

## Security Tutorial: Part 2

**Author: Vuyisile Memani**

### **Purpose:**

The purpose of this tutorial is to demonstrate how to secure a database-driven web application. In this application, we have two roles, a **teacher** and a **student**. A teacher can do the following functionalities:

- Add a student into the database
- Find a student in the database
- Remove a student from the database
- Remove a student from the database
- View all the students in the database

A student can only do the following:

- View all the students in the database

In this tutorial, we will have specific users. The table below shows the users:

No.	Username	Password
1	vuyi	123
2	khomo	456

The tutorial is divided into four parts:

- Part A: we create a web application that anyone can use to add, edit, find, remove, and get all students.
- Part B: we test the web application.
- Part C: we include security measures to the application.
- Part D: we test the secured application.

## A. Create the web application

### 1. Welcome page

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
  <head>
    <title>Home Page</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <h1>Welcome</h1>
    <p>
      Welcome to our web site. Please click <a href="menu.html">here</a> to view the options.
    </p>
  </body>
</html>
```

## 2. Menu

```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
  <head>
    <title>Menu Page</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <h1>Menu options</h1>
    <p>
      Please select one of the following options or click <a href="index.html">here</a> to go back to the welcome page:
    </p>
    <ol>
      <li><a href="addstudent.html">Add a student</a></li>
      <li><a href="findstudent.html">Find a student</a></li>
      <li><a href="removestudent.html">Remove a student</a></li>
      <li><a href="editstudent.html">Edit a student</a></li>
      <li><a href="getallstudents.html">View all students</a></li>
    </ol>
  </body>
</html>
```

### 3. HTML pages

#### 3.1 addstudent.html

```
<html>
  <head>
    <title>Add Student Page</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <h1>Add student</h1>
    <p>
      Please enter the details of the student below or <a href="menu.html">here</a> to go back to the menu page:
    </p>
    <form action="AddStudentServlet.do" method="POST">
      <table>
        <tr>
          <td>Name:</td>
          <td><input type="text" name="name"/></td>
        </tr>
        <tr>
          <td>Age</td>
          <td><input type="text" name="age"/></td>
        </tr>
        <tr>
          <td>Student number:</td>
          <td><input type="text" name="id"/></td>
        </tr>
        <tr>
          <td></td>
          <td><input type="submit" value="ADD"/></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

### 3.2 findstudent.html

```
<html>
  <head>
    <title>Find Student Page</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <h1>Find student</h1>
    <p>
      Please enter the student number below:
    </p>
    <form action="FindStudentServlet.do" method="POST">
      <table>
        <tr>
          <td>Student number:</td>
          <td><input type="text" name="id"/></td>
        </tr>
        <tr>
          <td></td>
          <td><input type="submit" value="FIND"/></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

### 3.3 removestudent.html

```
<html>
  <head>
    <title>Remove Student Page</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <h1>Remove student</h1>
    <p>
      Please enter the student number below:
    </p>
    <form action="RemoveStudentServlet.do" method="POST">
      <table>
        <tr>
          <td>Student number:</td>
          <td><input type="text" name="id"/></td>
        </tr>
        <tr>
          <td></td>
          <td><input type="submit" value="REMOVE"/></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

### 3.4 editstudent.html

```
<html>
  <head>
    <title>Edit Student Page</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <h1>Edit student</h1>
    <p>
      Please enter the student details below:
    </p>
    <form action="EditStudentServlet.do" method="POST">
      <table>
        <tr>
          <td>New student age</td>
          <td><input type="text" name="age"/></td>
        </tr>
        <tr>
          <td>Student number:</td>
          <td><input type="text" name="id"/></td>
        </tr>
        <tr>
          <td></td>
          <td><input type="submit" value="EDIT"/></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

### 3.5 getallstudents.html

```
<html>
  <head>
    <title>Get All Students Page</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <h1>Get all students</h1>
    <p>
      Click on the button below to get all the students.
    </p>
    <form action="GetAllStudentsServlet.do" method="GET">
      <table>
        <tr>
          <td></td>
          <td><input type="submit" value="GET ALL STUDENTS"/></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```



#### 4. Servlets

##### 4.1 AddStudentServlet (incomplete)

```
package za.ac.tut.web;

import java.io.IOException;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author Vuyisile Memani
 */
public class AddStudentServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        Long studentNum = Long.parseLong(request.getParameter("id"));
        String name = request.getParameter("name");
        Integer age = Integer.parseInt(request.getParameter("age"));

        //create a student

        //persist the student

        RequestDispatcher disp = request.getRequestDispatcher("addstudent_outcome.jsp");
        disp.forward(request, response);
    }
}
```

#### 4.2 FindStudentServlet (incomplete)

```
package za.ac.tut.web;

import java.io.IOException;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author Vuyisile Memani
 */
public class FindStudentServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        Long studentNum = Long.parseLong(request.getParameter("id"));

        //find student

        RequestDispatcher disp = request.getRequestDispatcher("student_outcome.jsp");
        disp.forward(request, response);
    }
}
```

#### 4.3 RemoveStudentServlet (incomplete)

```
package za.ac.tut.web;

import java.io.IOException;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author Student
 */
public class RemoveStudentServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        Long studentNum = Long.parseLong(request.getParameter("id"));

        //remove student

        RequestDispatcher disp = request.getRequestDispatcher("removestudent_outcome.jsp");
        disp.forward(request, response);
    }
}
```

#### 4.4 EditStudentServlet (incomplete)

```
package za.ac.tut.web;

import java.io.IOException;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author Student
 */
public class EditStudentServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        Long studentNum = Long.parseLong(request.getParameter("id"));
        Integer age = Integer.parseInt(request.getParameter("age"));

        //create a student

        //edit the student

        RequestDispatcher disp = request.getRequestDispatcher("editstudent_outcome.jsp");
        disp.forward(request, response);
    }
}
```

#### 4.5 GetAllStudentsServlet (incomplete)

```
package za.ac.tut.web;

import java.io.IOException;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author Student
 */
public class GetStudentsServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        //get all students

        RequestDispatcher disp = request.getRequestDispatcher("getallstudents_outcome.jsp");
        disp.forward(request, response);
    }
}
```

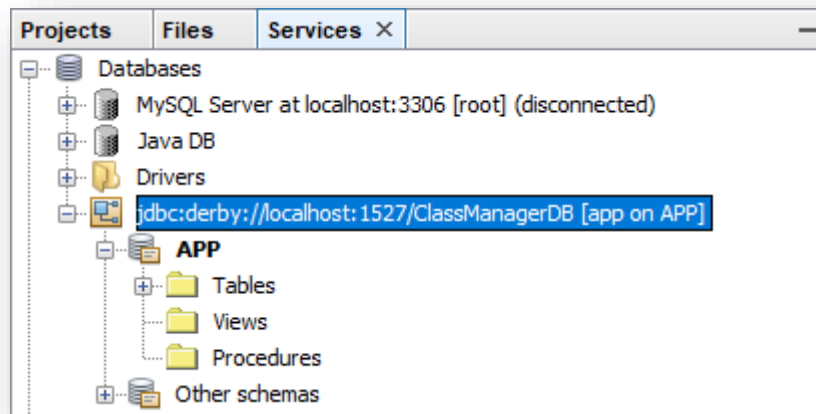
#### 4.6 LogoutServlet

```
/**
 *
 * @author MemaniV
 */
public class LogoutServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        HttpSession session = request.getSession();
        session.invalidate();

        RequestDispatcher disp = request.getRequestDispatcher("index.html");
        disp.forward(request, response);
    }
}
```

5. Create database

- Create a database called **ClassManagerDB**



6. Create connection pools and data source

6.1 Open the Admin Console



6.2 Click on **JDBC | JDBC Connection Pools**. Double click on **DerbyPool**.

The screenshot shows the JBoss AS console interface. On the left is a tree view of the system configuration. The 'Resources' folder is expanded, showing 'JDBC' and 'JDBC Connection Pools'. 'DerbyPool' is selected and highlighted. On the right, the 'Edit JDBC Connection Pool' dialog is open, showing the 'General' tab. The dialog has three tabs: 'General', 'Advanced', and 'Additional Properties'. Below the title bar, there is a description: 'Modify an existing JDBC connection pool. A JDBC connection pool is a group of reusable connections.' and three buttons: 'Load Defaults', 'Flush', and 'Ping'. The 'General Settings' section contains the following fields:

- Pool Name:** DerbyPool
- Resource Type:** javax.sql.DataSource (dropdown menu)
- Datasource Classname:** org.apache.derby.jdbc.ClientDataSource
- Driver Classname:** (empty text field)
- Ping:** ☐ Enabled
- Deployment Order:** 100

Below the 'Deployment Order' field, there is a description: 'Specifies the loading order of the resource at server startup. Lower values are loaded first.'



6.3 Click on the **Additional Properties** tab

GeneralAdvancedAdditional Properties

## Edit JDBC Connection Pool Properties

Modify properties of an existing JDBC connection pool.

Pool Name: DerbyPool

Additional Properties (6)

Add PropertyDelete Properties

Select	Name	Value
<input type="checkbox"/>	PortNumber	1527
<input type="checkbox"/>	Password	APP
<input type="checkbox"/>	User	APP
<input type="checkbox"/>	serverName	localhost
<input type="checkbox"/>	DatabaseName	sun-appserv-samples
<input type="checkbox"/>	connectionAttributes	;create=true

6.4 Click on Add Property, and add url

GeneralAdvancedAdditional Properties

## Edit JDBC Connection Pool Properties

Modify properties of an existing JDBC connection pool.

