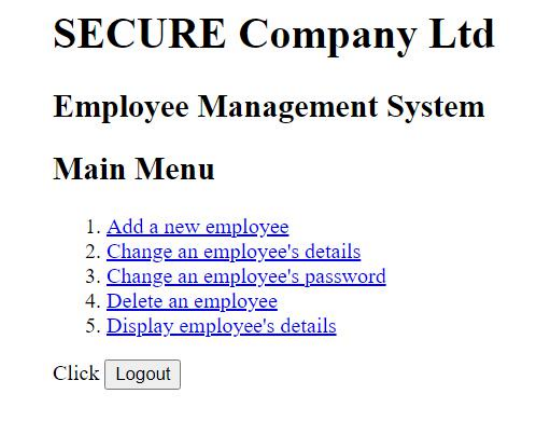
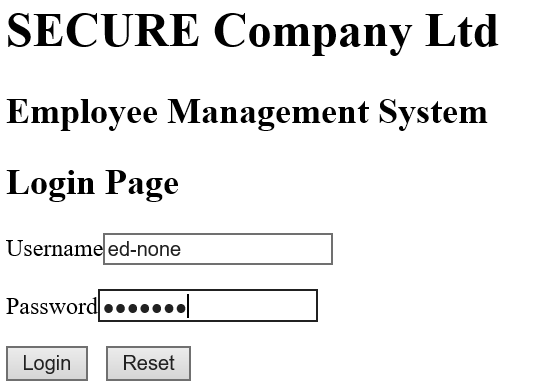
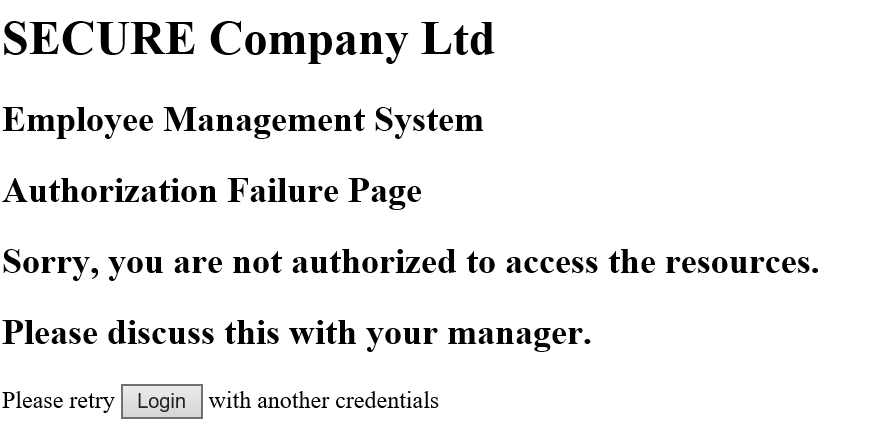
Task1:



Task2:







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.

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 updated in the employee table in the database at the same time.

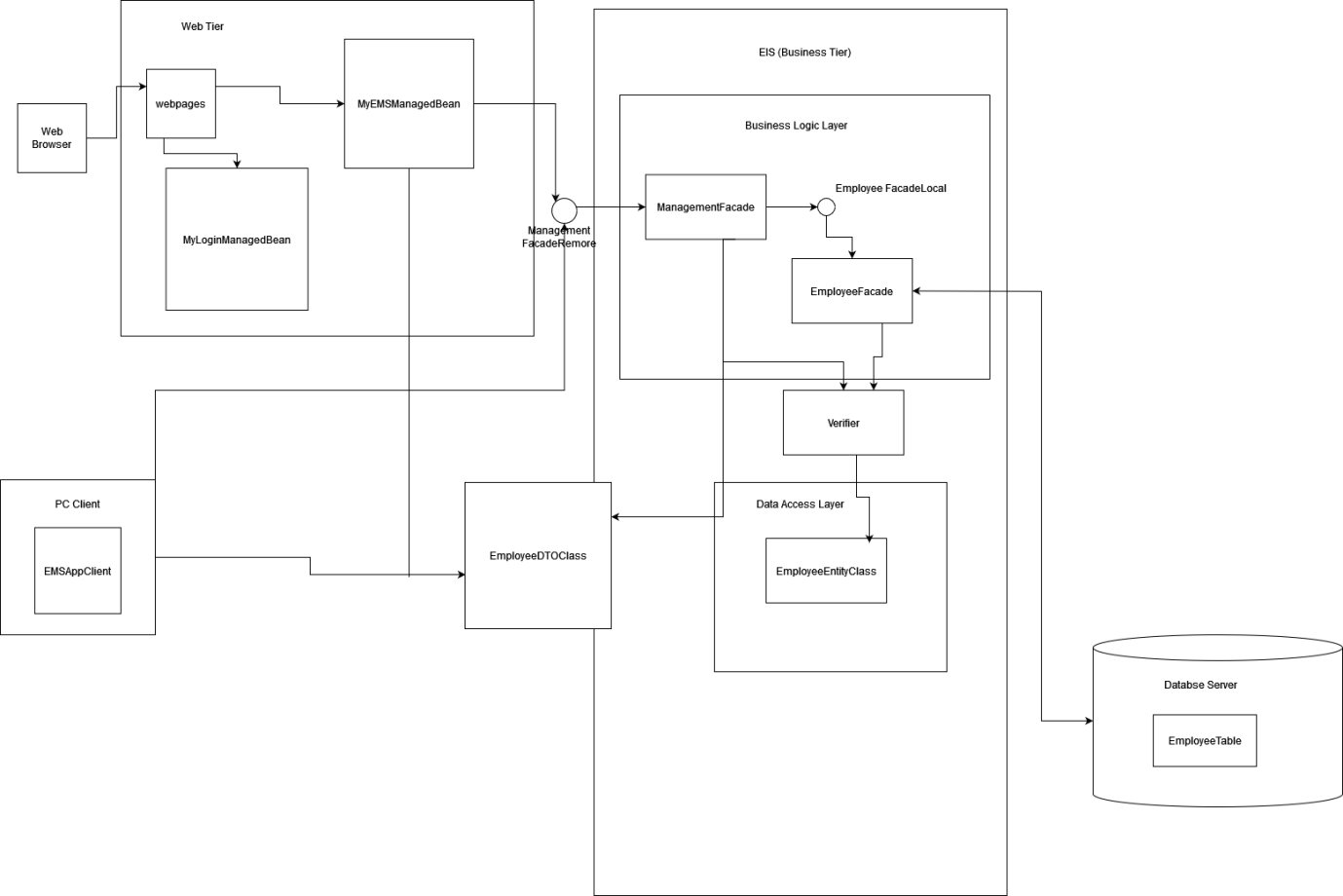
4.6 It will be better to not sent it at all. That’s because displaying password is a security risk and even if password isn’t displayed, as long as it is sent over, there are ways to find that data. Employees should instead be taken to a different page where they would be requested to enter their old 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 the 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.

