MyEmpManagedBean.java

```
// SHA-256 hash code to encrypt the password
   public String encryptPassword(String data) {
      try {
         MessageDigest md = MessageDigest.getInstance("SHA-256");
         md.update(data.getBytes("UTF-8")); // Change this to "UTF-16" if needed
         byte[] digest = md.digest();
         BigInteger bigInt = new BigInteger(1, digest);
         return bigInt.toString(16);
      } catch (NoSuchAlgorithmException | UnsupportedEncodingException e) {
      }
      return "encryption error";
   }

public String displayEmployeeP() {
      // check empId is null
      if (isNull(empId) || conversation == null) {
         return "debug";
      }
String password2 = employeeManagement.getEmployeeDetails(empId).getPassword();
      if (!password.equals(password2) ){
          return "failure";
      }
      return setEmployeeDetails();
   }

  public String changeEmployeePasswordP() {
      // check empId is null
      if (isNull(empId)) {
         return "debug";
      }
String password2 = employeeManagement.getEmployeeDetails(empId).getPassword();
      if (!password.equals(password2) ){
          return "failure";
      }

      boolean result = employeeManagement.updateEmployeePassword(empId, newPassword);

      System.out.println("result = " + result);
```

```
        if (result) {
            return "success";
        } else {
            return "failure";
        }
    }
```

faces-config.xml

```
<navigation-rule>
        <from-view-id>/user/changeEmployeeDetails.xhtml</from-view-id>
        <navigation-case>
            <from-outcome>success</from-outcome>
            <to-view-id>/user/changeEmployeeSuccessful.xhtml</to-view-id>
        </navigation-case>
        <navigation-case>
            <from-outcome>failure</from-outcome>
            <to-view-id>/user/changeEmployeeFailure.xhtml</to-view-id>
        </navigation-case>
    </navigation-rule>
    <navigation-rule>
        <from-view-id>/user/changeEmployeePassword.xhtml</from-view-id>
        <navigation-case>
            <from-outcome>success</from-outcome>
            <to-view-id>/user/changeEmployeePasswordSuccessful.xhtml</to-view-id>
        </navigation-case>
        <navigation-case>
            <from-outcome>failure</from-outcome>
            <to-view-id>/user/changeEmployeePasswordFailure.xhtml</to-view-id>
        </navigation-case>
    </navigation-rule>
</faces-config>
```

glassfish-web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE glassfish-web-app PUBLIC "-//GlassFish.org//DTD GlassFish Application Server 3.1 Servlet
3.0//EN" "http://glassfish.org/dtds/glassfish-web-app_3_0-1.dtd">
```

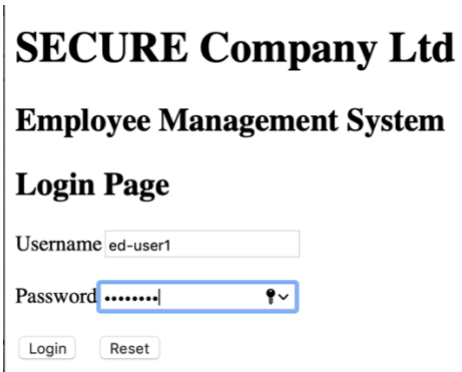
```
<glassfish-web-app error-url="">
  <class-loader delegate="true"/>
  <jsp-config>
    <property name="keepgenerated" value="true">
      <description>Keep a copy of the generated servlet class' java code.</description>
    </property>
  </jsp-config>
  <security-role-mapping>
      <role-name>ED-APP-ADMIN<role-name>
      <group-name>ED-APP-ADMIN<group-name>
  </security-role-mapping>
    <security-role-mapping>
      <role-name>ED-APP-USERS<role-name>
      <group-name>ED-APP-USERS<group-name>
  </security-role-mapping>
</glassfish-web-app>
```

glassfish-resources.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE resources PUBLIC "-//GlassFish.org//DTD GlassFish Application Server 3.1 Resource
Definitions//EN" "http://glassfish.org/dtds/glassfish-resources_1_5.dtd">
<resources>
  <jdbc-resource enabled="true" jndi-name="jdbc/authenticate" object-type="user" pool-
name="AuthenticationconnectionPool">
    <description/>
  </jdbc-resource>
  <jdbc-connection-pool allow-non-component-callers="false" associate-with-thread="false" connection-creation-
retry-attempts="0" connection-creation-retry-interval-in-seconds="10" connection-leak-reclaim="false"
connection-leak-timeout-in-seconds="0" connection-validation-method="auto-commit" datasource-
classname="org.apache.derby.jdbc.ClientDataSource" fail-all-connections="false" idle-timeout-in-seconds="300"
is-connection-validation-required="false" is-isolation-level-guaranteed="true" lazy-connection-association="false"
lazy-connection-enlistment="false" match-connections="false" max-connection-usage-count="0" max-pool-
size="32" max-wait-time-in-millis="60000" name="AuthenticationconnectionPool" non-transactional-
connections="false" pool-resize-quantity="2" res-type="javax.sql.ConnectionPoolDataSource" statement-timeout-
in-seconds="-1" steady-pool-size="8" validate-atmost-once-period-in-seconds="0" wrap-jdbc-objects="false">
    <property name="URL" value="jdbc:derby://localhost:1527/sun-appserv-samples;create=true"/>
    <property name="serverName" value="localhost"/>
    <property name="PortNumber" value="1527"/>
    <property name="DatabaseName" value="sun-appserv-samples"/>
    <property name="User" value="APP"/>
    <property name="Password" value="APP"/>
  </jdbc-connection-pool>
</resources>
```