Pool Name: DerbyPool

Additional Properties (7)

☒ ☐ | Add Property Delete Properties

Select	Name	Value
<input type="checkbox"/>	url	
<input type="checkbox"/>	PortNumber	1527
<input type="checkbox"/>	Password	APP
<input type="checkbox"/>	User	APP
<input type="checkbox"/>	serverName	localhost
<input type="checkbox"/>	DatabaseName	sun-appserv-samples
<input type="checkbox"/>	connectionAttributes	;create=true

6.5 Enter the values.

GeneralAdvancedAdditional Properties

## Edit JDBC Connection Pool Properties

Modify properties of an existing JDBC connection pool.

Pool Name: DerbyPool

Additional Properties (7)

✓

✗

Add Property

Delete Properties

Select	Name	Value
<input type="checkbox"/>	url	dbc:derby://localhost:1527/ClassManagerDB
<input type="checkbox"/>	PortNumber	1527
<input type="checkbox"/>	Password	123
<input type="checkbox"/>	User	app
<input type="checkbox"/>	serverName	localhost
<input type="checkbox"/>	DatabaseName	ClassManagerDB
<input type="checkbox"/>	connectionAttributes	;create=true

6.6 Click on Save and Ping the pool to test if it is working.

GeneralAdvancedAdditional Properties

✓ Ping Succeeded

## Edit JDBC Connection Pool

Modify an existing JDBC connection pool. A JDBC connection pool is a group of reusable connections for a particular database.

Load DefaultsFlushPing

### General Settings

Pool Name:	DerbyPool
Resource Type:	<div>javax.sql.DataSource</div> <div>Must be specified if the datasource class implements more than 1 of the interface.</div>
Datasource Classname:	<div>org.apache.derby.jdbc.ClientDataSource</div> <div>Vendor-specific classname that implements the DataSource and/or XADataSource APIs</div>
Driver Classname:	<div></div> <div>Vendor-specific classname that implements the java.sql.Driver interface.</div>
Ping:	<div><input type="checkbox"/> Enabled</div> <div>When enabled, the pool is pinged during creation or reconfiguration to identify and warn of any erroneous values for its attributes</div>
Deployment Order:	<div>100</div> <div>Specifies the loading order of the resource at server startup. Lower numbers are loaded first.</div>
Description:	<div></div>

6.7 Expand JDBC Resources and click on **jdbc/\_default**

## Edit JDBC Resource

Edit an existing JDBC data source.

[Load Defaults](#)

**JNDI Name:** jdbc/\_default

**Logical JNDI Name:** java:comp/DefaultDataSource

**Pool Name:**

Use the [JDBC Connection Pools](#) page to create new pools

**Deployment Order:**

Specifies the loading order of the resource at server startup. Lower numbers are loaded first.

**Description:**

**Status:** ☒ Enabled

### Additional Properties (0)

[Add Property](#)

[Delete Properties](#)

Select	Name	Value
--------	------	-------

## 6.8 Select **DerbyPool** under Pool name

### Edit JDBC Resource

Edit an existing JDBC data source.

[Load Defaults](#)

**JNDI Name:** jdbc/\_\_default

**Logical JNDI Name:** java:comp/DefaultDataSource

**Pool Name:** DerbyPool ▼  
Use the [JDBC Connection Pools](#) page to create new pools

**Deployment Order:**   
Specifies the loading order of the resource at server startup. Lower numbers are loaded first.

**Description:**


**Status:** ☒ **Enabled**

**Additional Properties (0)**

[Add Property](#) [Delete Properties](#)

Select	Name	Value
--------	------	-------

6.9 Click on **Save**.

 **New values successfully saved.**


## Edit JDBC Resource

Edit an existing JDBC data source.

[Load Defaults](#)

**JNDI Name:** jdbc/\_\_default

**Logical JNDI Name:** java:comp/DefaultDataSource

**Pool Name:**  

Use the [JDBC Connection Pools](#) page to create new pools

**Deployment Order:**

Specifies the loading order of the resource at server startup. Lower numbers are loaded first.

**Description:**

**Status:** ☒ **Enabled**

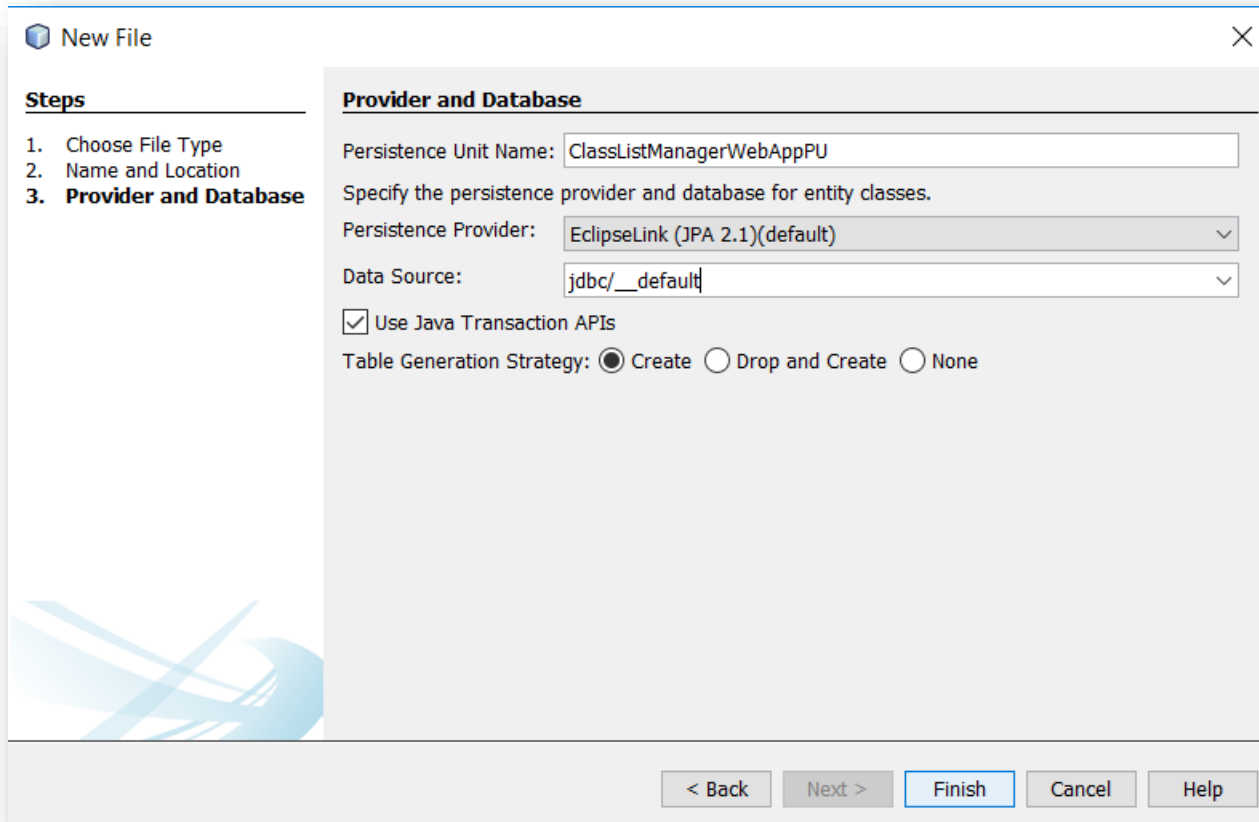
**Additional Properties (0)**

[Add Property](#) [Delete Properties](#)

Select	Name	Value	Description
--------	------	-------	-------------

## 7. Create Entity

- Create an entity called **Student** and choose **jdbc/\_default** as the data source.



The screenshot shows the 'New File' dialog box in the Eclipse IDE. The dialog is titled 'New File' and has a close button (X) in the top right corner. On the left side, there is a 'Steps' panel with three steps: 1. Choose File Type, 2. Name and Location, and 3. **Provider and Database**. The 'Provider and Database' step is currently selected. The main area of the dialog is titled 'Provider and Database' and contains the following fields and options:

- Persistence Unit Name:
- Specify the persistence provider and database for entity classes.
- Persistence Provider:
- Data Source:
- ☒ Use Java Transaction APIs
- Table Generation Strategy: ☒ Create ☐ Drop and Create ☐ None

At the bottom of the dialog, there are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Finish' button is highlighted with a blue border.



```

package za.ac.tut.entities;

import java.io.Serializable;
import java.util.Date;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Temporal;

/**
 *
 * @author Vuyisile Memani
 */
@Entity
public class Student implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    private Long id;
    private String name;
    private Integer age;
    @Temporal(javax.persistence.TemporalType.TIMESTAMP)
    private Date creationDate;

```

8. Create business logic
  - 8.1 StudentFacadeLocal

```
/**
 *
 * @author MemaniV
 */
@Local
public interface StudentFacadeLocal {

    void createStudent(Student student);

    void editStudent(Student student);

    void removeStudent(Student student);

    Student findStudent(Object id);

    List<Student> findAllStudent();
}
```