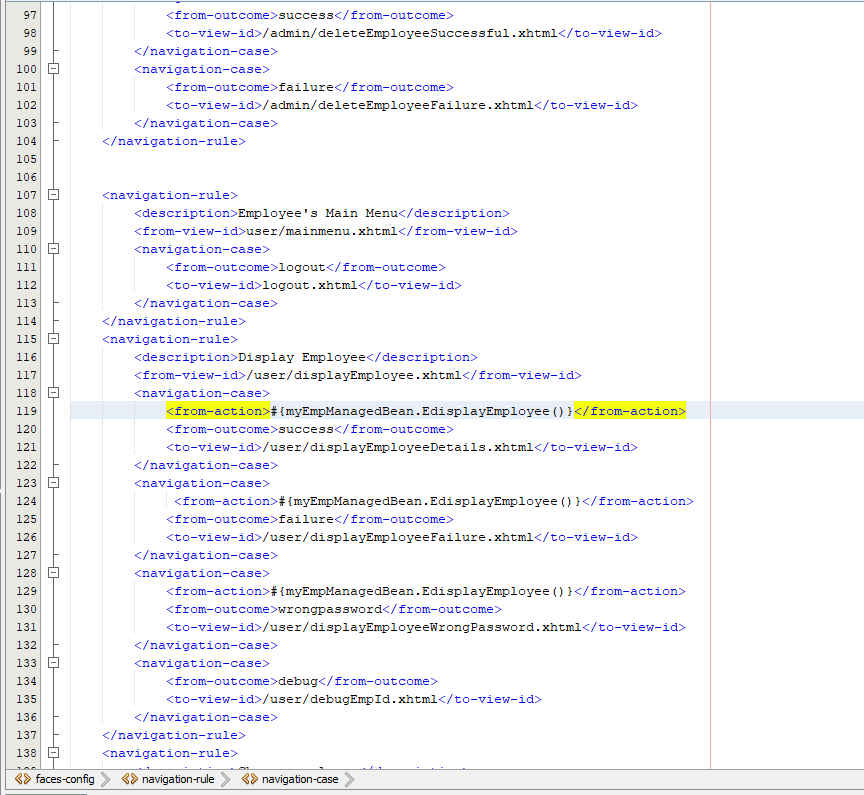
Task5:

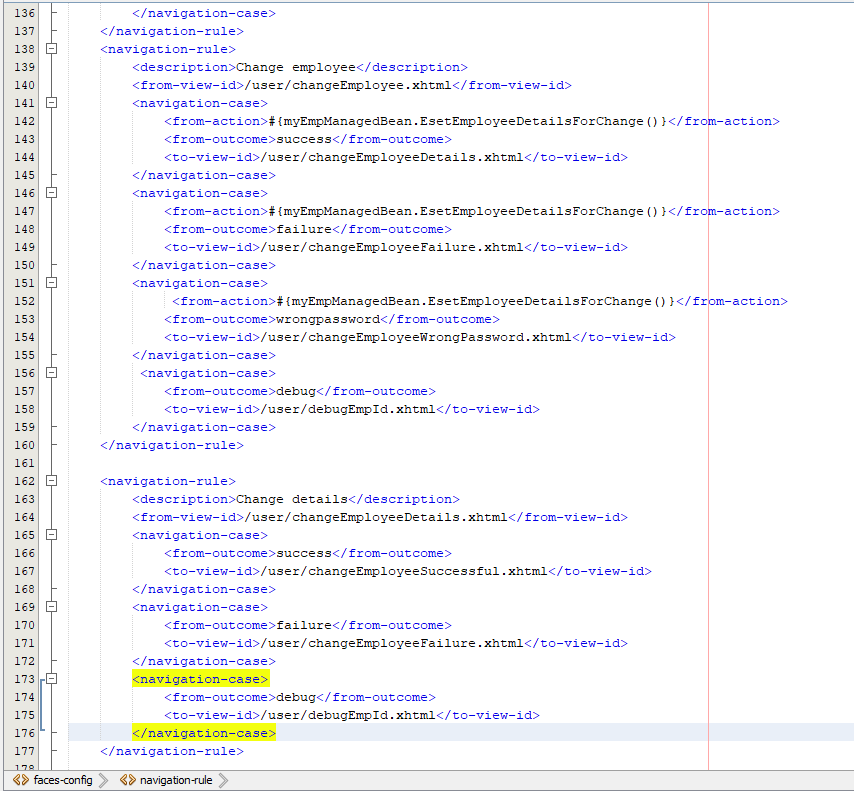


The user will use his or her browser which will give him access to webpages in web tier. These in turn handle the communication between client and business tier (ass users with different roles login to see different webpages or do different actions). Then in the business tier, ManagementFacade transfers 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



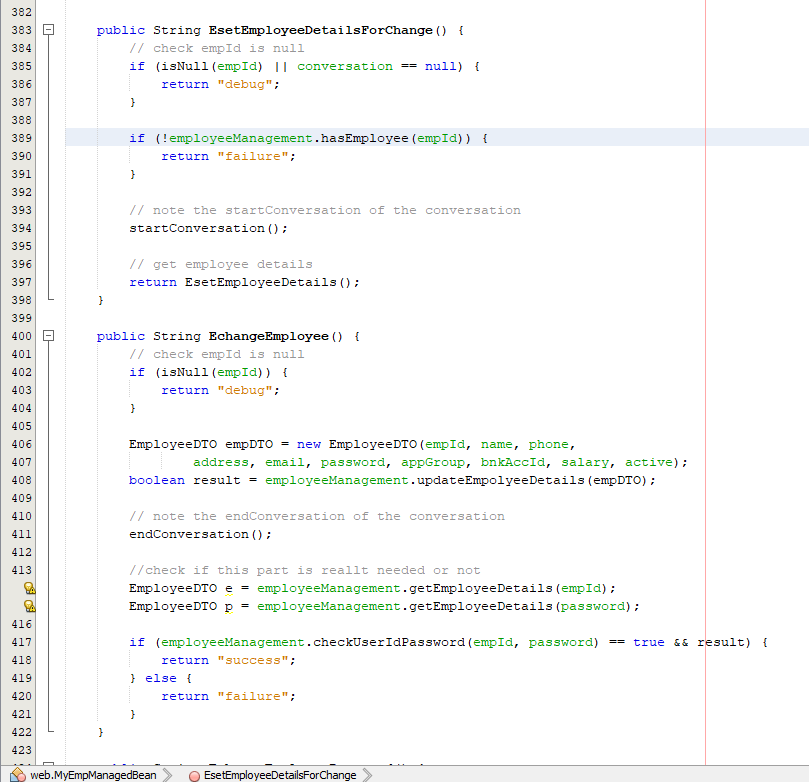


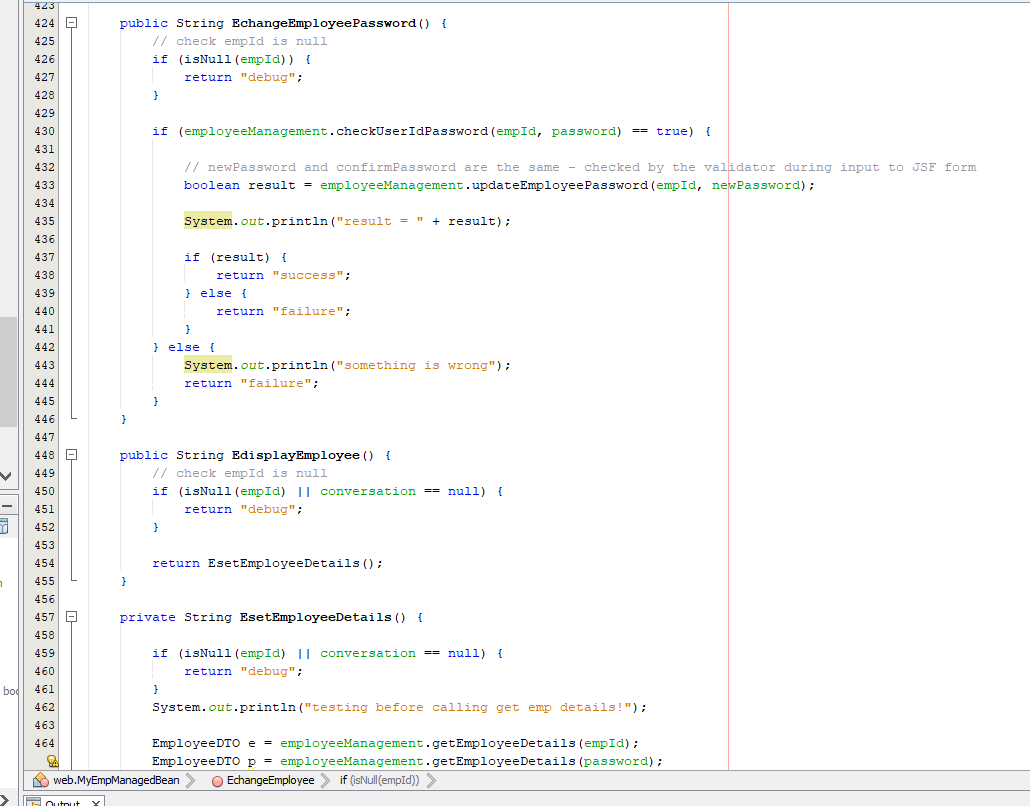


Web file:



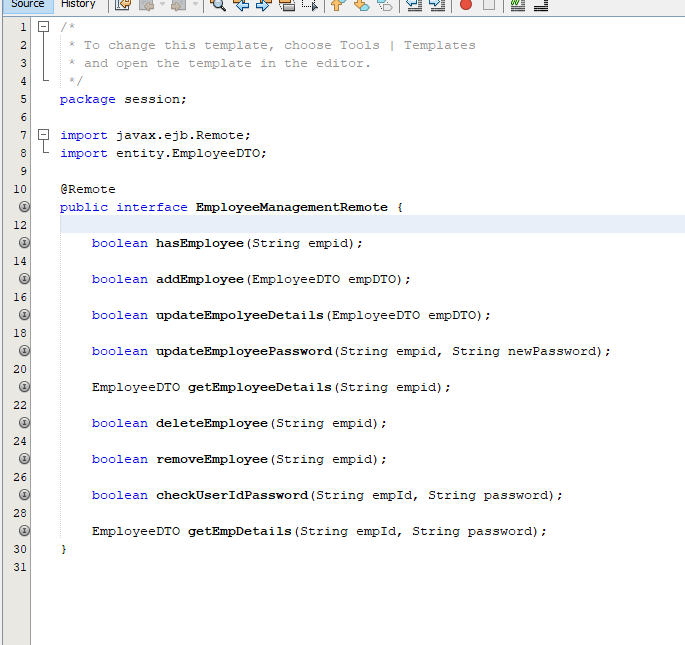
myEmpManagedBean file:



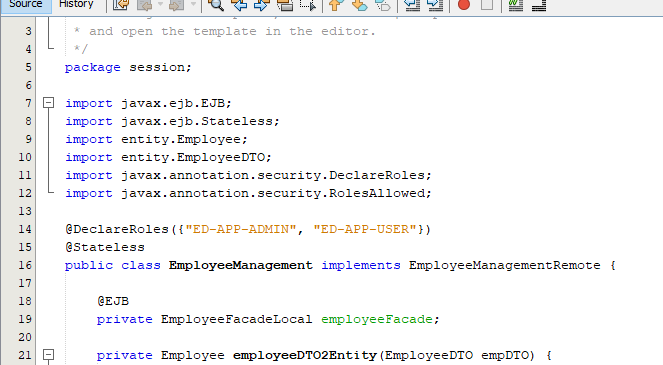


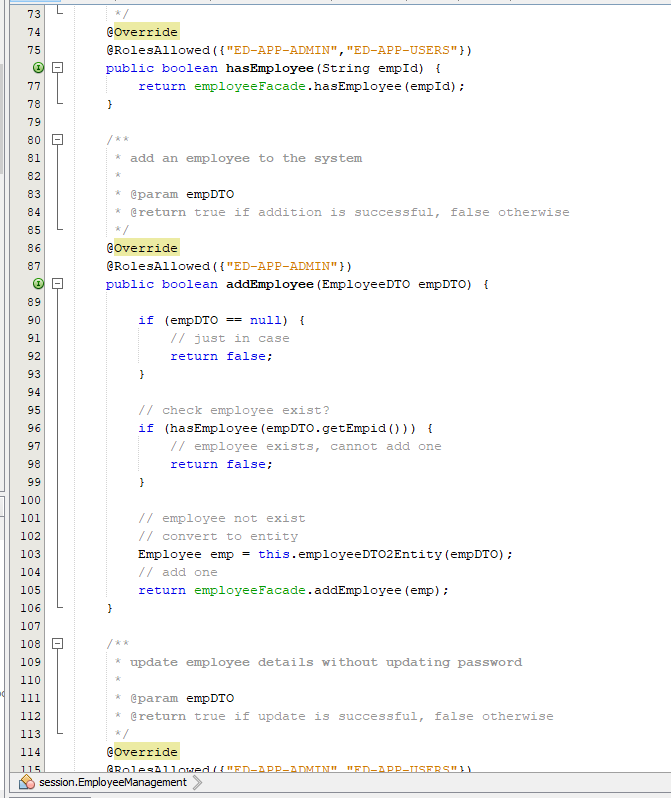


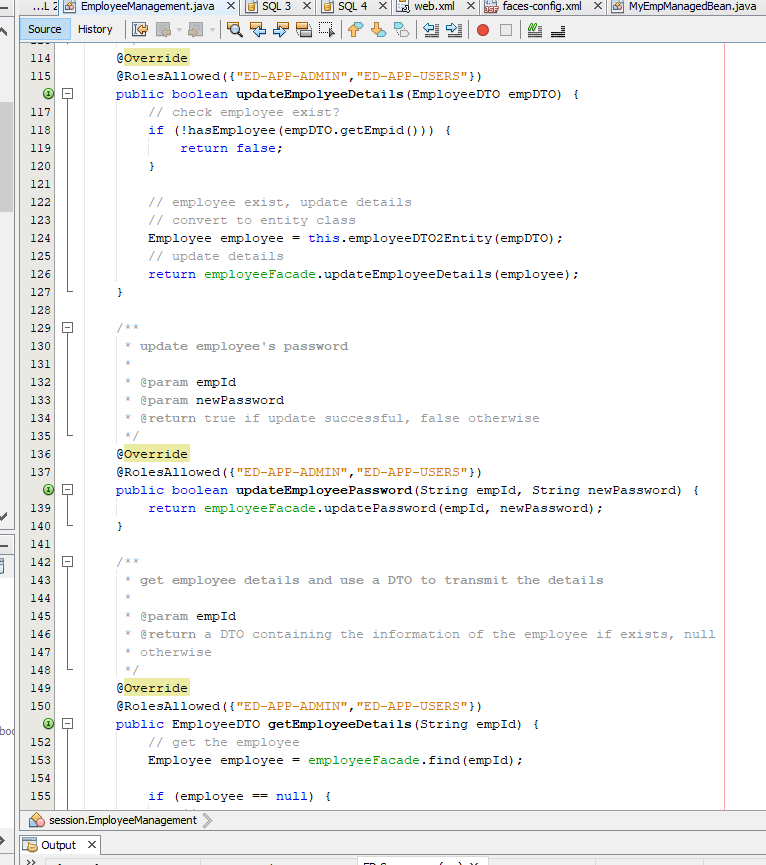
employeeManagementRemote

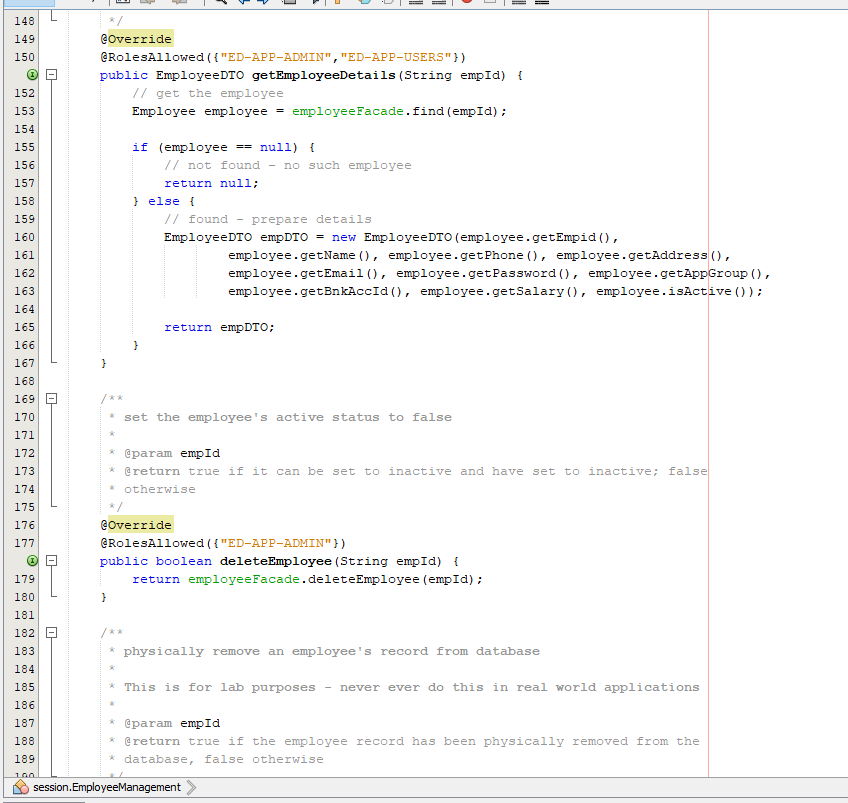


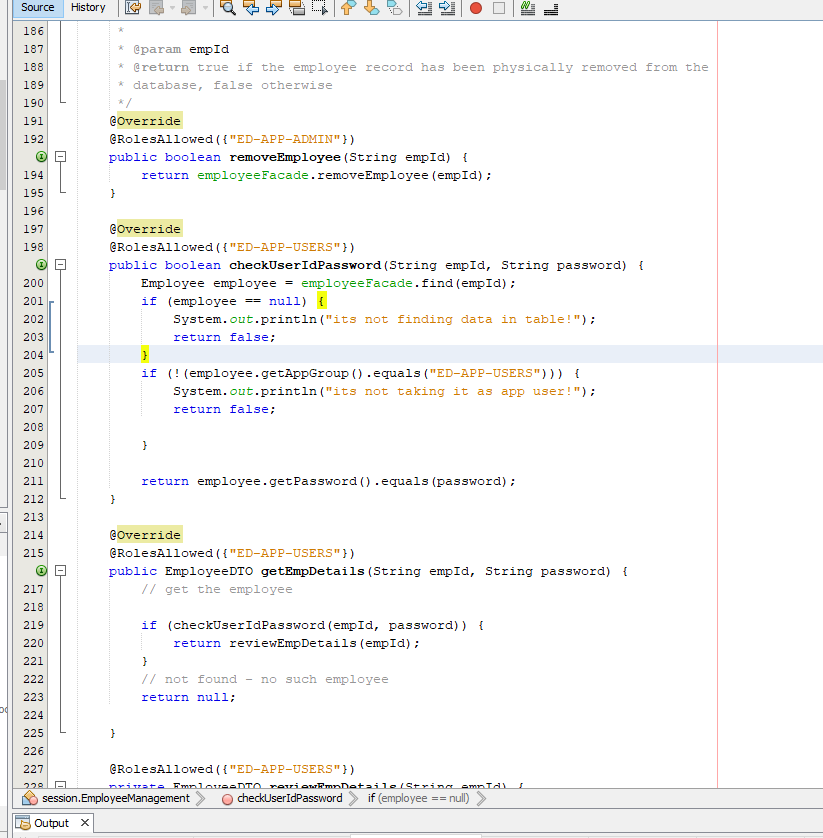
employeeManagement:

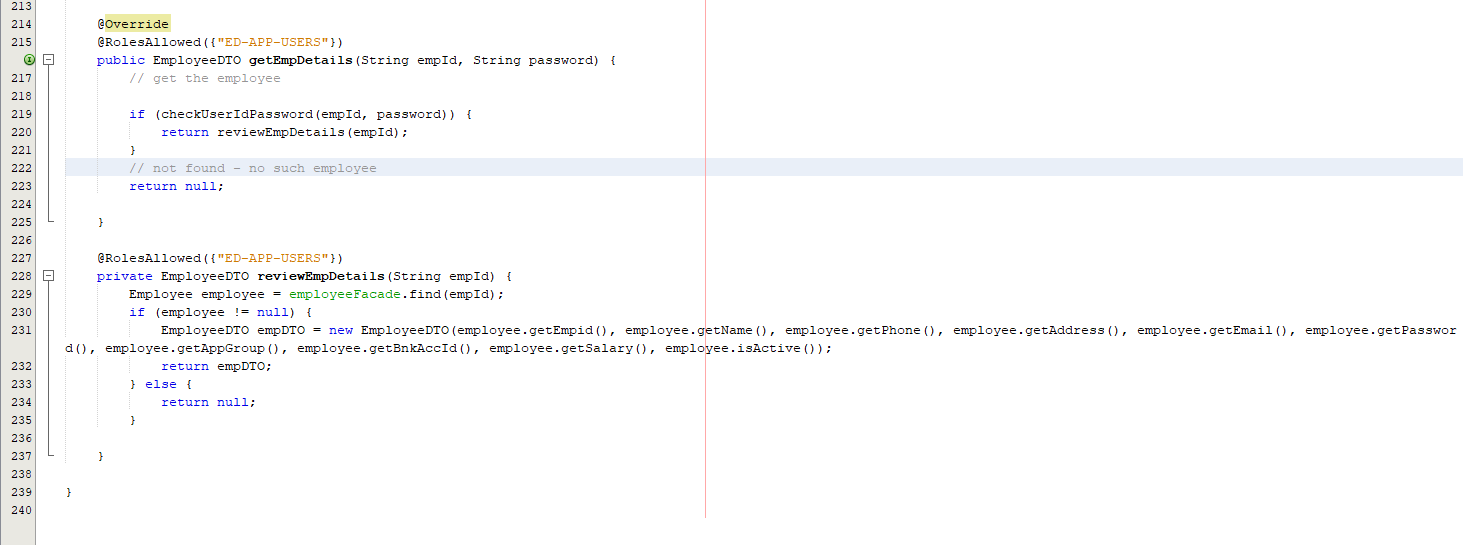




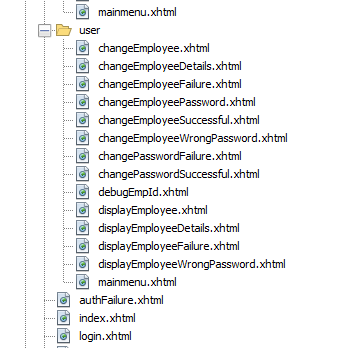






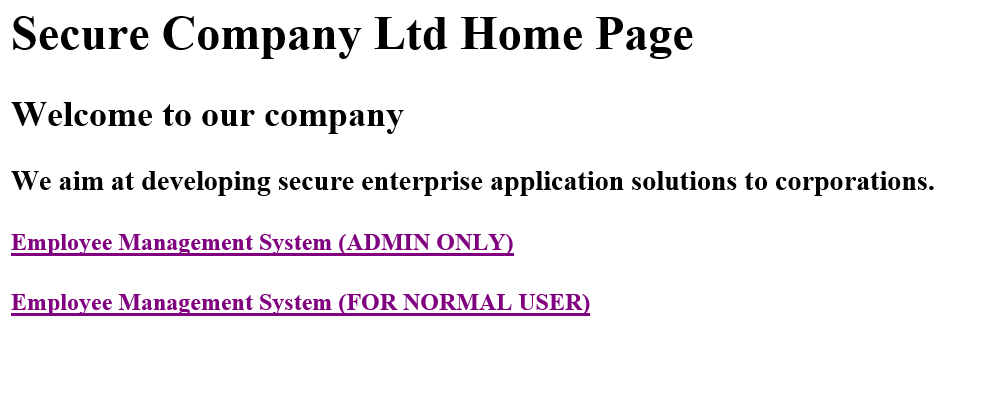


List of webpages added:



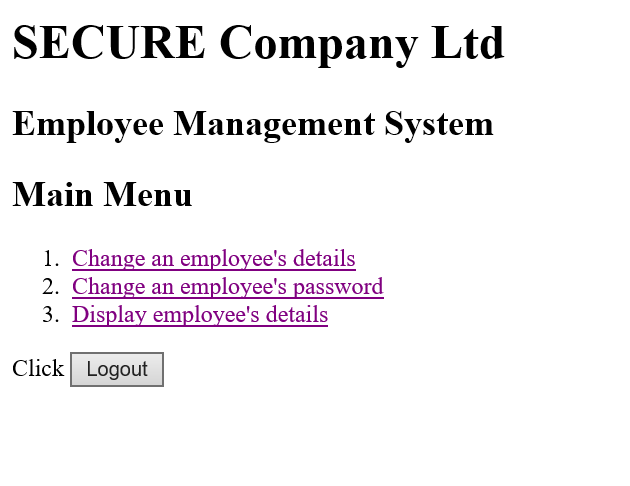
Outputs:

Index page:

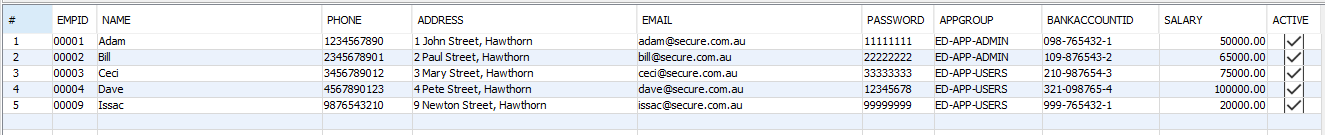
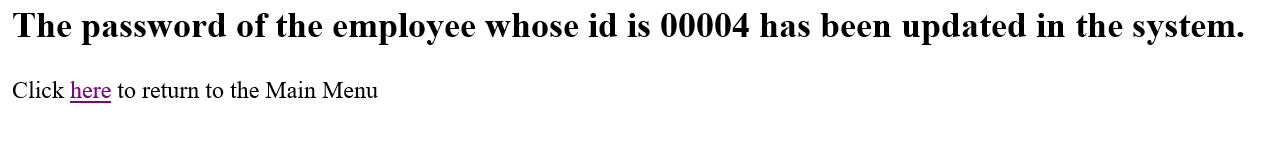
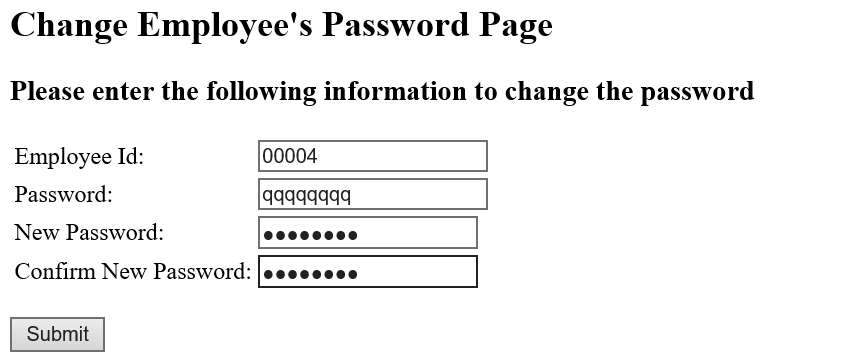
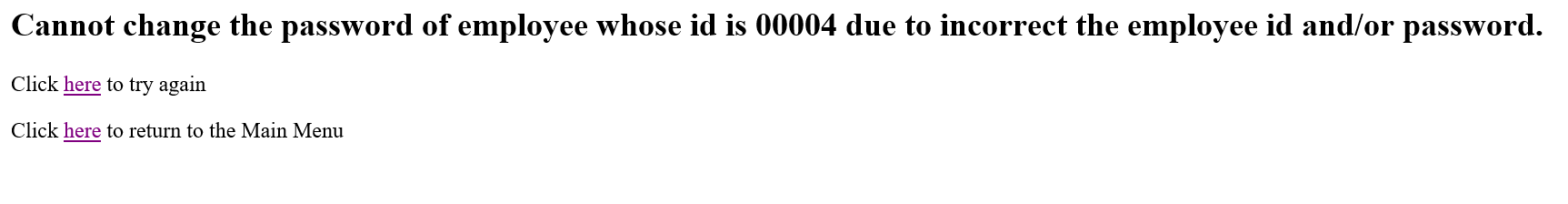
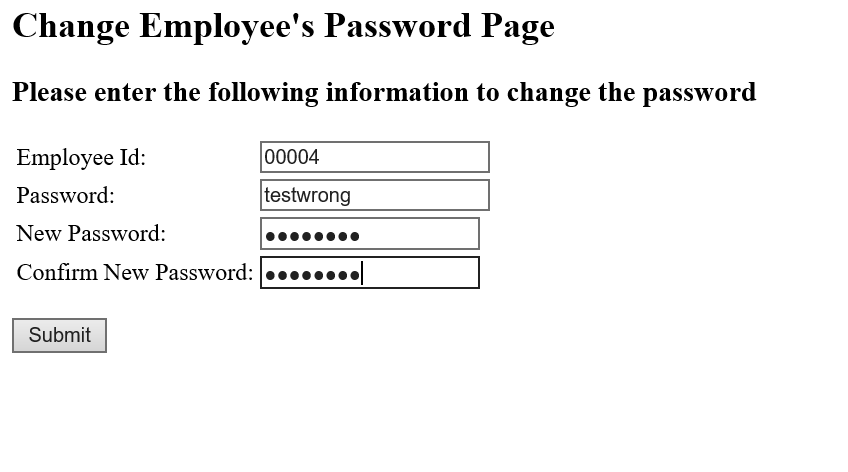