web.xml

```
<resource-ref>
        <res-ref-name>jdbc/authenticate</res-ref-name>
        <res-type>javax.sql.ConnectionPoolDataSource</res-type>
        <res-auth>Container</res-auth>
        <res-sharing-scope>Shareable</res-sharing-scope>
    </resource-ref>
```



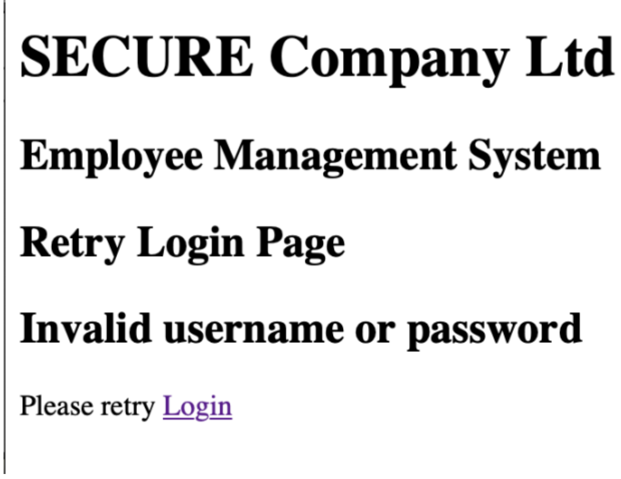
SECURE Company Ltd

Employee Management System

Login Page

Username ed-user1

Password ••••••••



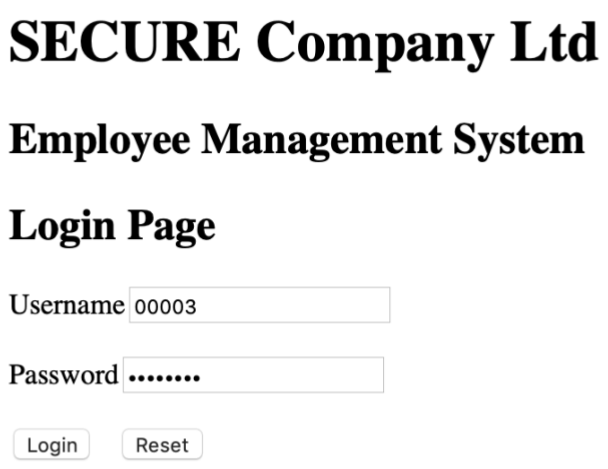
SECURE Company Ltd

Employee Management System

Retry Login Page

Invalid username or password

Please retry Login



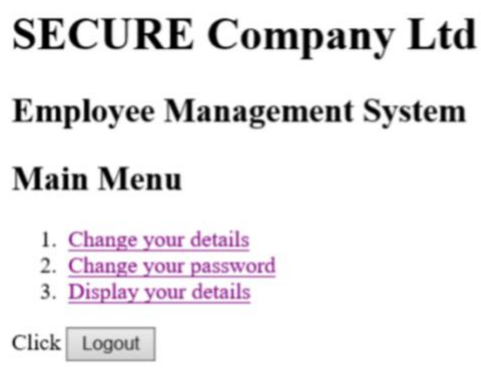
SECURE Company Ltd

Employee Management System

Login Page

Username 00003

Password ••••••••



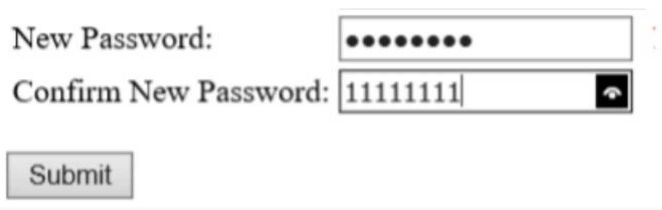
SECURE Company Ltd

Employee Management System

Main Menu

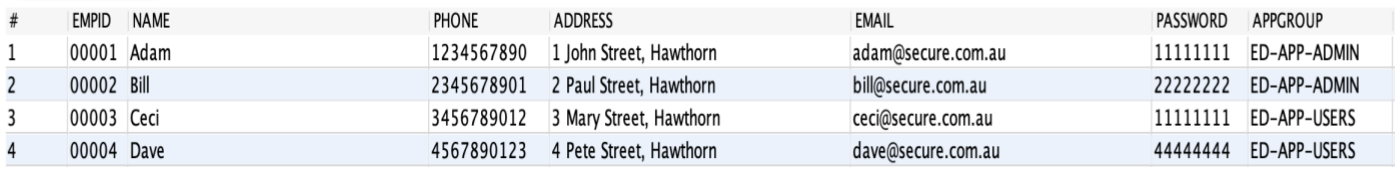
1. Change your details
2. Change your password
3. Display your details