## 8.2 StudentFacade

```
/**
 *
 * @author MemaniV
 */
@Stateless
public class StudentFacade extends AbstractFacade<Student> implements StudentFacadeLocal {

    @PersistenceContext(unitName = "ClassListManagerWebAppPU")
    private EntityManager em;

    @Override
    protected EntityManager getEntityManager() {
        return em;
    }

    public StudentFacade() {
        super(Student.class);
    }
}
```

```
@Override
public void createStudent(Student student) {
    create(student);
}

@Override
public void editStudent(Student student) {
    edit(student);
}

@Override
public void removeStudent(Student student) {
    //remove the student
    remove(student);
}
```

```
@Override
public Student findStudent(Object id) {
    //find student
    Student targetStudent = find(id);
    return targetStudent;
}
```

```
@Override
public List<Student> findAllStudent() {
    //get the students
    List<Student> list = findAll();

    return list;
}
```

```
}
```

## 9. Modify servlets

### 9.1 AddStudentServlet

```
/**
 *
 * @author MemaniV
 */
public class AddStudentServlet extends HttpServlet {
    @EJB
    private StudentFacadeLocal studentFacade;
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        Long studentNum = Long.parseLong(request.getParameter("id"));
        String name = request.getParameter("name");
        Integer age = Integer.parseInt(request.getParameter("age"));

        Student student = generateStudent(studentNum, name, age);
        studentFacade.createStudent(student);

        RequestDispatcher disp = request.getRequestDispatcher("addstudent_outcome.jsp");
        disp.forward(request, response);
    }
}
```

```
private Student generateStudent(Long studentNum, String name, Integer age) {  
    Student student = new Student();  
    student.setId(studentNum);  
    student.setName(name);  
    student.setAge(age);  
    student.setCreationDate(new Date());  
    return student;  
}  
  
}
```

## 9.2 FindStudentServlet

```
/**
 *
 * @author MemaniV
 */
public class FindStudentServlet extends HttpServlet {

    @EJB
    private StudentFacadeLocal studentFacade;
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        Long studentNum = Long.parseLong(request.getParameter("id"));

        Student student = studentFacade.findStudent(studentNum);
        request.setAttribute("student", student);

        RequestDispatcher disp = request.getRequestDispatcher("findstudent_outcome.jsp");
        disp.forward(request, response);
    }
}
```



### 9.3 RemoveStudentServlet

```
/**
 *
 * @author MemaniV
 */
public class RemoveStudentServlet extends HttpServlet {

    @EJB
    private StudentFacadeLocal studentFacade;
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        Long studentNum = Long.parseLong(request.getParameter("id"));

        Student student = generateStudent(studentNum);
        studentFacade.removeStudent(student);

        RequestDispatcher disp = request.getRequestDispatcher("removestudent_outcome.jsp");
        disp.forward(request, response);
    }
}
```

```
private Student generateStudent(Long studentNum) {  
    Student student = new Student();  
    student.setId(studentNum);  
    return student;  
}  
  
}
```

#### 9.4 EditStudentServlet

```
public class EditStudentServlet extends HttpServlet {
    @EJB
    private StudentFacadeLocal studentFacade;
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        Long studentNum = Long.parseLong(request.getParameter("id"));
        Integer age = Integer.parseInt(request.getParameter("age"));
        String location = "editstudent_outcome.jsp";

        Student targetStudent = studentFacade.findStudent(studentNum);

        if(targetStudent == null){ //check if student is found
            location = "studentnotfound.jsp";
        } else {
            Student student = modifyStudent(targetStudent, age);
            studentFacade.editStudent(student);
        }

        RequestDispatcher disp = request.getRequestDispatcher(location);
        disp.forward(request, response);
    }
}
```

```
private Student modifyStudent(Student targetStudent, Integer age) {  
    targetStudent.setAge(age);  
    return targetStudent;  
}  
  
}
```

## 9.5 GetAllStudentsServlet

```
/**
 *
 * @author MemaniV
 */
public class GetAllStudentsServlet extends HttpServlet {

    @EJB
    private StudentFacadeLocal studentFacade;
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        //get all students
        List<Student> students = studentFacade.findAllStudent();
        request.setAttribute("students", students);

        RequestDispatcher disp = request.getRequestDispatcher("getallstudents_outcome.jsp");
        disp.forward(request, response);
    }
}
```

## 10. JSPs

### 10.1 addstudent\_outcome

```
<%--
    Document    : addstudent_outcome
    Created on  : 27 May 2022, 4:12:37 AM
    Author      : MemaniV
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Add Student Outcome Page</title>
    </head>
    <body>
        <h1>Add student outcome</h1>
        <p>
            The student has been successfully added. Please click <a href="menu.html">here</a>
            Click <a href="LogoutServlet.do">here</a> to logout.
        </p>
    </body>
</html>
```

## 10.2 findstudent\_outcome

```
<%--
    Document      : findstudent_outcome
    Created on    : 27 May 2022, 4:21:48 AM
    Author       : MemaniV
--%>

<%@page import="za.ac.tut.entities.Student"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Find Student Outcome age</title>
    </head>
    <body>
        <h1>Find student outcome</h1>
        <%
            Student student = (Student)request.getAttribute("student");
            String name = student.getName();
            Long studentNum = student.getId();
            Integer age = student.getAge();
        %>
        <p>
            The student has been found. Below are the details of the student.
        </p>
```

```
<table>
  <tr>
    <td><b>Name</b>: </td>
    <td><%=name%></td>
  </tr>
  <tr>
    <td><b>Age</b>: </td>
    <td><%=age%></td>
  </tr>
  <tr>
    <td><b>Student number</b>: </td>
    <td><%=studentNum%></td>
  </tr>
</table>
<p>
  Click <a href="menu.html">here</a> to go back to the menu page.<br>
  Click <a href="LogoutServlet.do">here</a> to logout.
</p>
</body>
</html>
```



### 10.3 removestudent\_outcome

```
<%--
    Document    : removestudent_outcome
    Created on  : 27 May 2022, 4:31:27 AM
    Author      : MemaniV
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Remove Student Outcome Page</title>
    </head>
    <body>
        <h1>Remove student outcome</h1>
        <p>
            The student has been successfully removed. Please click <a href="menu.html">here</a>
            Click <a href="LogoutServlet.do">here</a> to logout.
        </p>
    </body>
</html>
```

#### 10.4 editstudent\_outcome

```
<%--
    Document    : editstudent_outcome
    Created on  : 27 May 2022, 5:07:03 AM
    Author      : MemaniV
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Edit Student Page</title>
    </head>
    <body>
        <h1>Edit student outcome</h1>
        <p>
            The student has been successfully edited. Click <a href="menu.html">here</a>
            Click <a href="LogoutServlet.do">here</a> to logout.
        </p>
    </body>
</html>
```

## 10.5 getallstudents\_outcome

```
<%--
    Document    : getallstudents_outcome
    Created on  : 27 May 2022, 5:16:59 AM
    Author      : MemaniV
--%>

<%@page import="java.util.List"%>
<%@page import="za.ac.tut.entities.Student"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>View Students Page</title>
    </head>
    <body>
        <h1>Class list</h1>
        <%
            List<Student> students = (List<Student>)request.getAttribute("students");
        %>
```

```

<table>
<%
    for(int i = 0; i < students.size(); i++){
        Student student = students.get(i);
        String name = student.getName();
        Long studentNum = student.getId();
        Integer age = student.getAge();
    %>
    <tr>
        <td><b>Name</b>: </td>
        <td><%=name%></td>
    </tr>
    <tr>
        <td><b>Age</b>: </td>
        <td><%=age%></td>
    </tr>
    <tr>
        <td><b>Student number</b>: </td>
        <td><%=studentNum%></td>
    </tr>
    <tr></tr>
<%
}

```

```
%>
</table>
<p>
Click <a href="menu.html">here</a> to go back to the menu page.<br>
Click <a href="LogoutServlet.do">here</a> to logout.
</p>
</body>
</html>
```

```
<%--
    Document      : studentfound
    Created on    : 27 May 2022, 5:05:11 AM
    Author       : MemaniV
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Student Found Page</title>
    </head>
    <body>
        <h1>Student found</h1>
        <p>
            The student has been found. Please click <a href="menu.html">here</a>
            Click <a href="LogoutServlet.do">here</a> to logout.
        </p>
    </body>
</html>
```

## 10.7 studentnotfound

```
<%--
    Document      : studentnotfound
    Created on    : 27 May 2022, 5:09:51 AM
    Author       : MemaniV
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Student Not Found Page</title>
    </head>
    <body>
        <h1>Student not found</h1>
        <p>
            The student was not found. Please click <a href="menu.html">here</a>
            Click <a href="LogoutServlet.do">here</a> to logout.
        </p>
    </body>
</html>
```

## B. Test the web application

1. Add students into the database

### Add student

Please enter the details of the student below or [here](#) to go back to the menu page:

Name:	<input type="text" value="Vuyisile Memani"/>
Age	<input type="text" value="12"/>
Student number:	<input type="text" value="111"/>
<input type="button" value="ADD"/>	



[illegible]

2. Remove student from database

## Remove student


















































Please enter the student number below:

Student number:

REMOVE

## Remove student outcome

The student has been successfully removed. Please click [here](#) to go back to the menu page.

select * from APP.STUDENT ×				
                                                				
Page Size: 20   Total Rows: 4   Page: 1 of 1				
#	ID	AGE	CREATIONDATE	NAME
1	111	12	2022-05-27 04:16:41.885	Vuyisile Memani
2	222	20	2022-05-27 04:16:57.076	Thandeka
3	333	15	2022-05-27 04:17:17.079	Thato
4	444	12	2022-05-27 04:17:29.816	Khomo

### 3. Edit student

## Edit student










Please enter the student details below:

New student age

Student number:

## Edit student outcome

The student has been successfully edited. Click [here](#) to go back to the main page.

select * from APP.STUDENT ×				
         Page Size: <input type="text" value="20"/>   Total Rows: 4   Page: 1 of 1				
#	ID	AGE	CREATIONDATE	NAME
1	111	100	2022-05-27 04:16:41.885	Vuyisile Memani
2	222	20	2022-05-27 04:16:57.076	Thandeka
3	333	15	2022-05-27 04:17:17.079	Thato
4	444	12	2022-05-27 04:17:29.816	Khomo

4. Find student

## Find student

Please enter the student number below:

Student number:

Edit JDBC Resource  
<http://localhost:4848/common/index.jsf>

# elcome

The student has been found. Below are the details of the student.

**Name:** Vuyisile Memani

**Age:** 100

**Student number:** 111

Click [here](#) to go back to the menu page.

5. Get all students

## Get all students

Click on the button below to get all the students.

GET ALL STUDENTS



## Class list

**Name:** Vuyisile Memani

**Age:** 100

**Student number:** 111

**Name:** Thandeka

**Age:** 20

**Student number:** 222

**Name:** Thato

**Age:** 15

**Student number:** 333

**Name:** Khomo

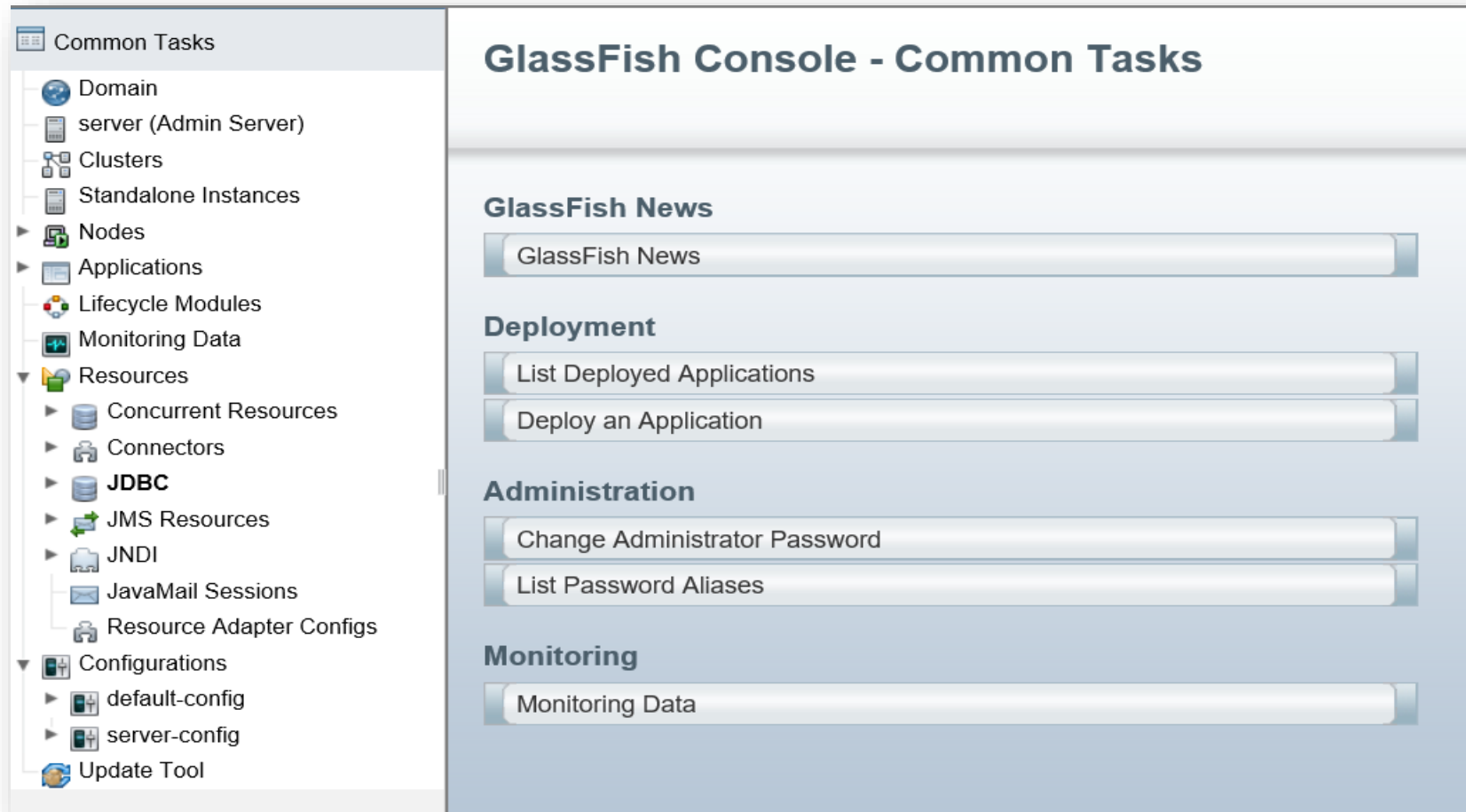
**Age:** 12

**Student number:** 444

Click [here](#) to go back to the menu page.

## C. Add security measures

1. Define roles in GlassFish
  - 1.1 Open the Admin Console



1.2 Under Common Tasks, make the following expansions: **Configurations** | **server-config** | **Security** | **Realms**

The screenshot shows the JBoss configuration console. On the left, a tree view is expanded to 'Configurations' > 'server-config' > 'Security' > 'Realms'. The main panel on the right is titled 'Realms' and contains the following information:

Create, modify, or delete security (authentication) realms.

**Configuration Name:** server-config

**Realms (3)**

Buttons: ☒ ☐ | [New...](#) [Delete](#)

Select	Name	Class Name
<input type="checkbox"/>	<a href="#">admin-realm</a>	com.sun.enterprise.security.auth.realm.file.FileRealm
<input type="checkbox"/>	<a href="#">certificate</a>	com.sun.enterprise.security.auth.realm.certificate.CertificateRealm
<input type="checkbox"/>	<a href="#">file</a>	com.sun.enterprise.security.auth.realm.file.FileRealm

1.3 Under Realms, click on **New**.

New Realm

OKCancel

Create a new security (authentication) realm. Valid realm types are PAM, OSGi, File, Certificate, LDAP, JDBC, Digest, Oracle Solaris, and Custom.

\* Indicates required field

Configuration Name: server-config

Name: \*

Class Name: ☒ com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm ☐

Choose a realm class name from the drop-down list or specify a custom class

Properties specific to this Class

JAAS Context: \*

Identifier for the login module to use for this realm

JNDI: \*

JNDI name of the JDBC resource used by this realm

#### 1.4 Fill-in the form.

### Edit Realm

SaveCancel

Edit an existing security (authentication) realm.

Manage Users

\* Indicates required field

**Configuration Name:** server-config

---

**Realm Name:** FRealm

**Class Name:** com.sun.enterprise.security.auth.realm.file.FileRealm

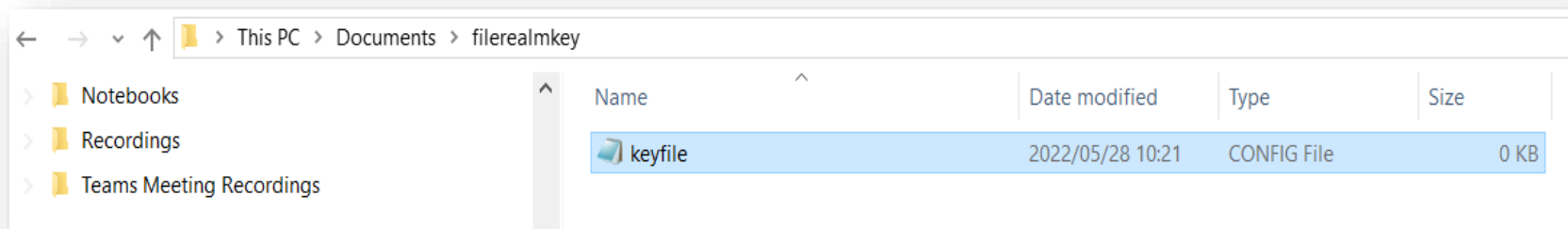
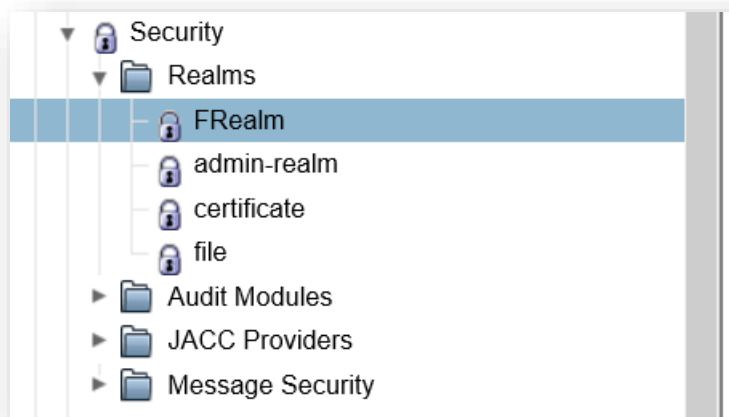
**Properties specific to this Class**

**JAAS Context:** \* fileRealm  
Identifier for the login module to use for this realm

**Key File:** \* C:\Users\memaniv\Documents\filerealmkey\keyfile.config  
Full path and name of the file where the server will store all user, group, and password information for this realm

**Assign Groups:** default  
Comma-separated list of group names

1.5 Click OK and restart (close and open again) the Admin Console



- 1.6 Create users accounts
  - 1.6.1 Click on FRealm

## Edit Realm

Edit an existing security (authentication) realm.

