-USERS"))) {
206         System.out.println("its not taking it as app user!");
207         return false;
208     }
209
210     return employee.getPassword().equals(password);
211 }
212
213
214 @Override
215 @RolesAllowed({"ED-APP-USERS"})
216 public EmployeeDTO getEmpDetails(String empId, String password) {
217     // get the employee
218
219     if (checkUserIdPassword(empId, password)) {
220         return reviewEmpDetails(empId);
221     }
222     // not found - no such employee
223     return null;
224 }
225
226
227 @RolesAllowed({"ED-APP-USERS"})
228 private EmployeeDTO reviewEmpDetails(String empId) {
229     Employee employee = employeeFacade.find(empId);
230     if (employee != null) {
231         EmployeeDTO empDTO = new EmployeeDTO(employee.getEmpId(), employee.getName(), employee.getPhone(), employee.getAddress(), employee.getEmail(), employee.getPassword(), employee.getAppGroup(), employee.getBnkAccId(), employee.getSalary(), employee.isActive());
232         return empDTO;
233     } else {
234         return null;
235     }
236 }
237
238 }
239
240 }
```



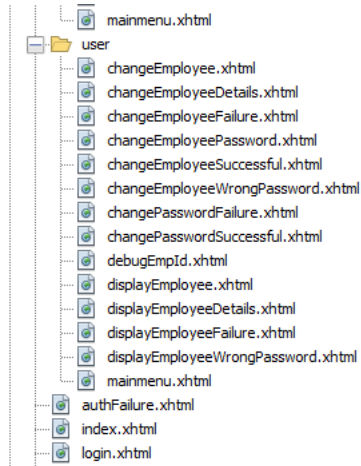
```
213
214
215 @Override
216 @RolesAllowed({"ED-APP-USERS"})
217 public EmployeeDTO getEmpDetails(String empId, String password) {
218     // get the employee
219
220     if (checkUserIdPassword(empId, password)) {
221         return reviewEmpDetails(empId);
222     }
223     // not found - no such employee
224     return null;
225 }
226
227 @RolesAllowed({"ED-APP-USERS"})
228 private EmployeeDTO reviewEmpDetails(String empId) {
229     Employee employee = employeeFacade.find(empId);
230     if (employee != null) {
231         EmployeeDTO empDTO = new EmployeeDTO(employee.getEmpId(), employee.getName(), employee.getPhone(), employee.getAddress(), employee.getEmail(), employee.getPassword(), employee.getAppGroup(), employee.getBnkAccId(), employee.getSalary(), employee.isActive());
232         return empDTO;
233     } else {
234         return null;
235     }
236 }
237
238 }
239
240 }
```

List of webpages added:

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai



Outputs:

Index page:

Secure Company Ltd Home Page

Welcome to our company

We aim at developing secure enterprise application solutions to corporations.

[Employee Management System \(ADMIN ONLY\)](#)

[Employee Management System \(FOR NORMAL USER\)](#)

When admin tries to access:

SECURE Company Ltd

Employee Management System

Authorization Failure Page

Sorry, you are not authorized to access the resources.

Please discuss this with your manager.

Please retry with another credentials

Name: S M Ragib Rezwan
ID: 103172423
Tutor: Wei Lai
When eduser1 accesses it:

SECURE Company Ltd

Employee Management System

Main Menu

1. [Change an employee's details](#)
2. [Change an employee's password](#)
3. [Display employee's details](#)

Click

All cases for changing employee password:

Change Employee's Password Page

Please enter the following information to change the password

Employee Id:	<input type="text" value="00004"/>
Password:	<input type="text" value="testwrong"/>
New Password:	<input type="password" value="••••••••"/>
Confirm New Password:	<input type="password" value="••••••••"/>

Cannot change the password of employee whose id is 00004 due to incorrect the employee id and/or password.

Click [here](#) to try again

Click [here](#) to return to the Main Menu

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

Change Employee's Password Page

Please enter the following information to change the password

Employee Id:	<input type="text" value="00004"/>
Password:	<input type="text" value="qqqqqqqq"/>
New Password:	<input type="text" value="••••••••"/>
Confirm New Password:	<input type="text" value="••••••••"/>

The password of the employee whose id is 00004 has been updated in the system.

Click [here](#) to return to the Main Menu

#	EMPID	NAME	PHONE	ADDRESS	EMAIL	PASSWORD	APPGROUP	BANKACCOUNTID	SALARY	ACTIVE
1	00001	Adam	1234567890	1 John Street, Hawthorn	adam@secure.com.au	11111111	ED-APP-ADMIN	098-765432-1	50000.00	✓
2	00002	Bill	2345678901	2 Paul Street, Hawthorn	bill@secure.com.au	22222222	ED-APP-ADMIN	109-876543-2	65000.00	✓
3	00003	Ceci	3456789012	3 Mary Street, Hawthorn	ceci@secure.com.au	33333333	ED-APP-USERS	210-987654-3	75000.00	✓
4	00004	Dave	4567890123	4 Pete Street, Hawthorn	dave@secure.com.au	12345678	ED-APP-USERS	321-098765-4	100000.00	✓
5	00009	Issac	9876543210	9 Newton Street, Hawthorn	issac@secure.com.au	99999999	ED-APP-USERS	999-765432-1	20000.00	✓

All cases for changing employee detail:

Change Employee Details Page

Please Login with your employee id and password

Employee Id:	<input type="text" value="00004"/>
Password:	<input type="text" value="wrongpas"/> ×

Your password for employee id 00004 is incorrect !

Please ensure that the employee id and password is correct

Click [here](#) to try again

Click [here](#) to return to the Main Menu

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

Change Employee Details Page

Please Login with your employee id and password

Employee Id:

Password:

Change Employee Details Page

Please update the details of employee 00004

Employee Id: 00004

Name:

Phone:

Address:

Email:

User Group: ▼

Bank Account No

Salary:

Active: ☒

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

Change Employee Details Page

Please update the details of employee 00004

Employee Id:	00004
Name:	<input type="text" value="DaveChanged"/>
Phone:	<input type="text" value="4567890123"/>
Address:	<input type="text" value="4 Pete Street, Hawthorn"/>
Email:	<input type="text" value="dave@secure.com.au"/>
User Group:	<input type="text" value="ED-APP-USERS"/>
Bank Account No	<input type="text" value="321-098765-4"/>
Salary:	<input type="text" value="100000.0"/>
Active:	<input checked="" type="checkbox"/>

The details of the employee whose id is 00004 has been updated in the system.

Click [here](#) to return to the Main Menu

All cases for seeing employee detail:

Search for an employee

Please enter the employee id

Employee Id:	<input type="text" value="00004"/>
Password:	<input type="password" value="wrongpas"/>

Your password for employee id 00004 is incorrect !

Please ensure that the employee id and password is correct

Click [here](#) to try again

Click [here](#) to return to the Main Menu

Name: S M Ragib Rezwan

ID: 103172423

Tutor: Wei Lai

Search for an employee

Please enter the employee id

Employee Id:

Password: ×

Details of Employee 00004

Name: DaveChanged
Phone: 4567890123
Address: 4 Pete Street, Hawthorn
Email: dave@secure.com.au
User Group: ED-APP-USERS
Bank Account No: 321-098765-4
Salary: 100000.0
Active: true

Click [here](#) to return to the Main Menu