Click Logout



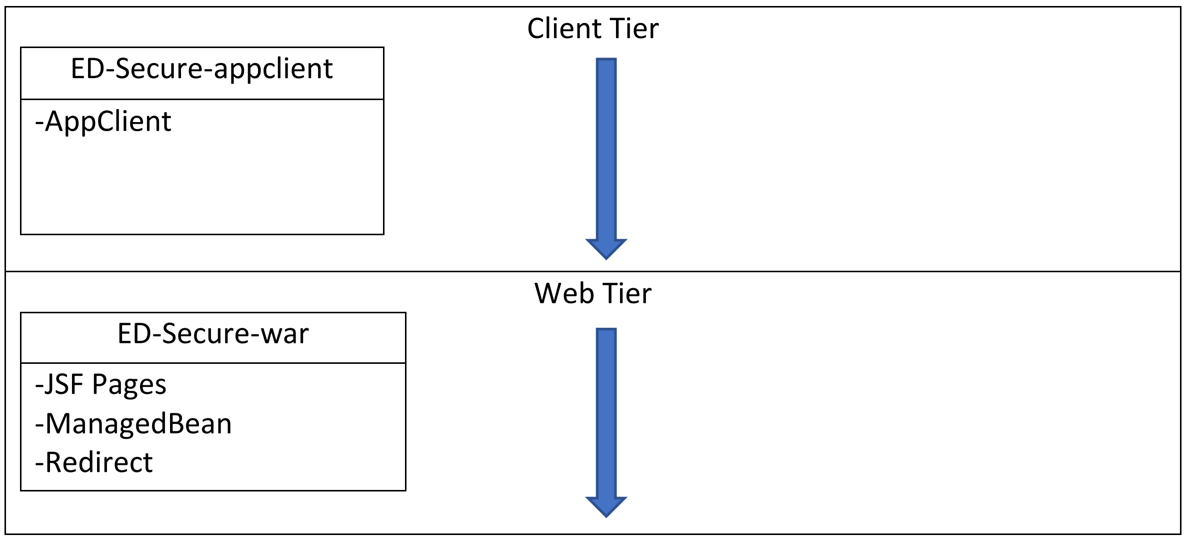
New Password: •••••••••

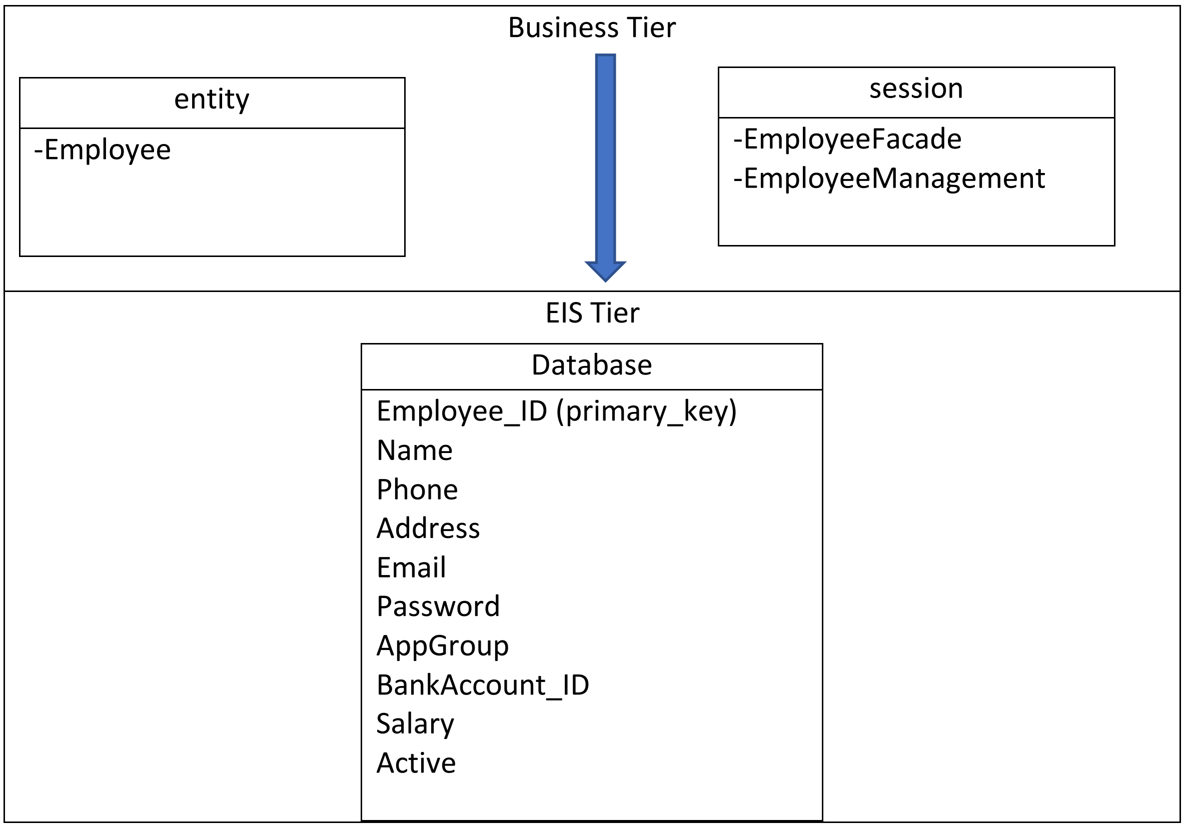
Confirm New Password: 11111111



#	EMPID	NAME	PHONE	ADDRESS	EMAIL	PASSWORD	APPGROUP
1	00001	Adam	1234567890	1 John Street, Hawthorn	adam@secure.com.au	11111111	ED-APP-ADMIN
2	00002	Bill	2345678901	2 Paul Street, Hawthorn	bill@secure.com.au	22222222	ED-APP-ADMIN
3	00003	Ceci	3456789012	3 Mary Street, Hawthorn	ceci@secure.com.au	11111111	ED-APP-USERS
4	00004	Dave	4567890123	4 Pete Street, Hawthorn	dave@secure.com.au	44444444	ED-APP-USERS

4.1





Client tier component runs on the client machine. For this instance, the web clients contain dynamic web pages from HTML and XML generated by web components running on the web tier.
Web tier component runs on the Jave EE server. This web tier consists of web pages created using JSF. The SHA-256 encryption is encoded here. When the user puts in or change their password, it will transform the data from string to hash code.
Business tier component runs on the Java EE component. Business code/ logic receives data from client, processes it and sends it to the enterprise information system tier for storage, vice versa.
EIS tier stores database along with the credentials of each employee. The database is encrypted with SHA-256 for security purposes. The data is accessed using jdbcRealm upon request.

4.2

SHA-256 was implemented to encrypt any password is inserted or updated in the system to preserve credential security of the users.
To change the passwords, new password needs to be double-checked to confirm correct. This is a form of data validation, by double-checking the inputted data, it will detect any error if misusing and avoid data collision.

4.3 SHA-256 encryption technology is encoded in the application for security purposes. Firstly, if attacked by brute-force, it would need to make 2^256 attempts to retrieve the initial data. Furthermore, it decreases the likelihood of having 2 messages with the same hash value, hence avoid collision. Finally, it minorly alters the hash value so that it’s hard to determine between the hash value and its derived data.

5

During the process of completing this portfolio task, the most valuable knowledge that I have learnt is the connection between different components in 1 program (client tier, web tier, business tier, EIS tier), moreover the importance of password encryption to secure the application and its users. I learnt about the importance and mission of each tier when it comes to building a complete web app program. In task 7.2, the implementation of SHA-256 is a very interesting encryption as it is a very important security aspect of a program which is highly regard nowadays. The lack of information security is dangerously shown in current apps in the market with old-fashioned techniques, however it is also reasonable why they did so.
This lab task was a recap of what I learnt in a previous Creating Web Application unit. GUI-wise, there could had been updates with colour, picture, etc if the time allowed.
In the process of doing this lab task, there were several problems regarding to the run, deploy issues. Setting to different realm and encryption caused myself a bit of trouble logging in with new credentials. Alternatively, in order to perfect this task, HTML CSS PHP JavaScript could be used in order to further enhance the task.