[Manage Users](#)

**Configuration Name:** server-config

**Realm Name:** FRealm

**Class Name:** com.sun.enterprise.security.auth.realm.file.FileRealm

### Properties specific to this Class

**JAAS Context: \***

Identifier for the login module to use for this realm

**Key File: \***

Full path and name of the file where the server will store all user, group, and password information for this realm

**Assign Groups:**

Comma-separated list of group names

1.6.2 Click on **Manage Users**.

## File Users

Manage user accounts for the currently selected security realm.

**Configuration Name:** server-config

**Realm Name:** FRealm

File Users (0)

New...

Delete

Select	User ID	Group List:
No items found.		



1.6.3 Click on **New**.

## New File Realm User

Create new user accounts for the currently selected security realm.

**Configuration Name:** server-config

**Realm Name:** FRealm

**User ID:** \*

Name can be up to 255 characters, must contain only letters, digits, underscore, dash, or dot characters

**Group List:**

Separate multiple groups with colon

**New Password:**

**Confirm New Password:**

1.6.4 Fill-in the form.

## New File Realm User

Create new user accounts for the currently selected security realm.

**Configuration Name:** server-config

---

<b>Realm Name:</b>	FRealm
<b>User ID:</b> *	<input type="text" value="vuyi"/> <small>Name can be up to 255 characters, must contain only letters, digits, underscore, dash, or dot characters</small>
<b>Group List:</b>	<input type="text" value="teacher"/> <small>Separate multiple groups with colon</small>
<b>New Password:</b>	<input type="password" value="..."/>
<b>Confirm New Password:</b>	<input type="password" value="..."/>

The password is **123**

1.6.5 Click **OK**

## File Users

Manage user accounts for the currently selected security realm.

**Configuration Name:** server-config

**Realm Name:** FRealm

File Users (1)

New...

Delete

Select	User ID	Group List:
<input type="checkbox"/>	vuyi	<div>teacher</div> <div>default</div>

1.6.6 Add a second user. Click on **New** and fill-in the form.

## New File Realm User

Create new user accounts for the currently selected security realm.

**Configuration Name:** server-config

---

<b>Realm Name:</b>	FRealm
<b>User ID:</b> *	<input type="text" value="khomo"/> <small>Name can be up to 255 characters, must contain only letters, digits, underscore, dash, or dot characters</small>
<b>Group List:</b>	<input type="text" value="student"/> <small>Separate multiple groups with colon</small>
<b>New Password:</b>	<input type="password" value="..."/>
<b>Confirm New Password:</b>	<input type="password" value="..."/>

The password is **456**

1.6.7 Click **OK**.

### File Users

Manage user accounts for the currently selected security realm.

**Configuration Name:** server-config

**Realm Name:** FRealm

File Users (2)		
<a href="#">New...</a> <a href="#">Delete</a>		
Select	User ID	Group List:
<input type="checkbox"/>	khomo	<div>student</div> <div>default</div>
<input type="checkbox"/>	vuyi	<div>teacher</div> <div>default</div>

```
1 khomo;{SSHA256}8fGDUieDPd7r0ZOoUBTQq0rcLi9yaHm2GByx6g+4kDEfkmJN06gOhg==;student
2 vuyi;{SSHA256}4qFRaHV9eSoZFT8VkgDcfiob9gOh9B3Jl4KQXGLyVGd4HAsDbLMy2g==;teacher
3
```

2. Go back to the web application.

2.1 Create a login page

```
<%--
    Document      : login
    Created on    : 28 May 2022, 10:43:25 AM
    Author       : MemaniV
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Login Page</title>
    </head>
    <body>
        <h1>Login</h1>
        <p>
            Please enter your login details below:
        </p>
```

```
<form action="j_security_check" method="POST">
  <table>
    <tr>
      <td>Username:</td>
      <td><input type="text" name="j_username" required=""/></td>
    </tr>
    <tr>
      <td>Password:</td>
      <td><input type="text" name="j_password" required=""/></td>
    </tr>
    <tr>
      <td></td>
      <td><input type="submit" value="LOGIN"/></td>
    </tr>
  </table>
</form>
</body>
</html>
```

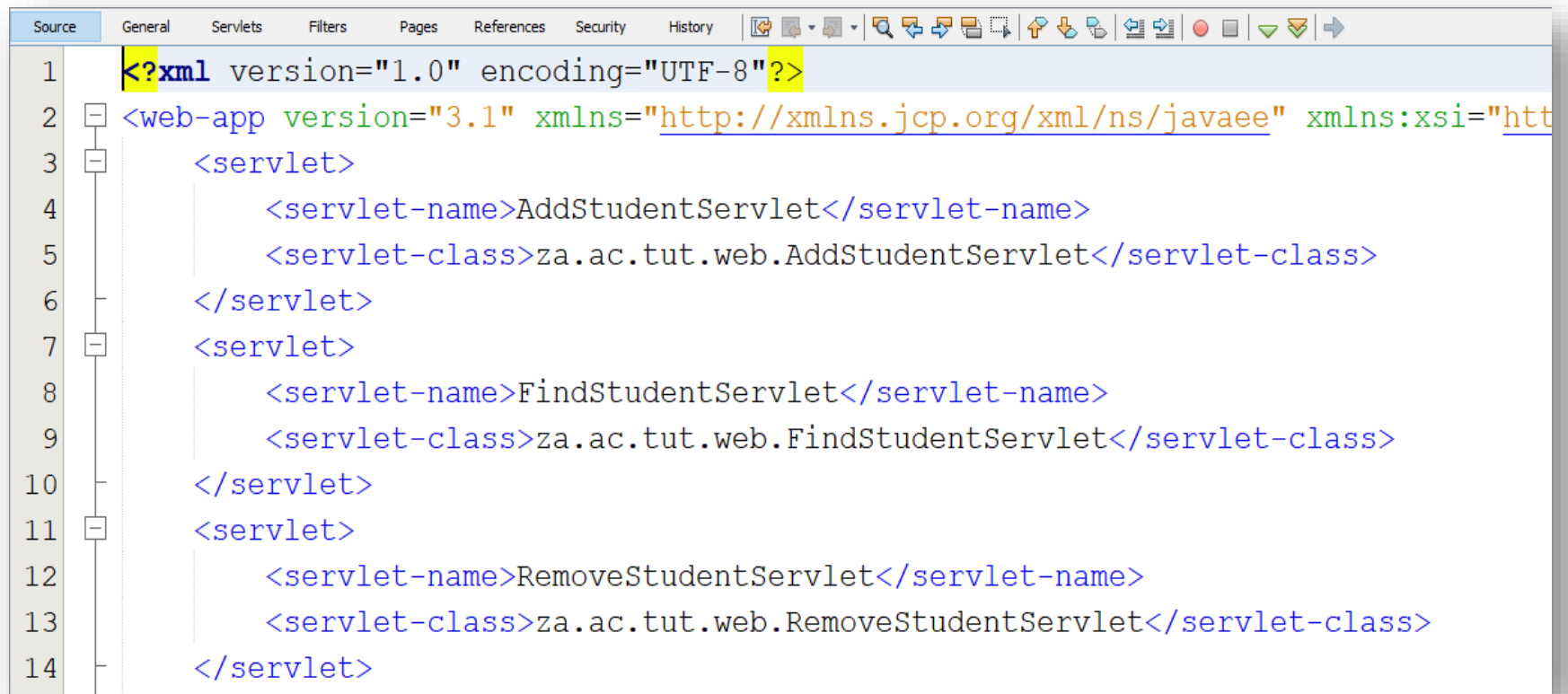
## 2.2 Create a login error page

```
<%--
    Document    : loginerror
    Created on  : 28 May 2022, 10:51:15 AM
    Author      : MemaniV
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Login Error Page</title>
    </head>
    <body>
        <h1>Login error</h1>
        <p>
            Your login details are invalid. Please click <a href="menu.html">here</a>
        </p>
    </body>
</html>
```