When admin tries to access: 

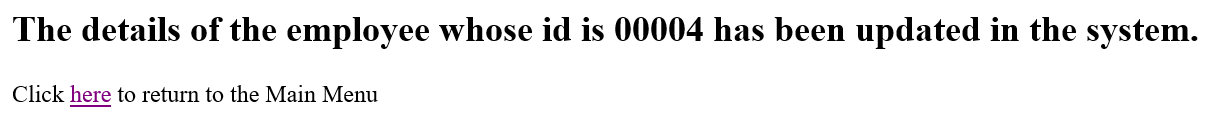
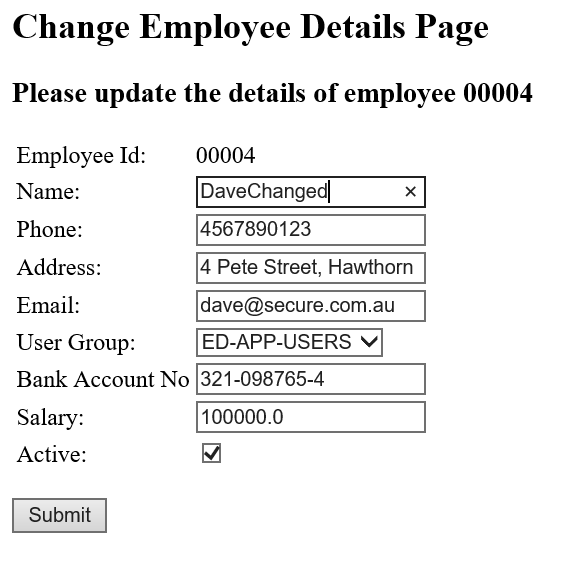
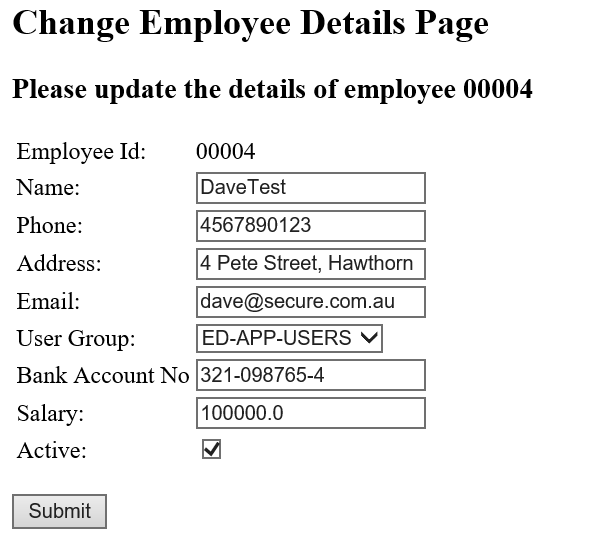
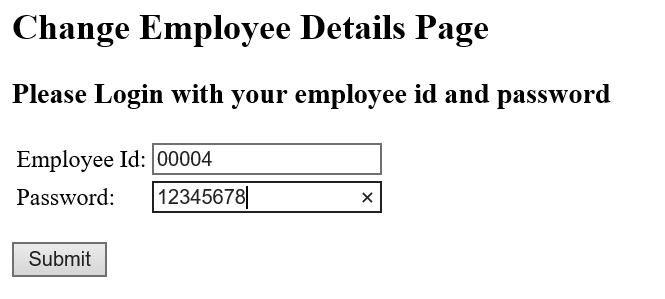
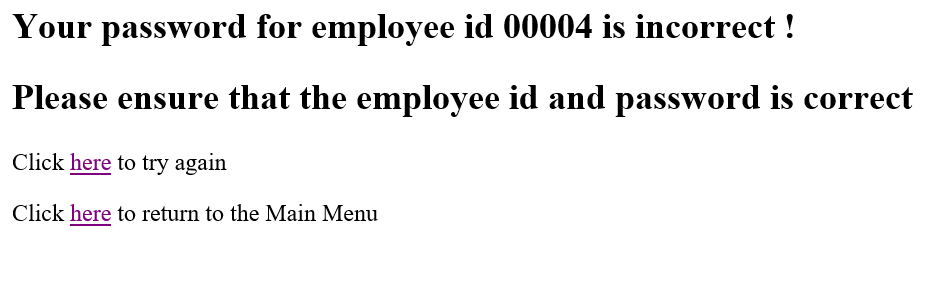
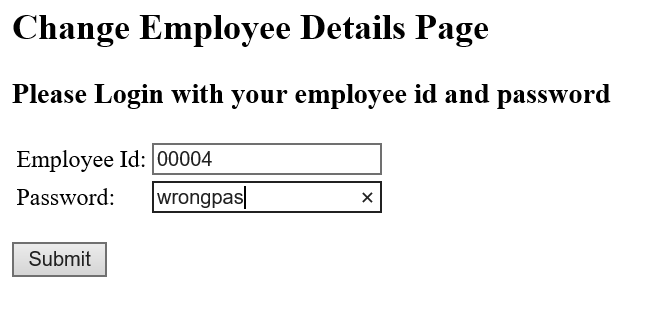
When eduser1 accesses it:



All cases for changing employee password:

Aen eduser1 accesses itss 6ter passing through the verifiyer s or do different actions transfer frome updating thier eded by the

All cases for changing employee detail:



All cases for seeing employee detail:

