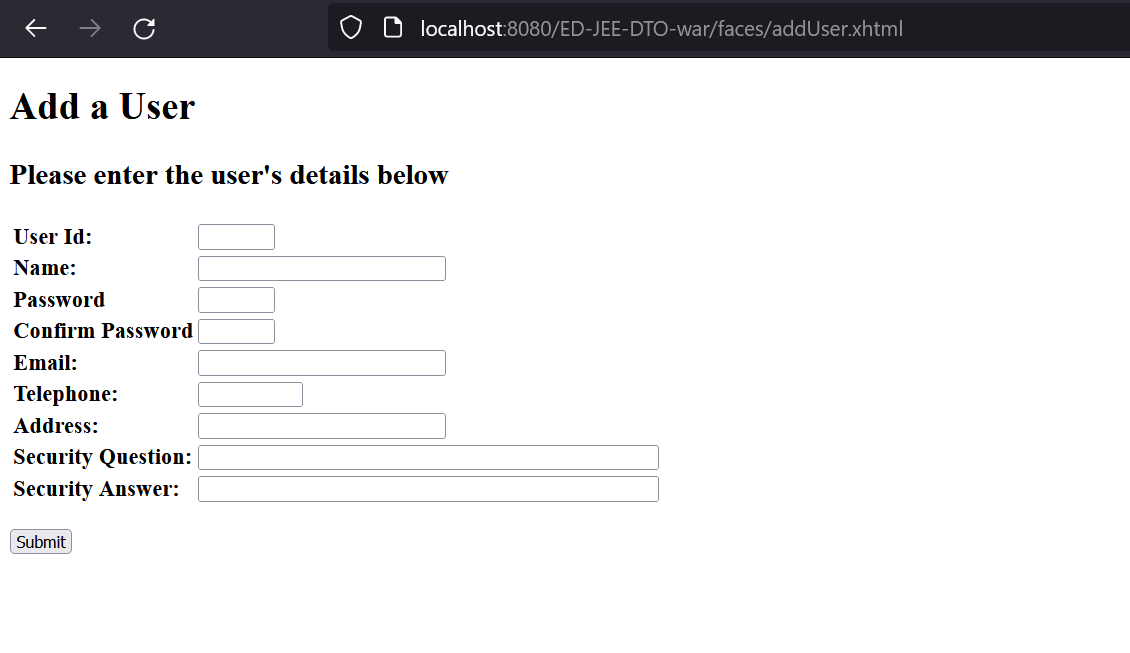
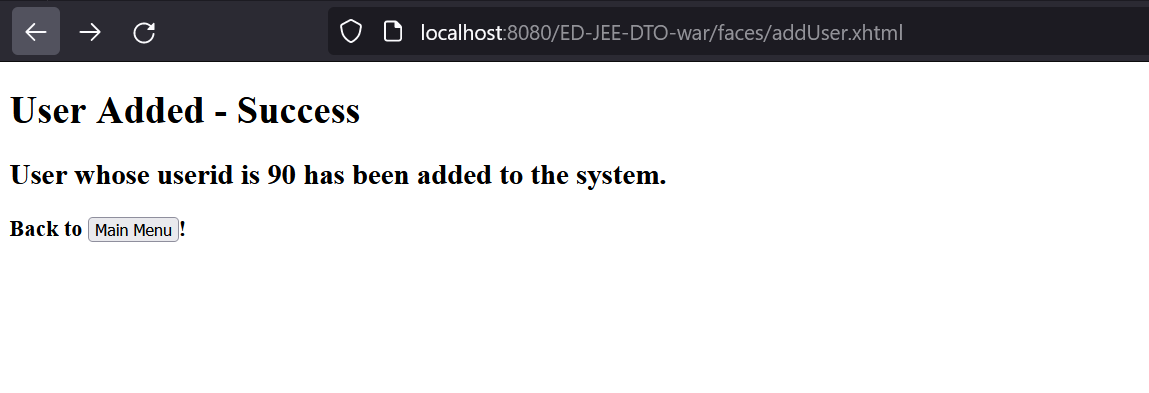
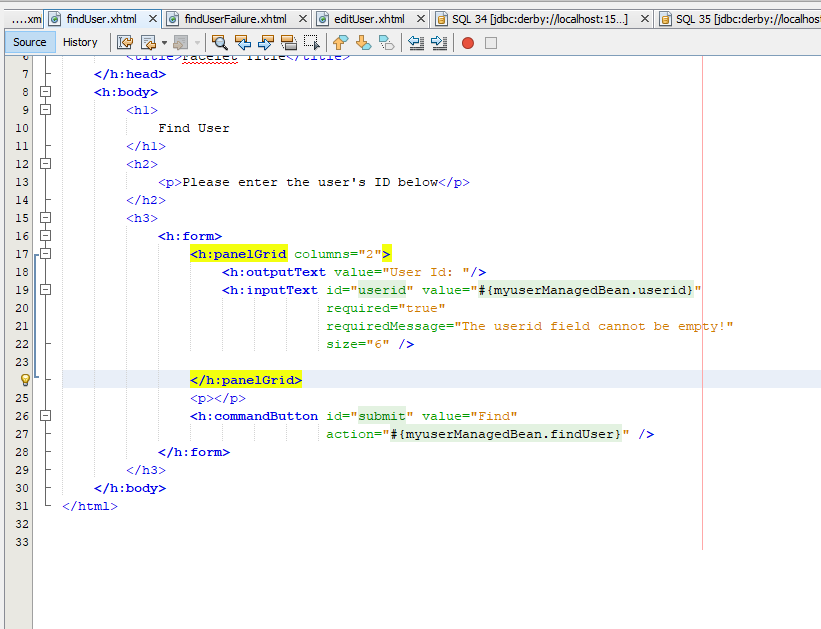
Step 1 ending part:



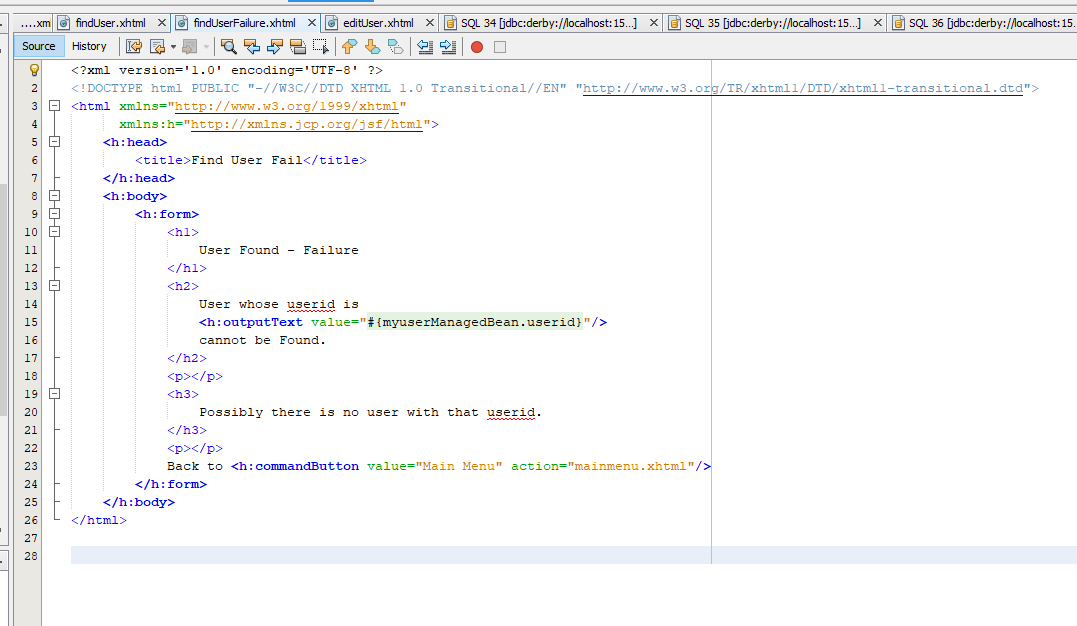


Step 2 all code parts:

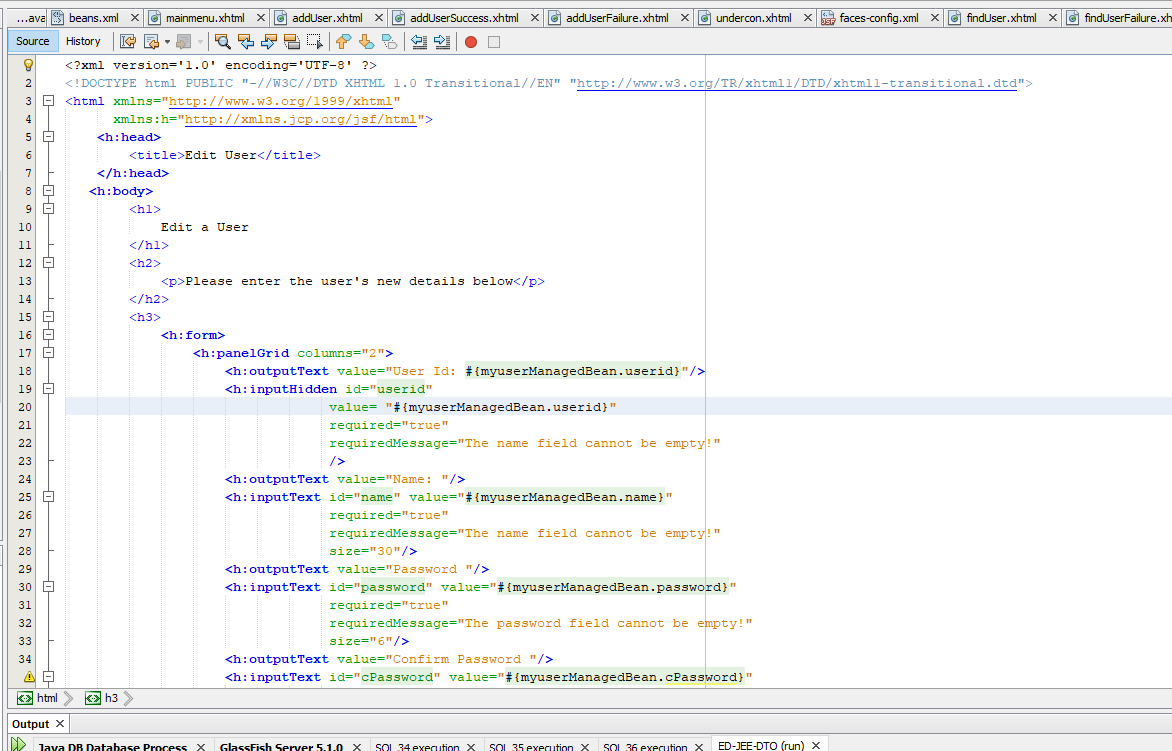
Finduser webpage:

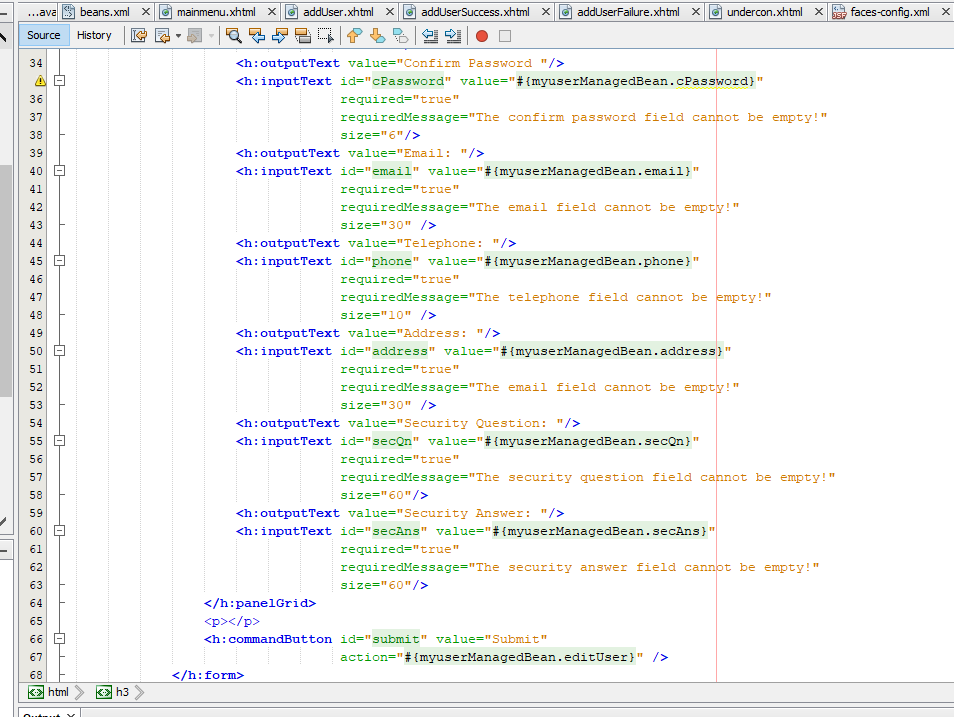


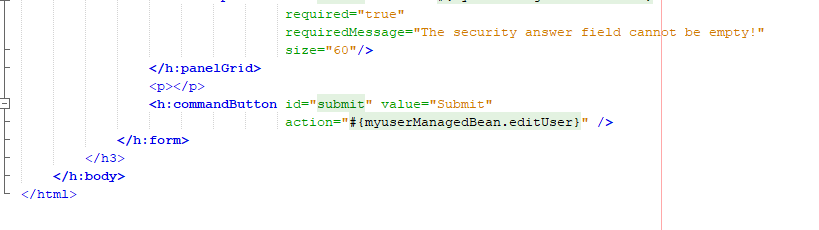
Find user fail webpage:



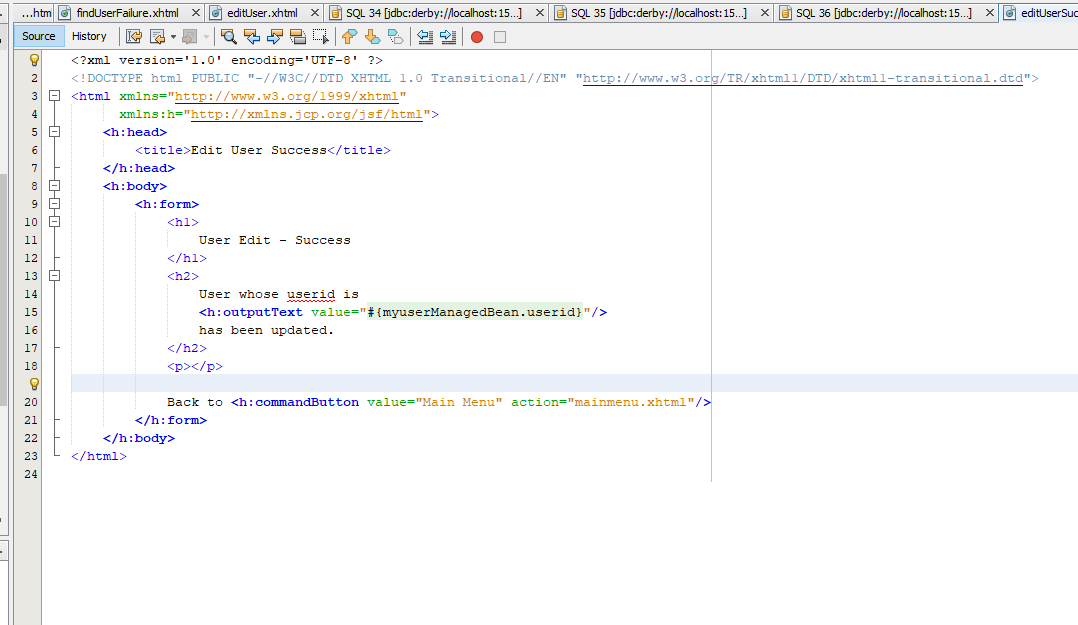
Edit user webpage:



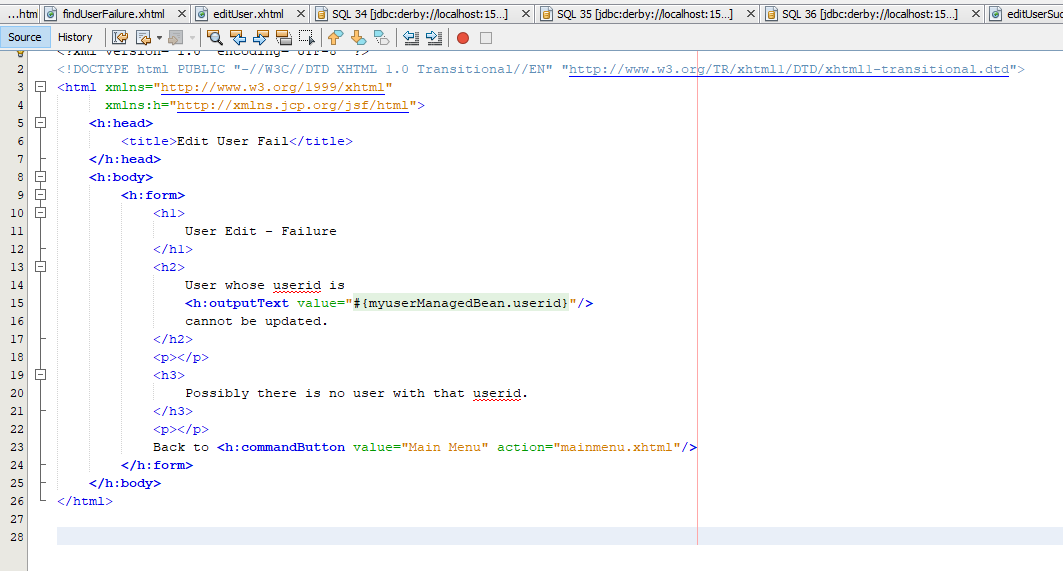




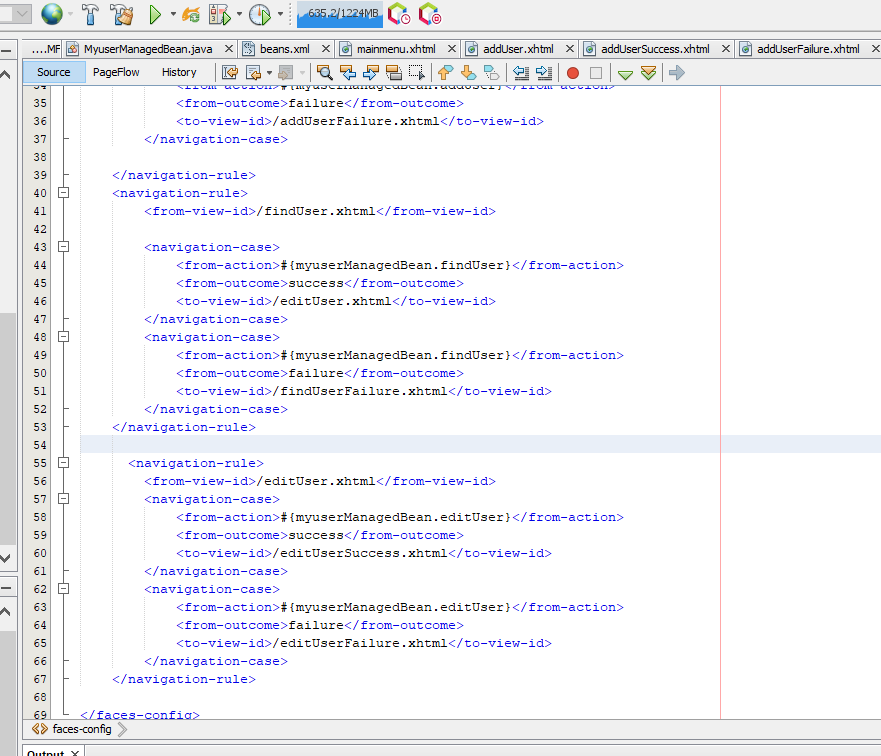
Edit user success webpage:



Edit user fail:

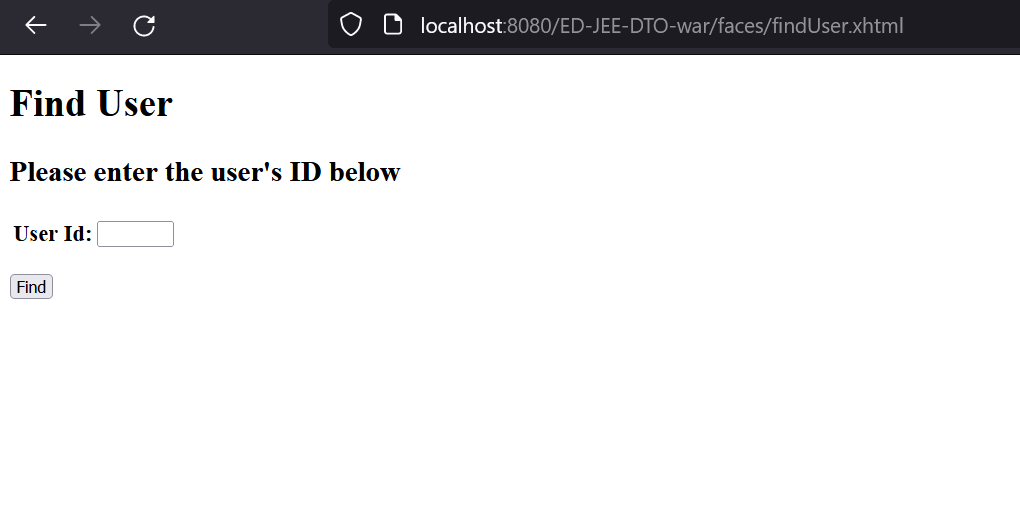
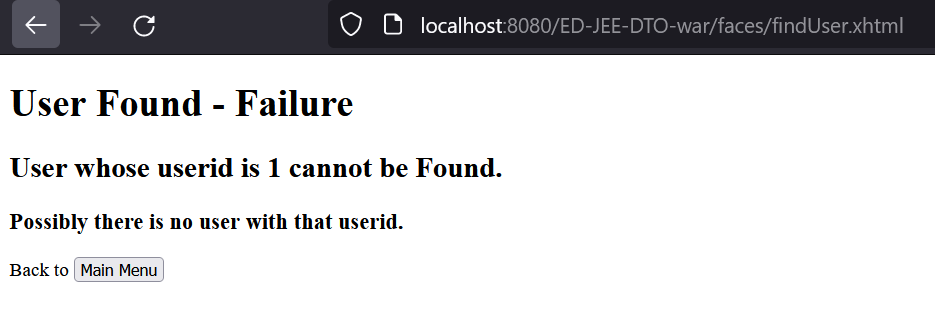


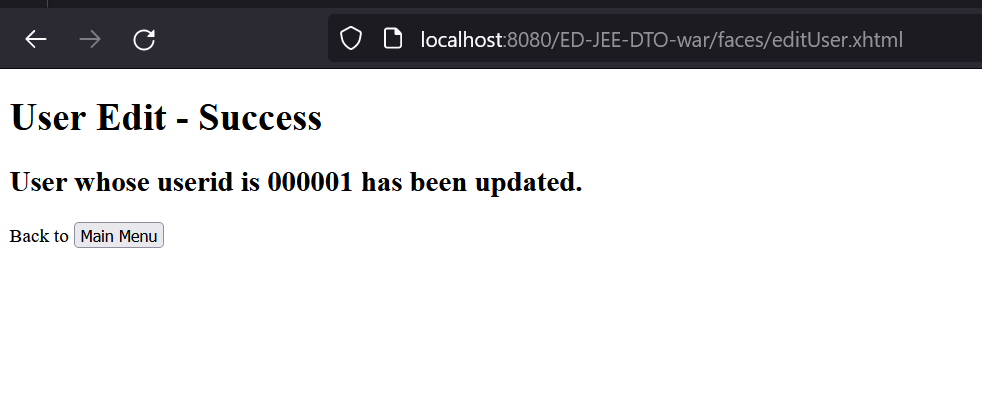
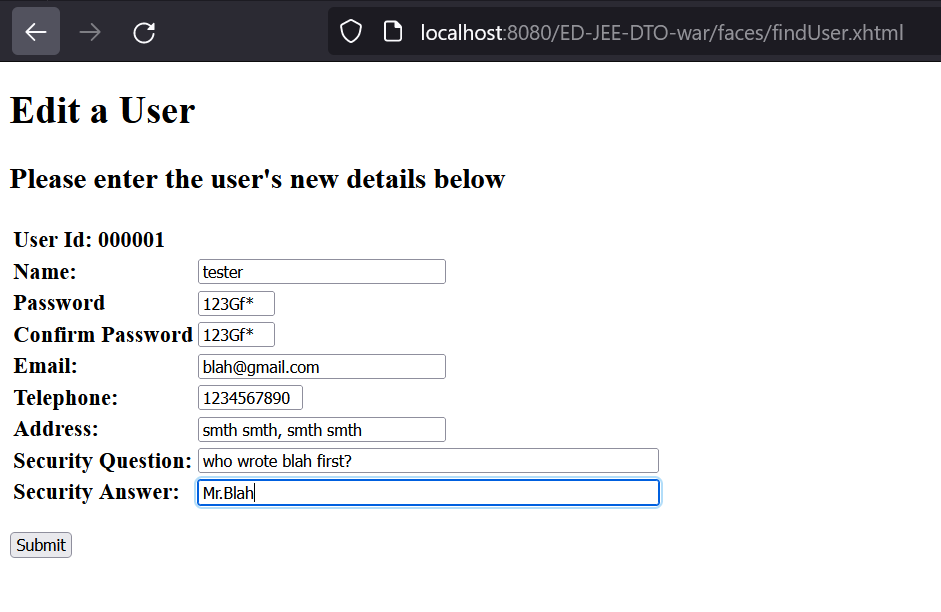
Navigation rules form face-config for find user and edit user all cases:

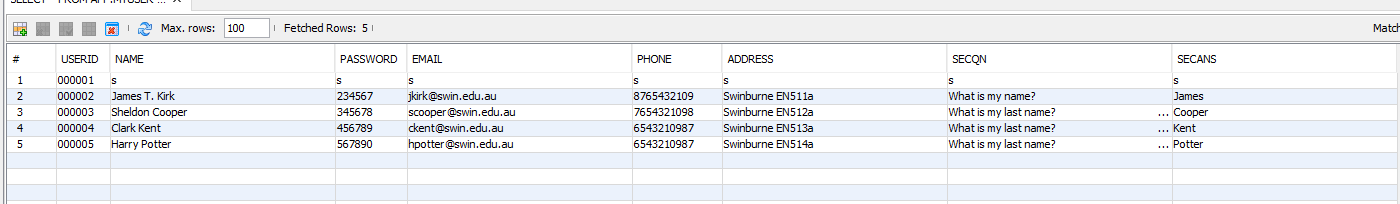


(email code is showing error, so kept it below at end of task 3 screenshots)

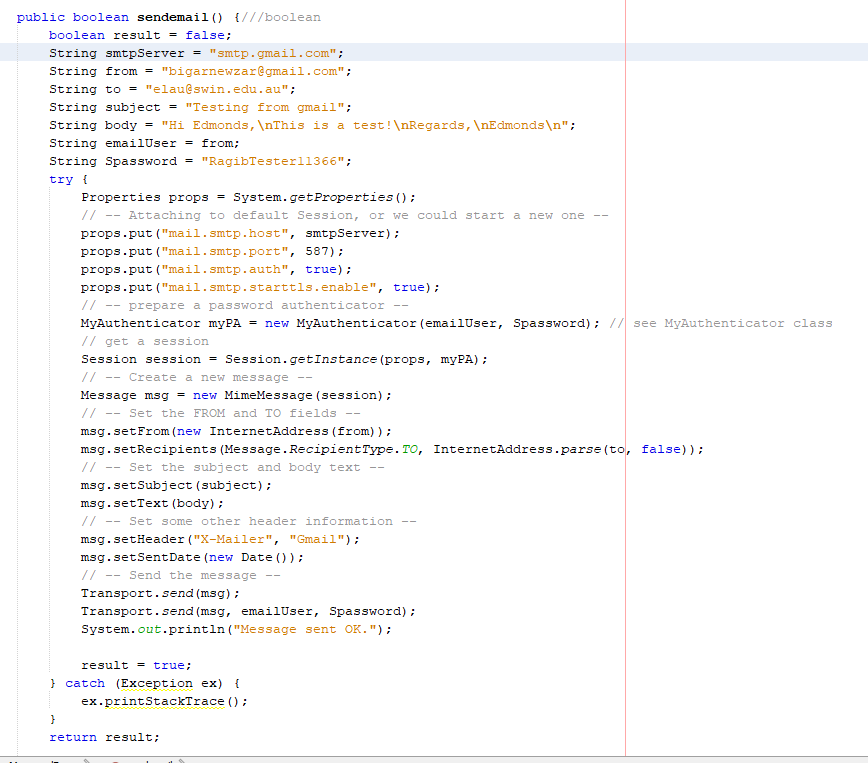
Outputs for find user, find user fail, edit user, edit user success and edit user fail:

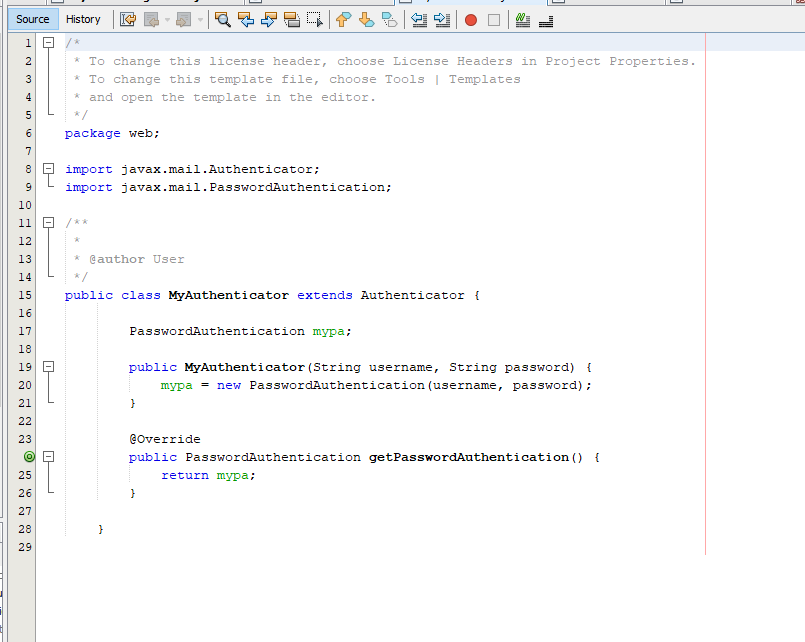
 

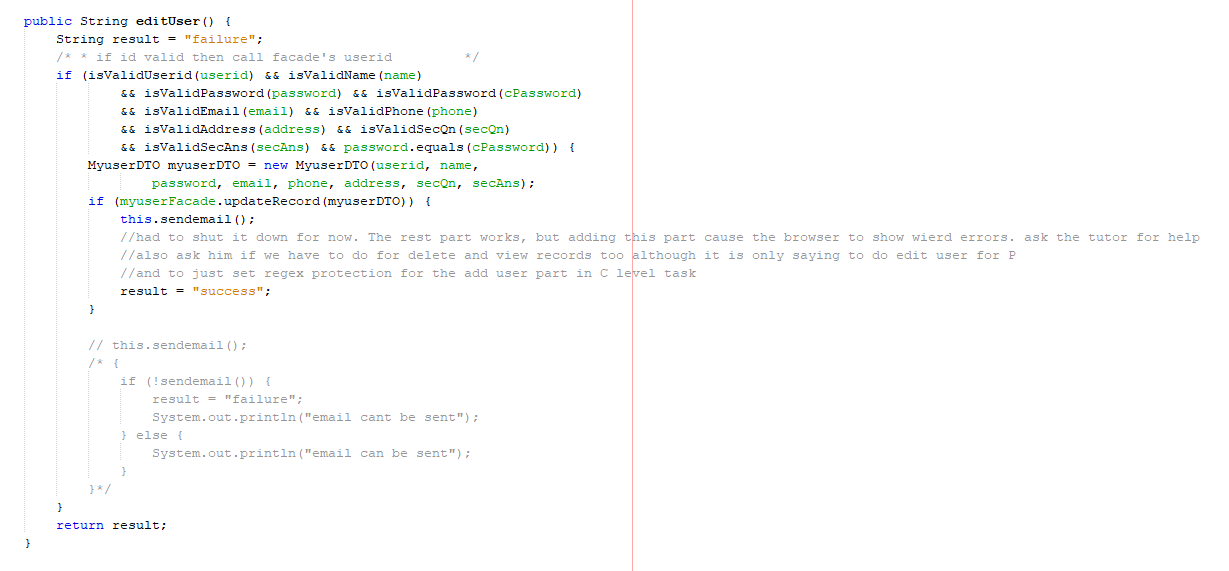


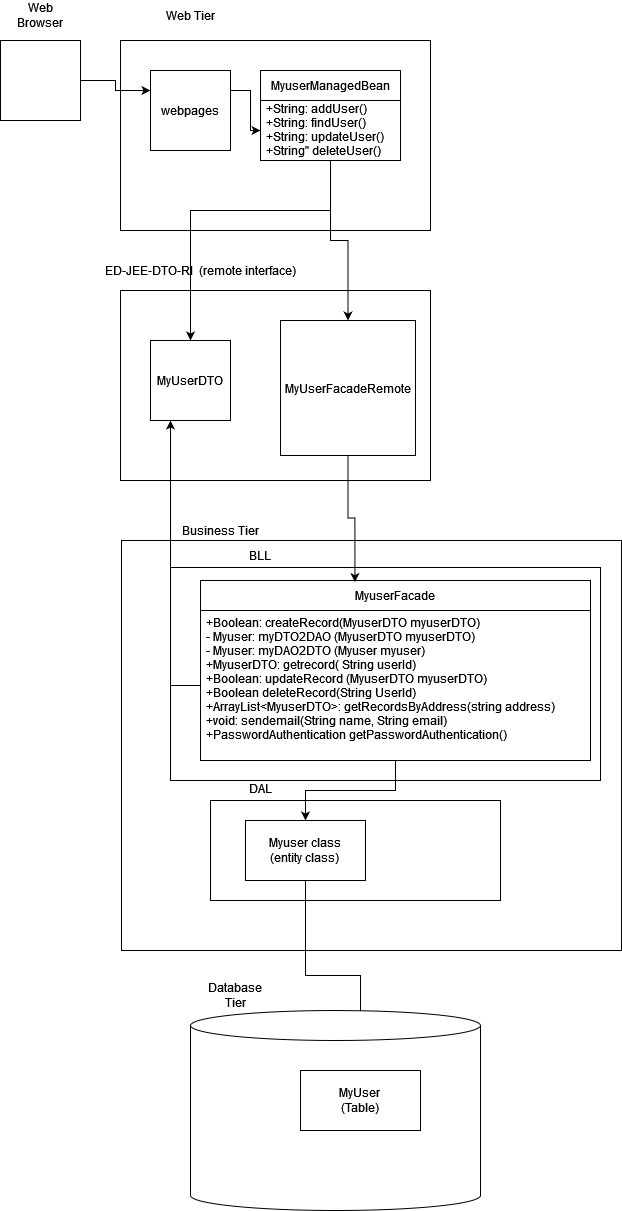


(problem: after adding the email part, the edit user success is no longer working and instead an error shows up. Check that part with him before adding the codes for email and outputs)

Thus sending only the email codes for that one:





Task 4:

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 promt 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.