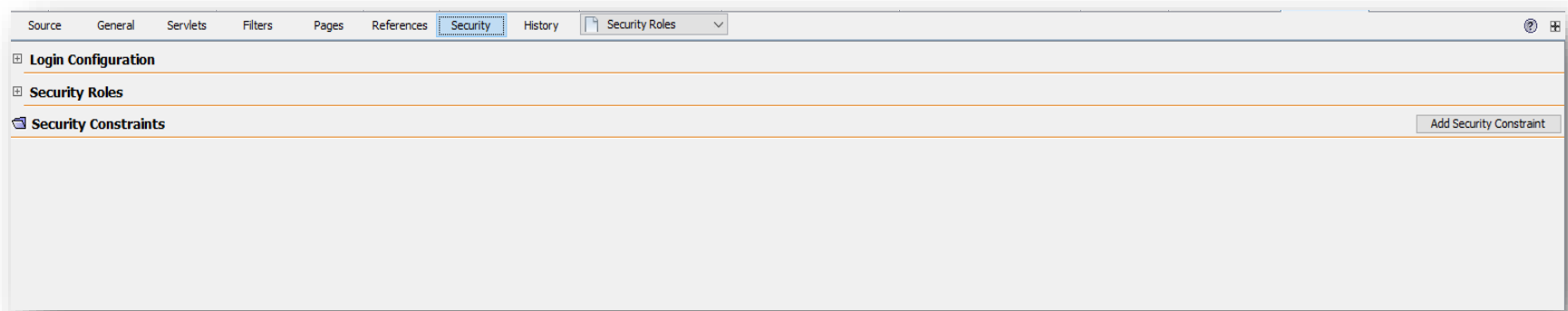


### 2.3 Open the Deployment Descriptor (DD file→ web.xml)

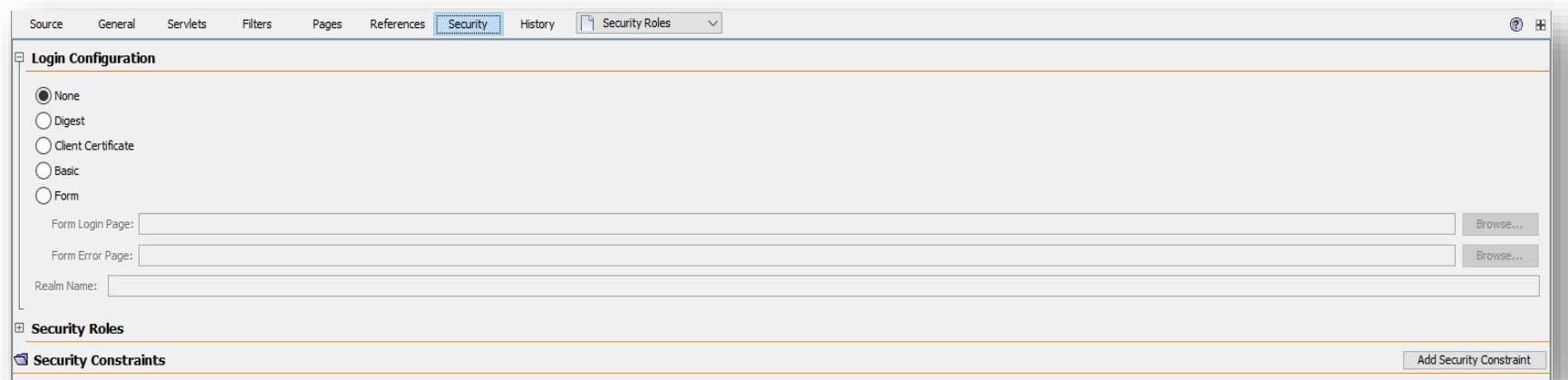


```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://xmlns.jcp.org/xml/ns/xsi" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">
3     <servlet>
4         <servlet-name>AddStudentServlet</servlet-name>
5         <servlet-class>za.ac.tut.web.AddStudentServlet</servlet-class>
6     </servlet>
7     <servlet>
8         <servlet-name>FindStudentServlet</servlet-name>
9         <servlet-class>za.ac.tut.web.FindStudentServlet</servlet-class>
10    </servlet>
11    <servlet>
12        <servlet-name>RemoveStudentServlet</servlet-name>
13        <servlet-class>za.ac.tut.web.RemoveStudentServlet</servlet-class>
14    </servlet>
```

2.4 Click on the Security tab.



2.5 Expand Login Configuration



## 2.6 Select Form

The screenshot shows a 'Login Configuration' dialog box with the following elements:

- Tabbed Interface:** Source, General, Servlets, Filters, Pages, References, **Security**, History. A dropdown menu shows 'Login Configuration'.
- Login Configuration Section:**
  - Radio buttons for authentication methods: None, Digest, Client Certificate, Basic, and **Form** (selected).
  - Form Login Page:  Browse...
  - Form Error Page:  Browse...
  - Realm Name:
- Security Roles Section:** A section header with a small icon.
- Security Constraints Section:** A section header with a small icon and an Add Security Constraint button.

2.7 Locate the login and error page by browsing to them.

The screenshot shows a 'Login Configuration' dialog box with the following elements:

- Login Configuration** section:
  - Radio buttons for authentication methods: None, Digest, Client Certificate, Basic, and **Form** (selected).
  - Form Login Page:  with a **Browse...** button.
  - Form Error Page:  with a **Browse...** button.
  - Realm Name:
- Security Roles** section: A collapsed section indicated by a minus sign icon.
- Security Constraints** section: A collapsed section indicated by a minus sign icon, with an **Add Security Constraint** button.

2.8 Our authentication realm is called **FRealm**. Write it down under **Realm Name**.

Source General Servlets Filters Pages References **Security** History Login Configuration ▾

**Login Configuration**

☐ None  
☐ Digest  
☐ Client Certificate  
☐ Basic  
☒ Form

Form Login Page:  Browse...

Form Error Page:  Browse...

Realm Name:

**Security Roles**

**Security Constraints** Add Security Constraint

2.9 Minimize Login Configuration.

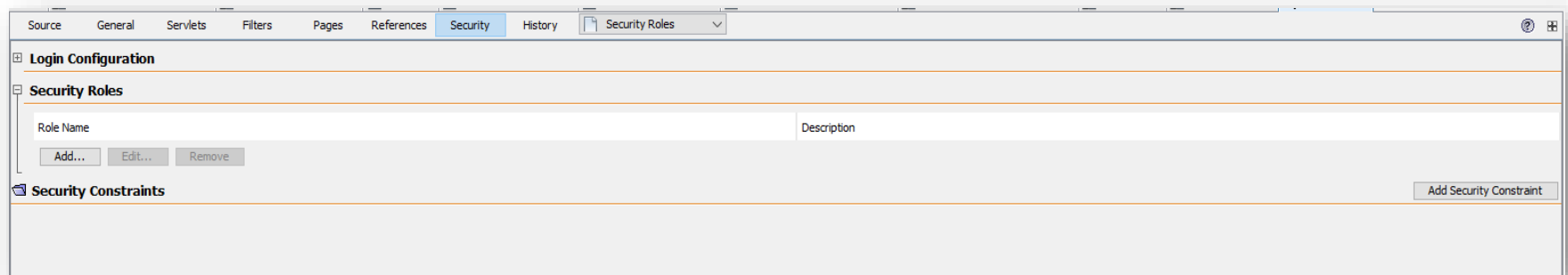
Source General Servlets Filters Pages References **Security** History Security Roles ▾

**Login Configuration**

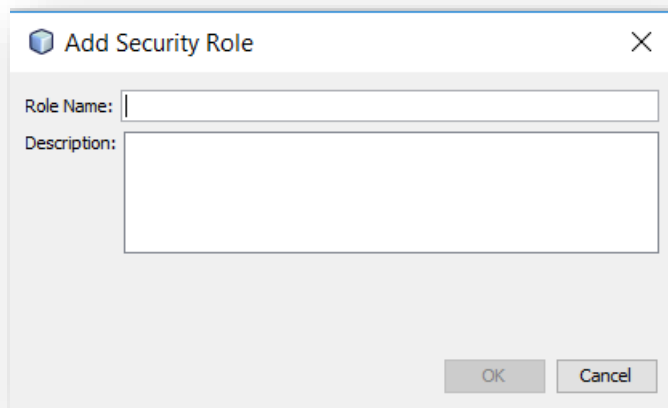
**Security Roles**

**Security Constraints** Add Security Constraint

## 2.10 Expand Security Roles



## 2.11 Click on **Add**



## 2.12 Fill in the form

The screenshot shows the 'Security Roles' configuration window. The 'Security Roles' section is expanded, displaying a table with two columns: 'Role Name' and 'Description'. The 'teacher' role is listed with the description 'A teacher can add, remove, edit, find, and view all students in a class.' Below the table are three buttons: 'Add...', 'Edit...', and 'Remove'.

Role Name	Description
teacher	A teacher can add, remove, edit, find, and view all students in a class.

Buttons: Add... Edit... Remove

## 2.13 Add a second role, a student.

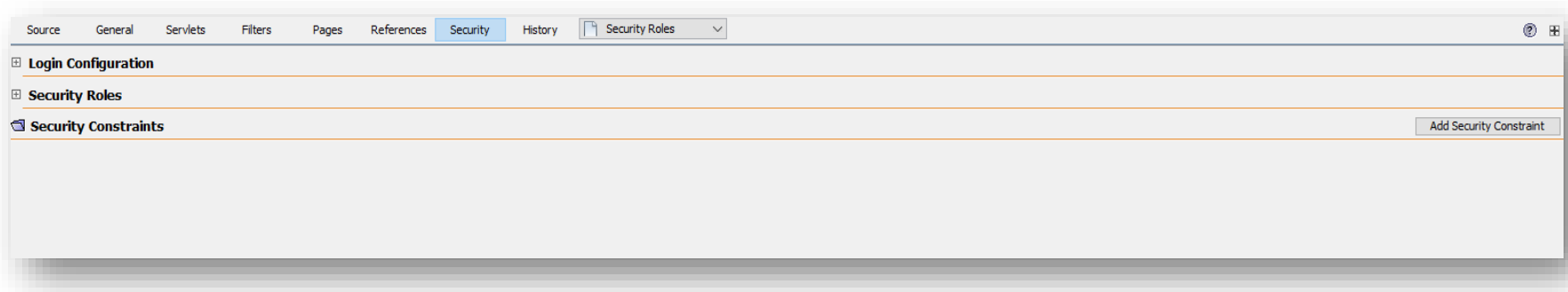
The screenshot shows the 'Security Roles' configuration window after adding a second role. The 'Security Roles' section is expanded, displaying a table with two columns: 'Role Name' and 'Description'. The 'teacher' role is listed with the description 'A teacher can add, remove, edit, find, and view all students in a class.' The 'student' role is listed with the description 'A student can only view all the students in a class.' Below the table are three buttons: 'Add...', 'Edit...', and 'Remove'.

Role Name	Description
teacher	A teacher can add, remove, edit, find, and view all students in a class.
student	A student can only view all the students in a class.

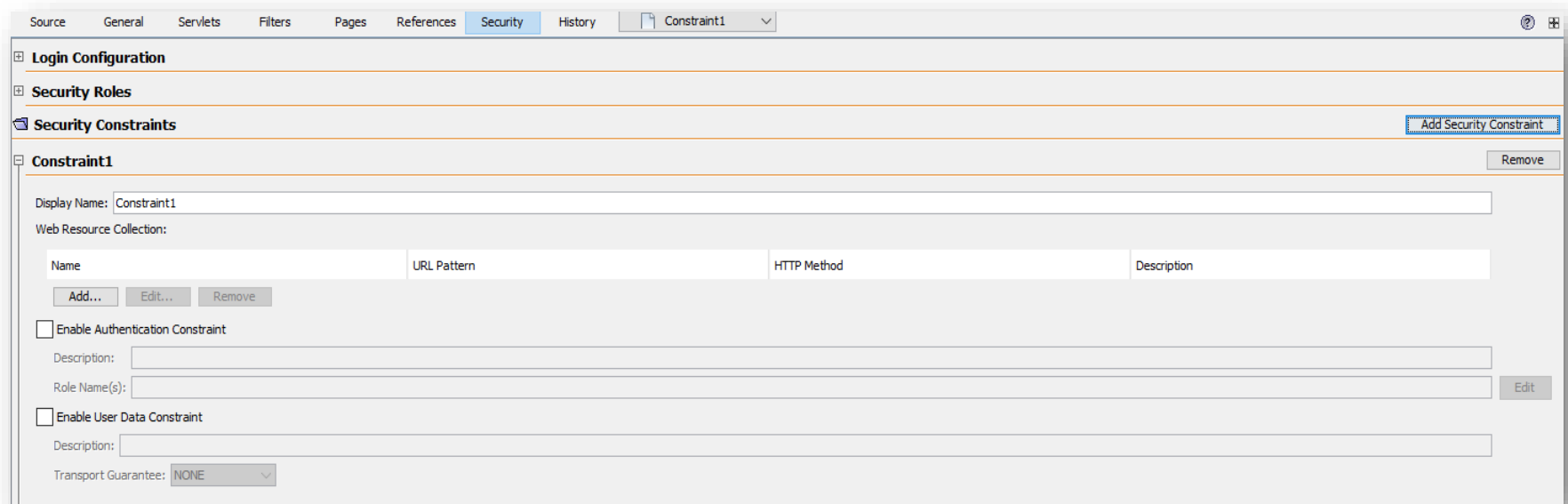
Buttons: Add... Edit... Remove

Close the DD file.

## 2.14 Minimize Security Roles.



## 2.15 Add security constraints. Click on the Add Security Constraint button.





2.16 Fill in the form.

2.16.1 Name the constraint as TeacherConstraint.

**Security Constraints**

**TeacherConstraint**

Display Name:

Web Resource Collection:

Name	URL Pattern	HTTP Method	Description
<input type="button" value="Add..."/>	<input type="button" value="Edit..."/>	<input type="button" value="Remove"/>	

☐ Enable Authentication Constraint

Description:

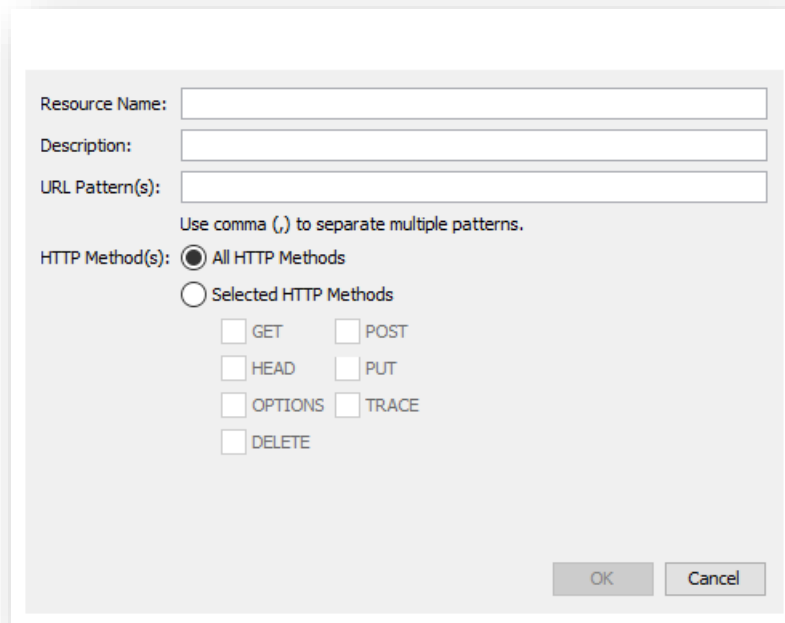
Role Name(s):

☐ Enable User Data Constraint

Description:

Transport Guarantee:

### 2.16.2 Click on Add



A screenshot of a dialog box with a light gray background and a white border. The dialog box contains the following elements:

- Resource Name:** A text input field.
- Description:** A text input field.
- URL Pattern(s):** A text input field.
- Use comma (,) to separate multiple patterns.** A small text label below the URL Pattern(s) field.
- HTTP Method(s):** A section with two radio buttons:
  - ☒ **All HTTP Methods**
  - ☐ **Selected HTTP Methods**
- Selected HTTP Methods:** A group of six checkboxes arranged in two columns:
  - ☐ GET
  - ☐ POST
  - ☐ HEAD
  - ☐ PUT
  - ☐ OPTIONS
  - ☐ TRACE
  - ☐ DELETE
- OK** and **Cancel** buttons at the bottom right.

### 2.16.3 Complete the form.

Resource Name:

Description:

URL Pattern(s):

Use comma (,) to separate multiple patterns.

HTTP Method(s): ☒ All HTTP Methods  
☐ Selected HTTP Methods

☐ GET ☐ POST  
☐ HEAD ☐ PUT  
☐ OPTIONS ☐ TRACE  
☐ DELETE

#### 2.16.4 Click OK

**Login Configuration**

**Security Roles**

**Security Constraints** Add Security Constraint

**TeacherConstraint** Remove

Display Name:

Web Resource Collection:

Name	URL Pattern	HTTP Method	Description
TeacherResources	/addstudent.html, /removestudent.html, findstudent.html, getallst...		Resources available to a teacher

Add... Edit... Remove

☐ **Enable Authentication Constraint**

Description:

Role Name(s):  Edit

☐ **Enable User Data Constraint**

Description:

Transport Guarantee: NONE ▼

### 2.16.5 Click on the Enable Authentication Constraint.

TeacherConstraint

Remove

Display Name: TeacherConstraint

Web Resource Collection:

Name	URL Pattern	HTTP Method	Description
TeacherResources	addstudent.html, removestudent.html, findstudent.html, getallstud...		Resources available to a teacher

Add...

Edit...

Remove

☒ Enable Authentication Constraint

Description: Enable authentication of teacher

Role Name(s):

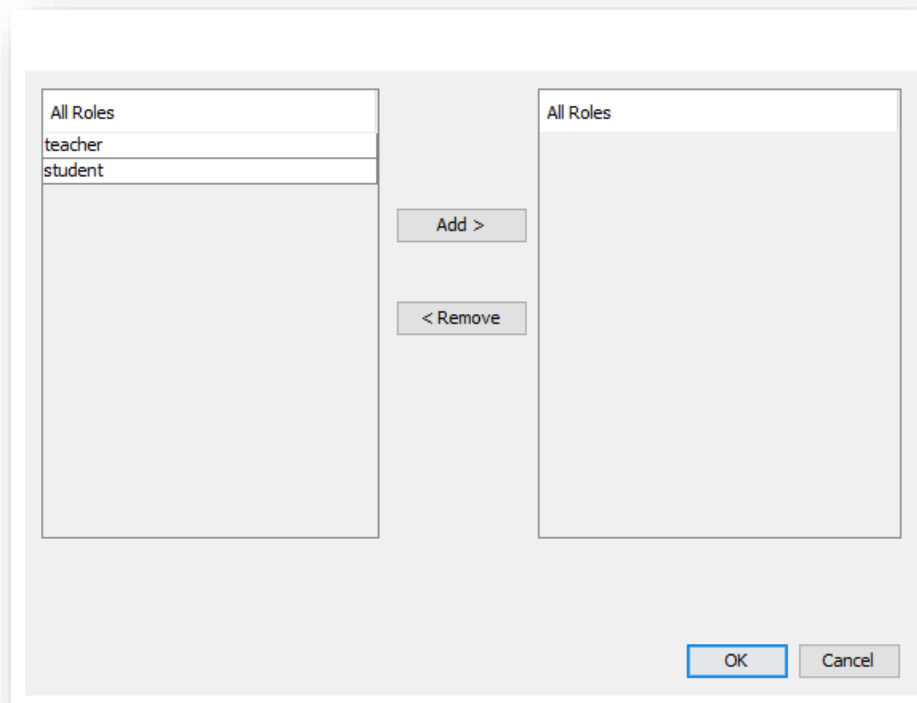
Edit

☐ Enable User Data Constraint

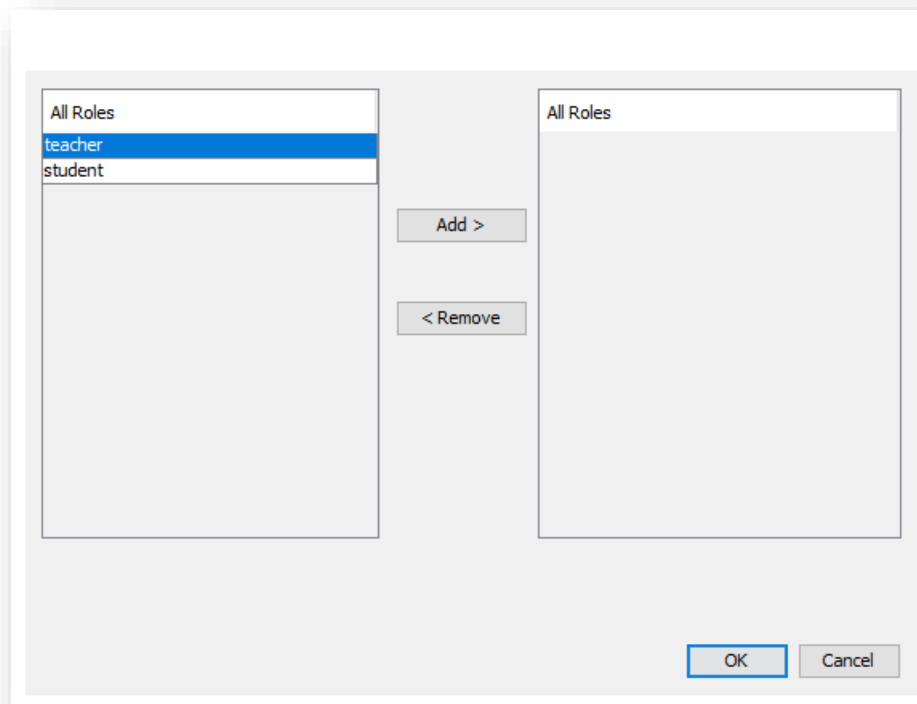
Description:

Transport Guarantee: NONE

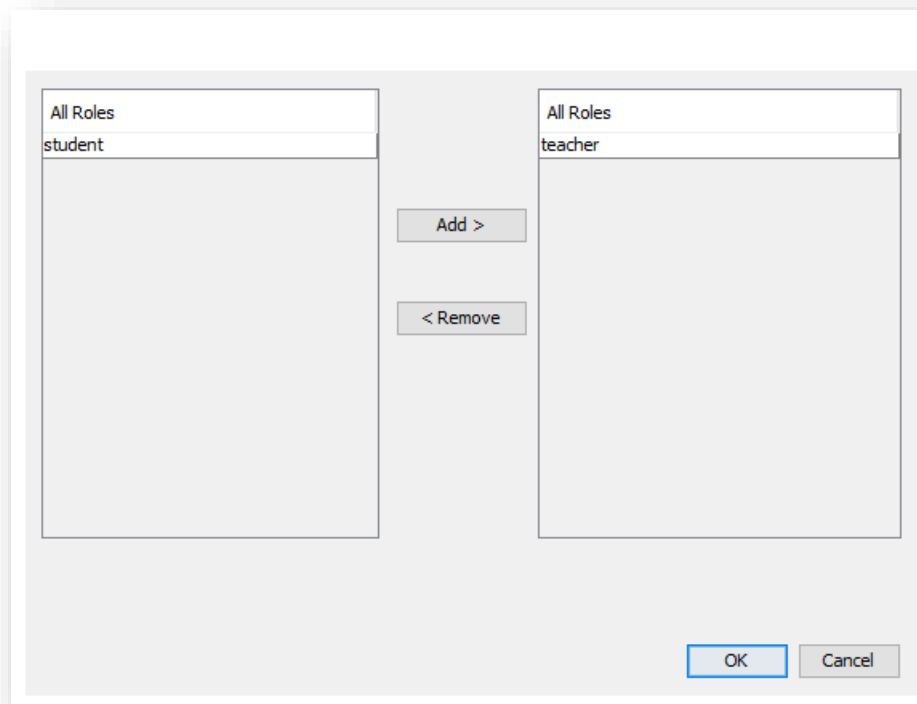
#### 2.16.6 Click on the Edit button



2.16.7 Select teacher under All Roles.



#### 2.16.8 Click on Add





## 2.16.9 Click OK

The screenshot shows the 'Security Constraints' dialog box with the 'TeacherConstraint' tab selected. The 'Display Name' is 'TeacherConstraint'. Under 'Web Resource Collection', there is a table with one entry: 'TeacherResources' with a URL pattern of 'addstudent.html, removestudent.html, findstudent.html, getallstud...' and a description of 'Resources available to a teacher'. Below the table are 'Add...', 'Edit...', and 'Remove' buttons. The 'Enable Authentication Constraint' checkbox is checked, with a description of 'Enable authentication of teacher' and a role name of 'teacher'. The 'Enable User Data Constraint' checkbox is unchecked. The 'Transport Guarantee' is set to 'NONE'. An 'Edit' button is visible on the right side of the dialog.

**Security Constraints** Add Security Constraint

**TeacherConstraint** Remove

Display Name: TeacherConstraint

Web Resource Collection:

Name	URL Pattern	HTTP Method	Description
TeacherResources	addstudent.html, removestudent.html, findstudent.html, getallstud...		Resources available to a teacher

Add... Edit... Remove

☒ Enable Authentication Constraint

Description: Enable authentication of teacher

Role Name(s): teacher Edit

☐ Enable User Data Constraint

Description:

Transport Guarantee: NONE

2.17 Add a constraint for the student. Click on Add Security Constraint and fill in the form.

Security Constraints

Add Security Constraint

TeacherConstraint

Remove

StudentConstraint

Remove

Display Name: StudentConstraint

Web Resource Collection:

Name	URL Pattern	HTTP Method	Description
StudentResources	/getallstudents.html		The resources available to a student

Add...

Edit...

Remove

☒ Enable Authentication Constraint

Description: Enable authentication of student

Role Name(s): student

Edit

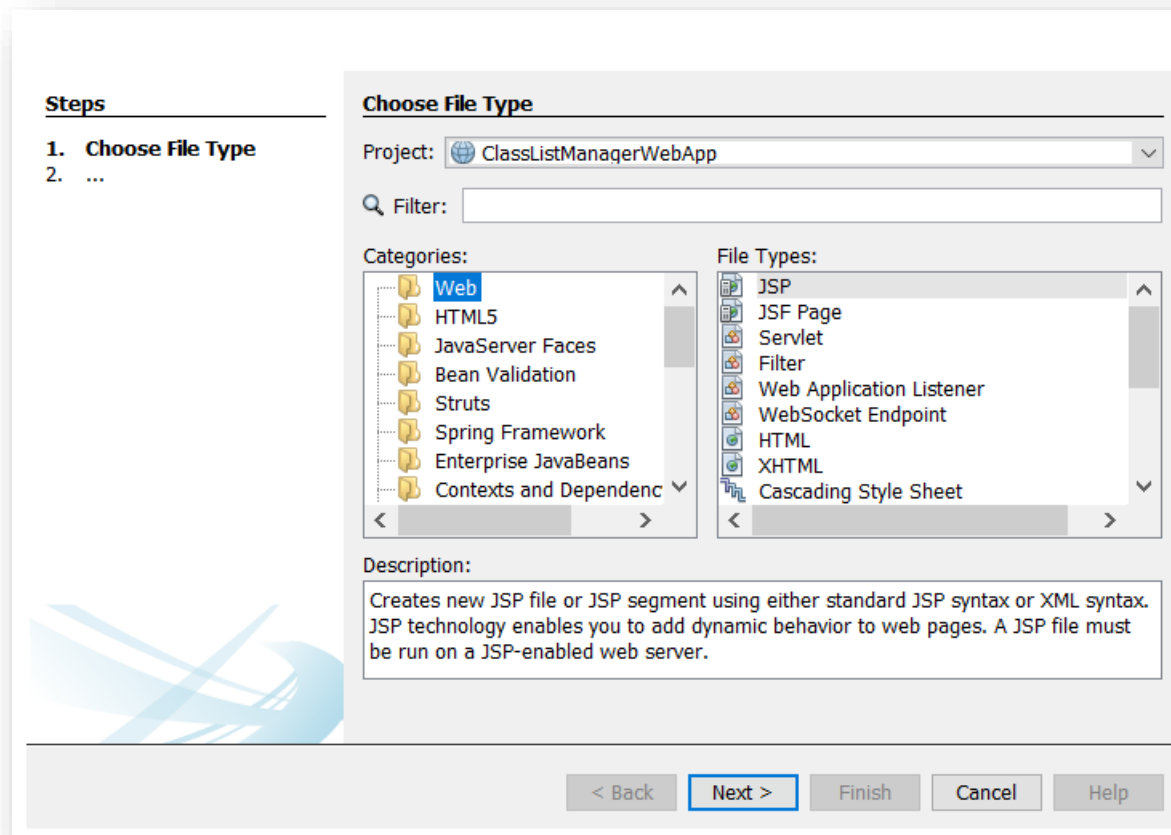
☐ Enable User Data Constraint

Description:

Transport Guarantee: NONE

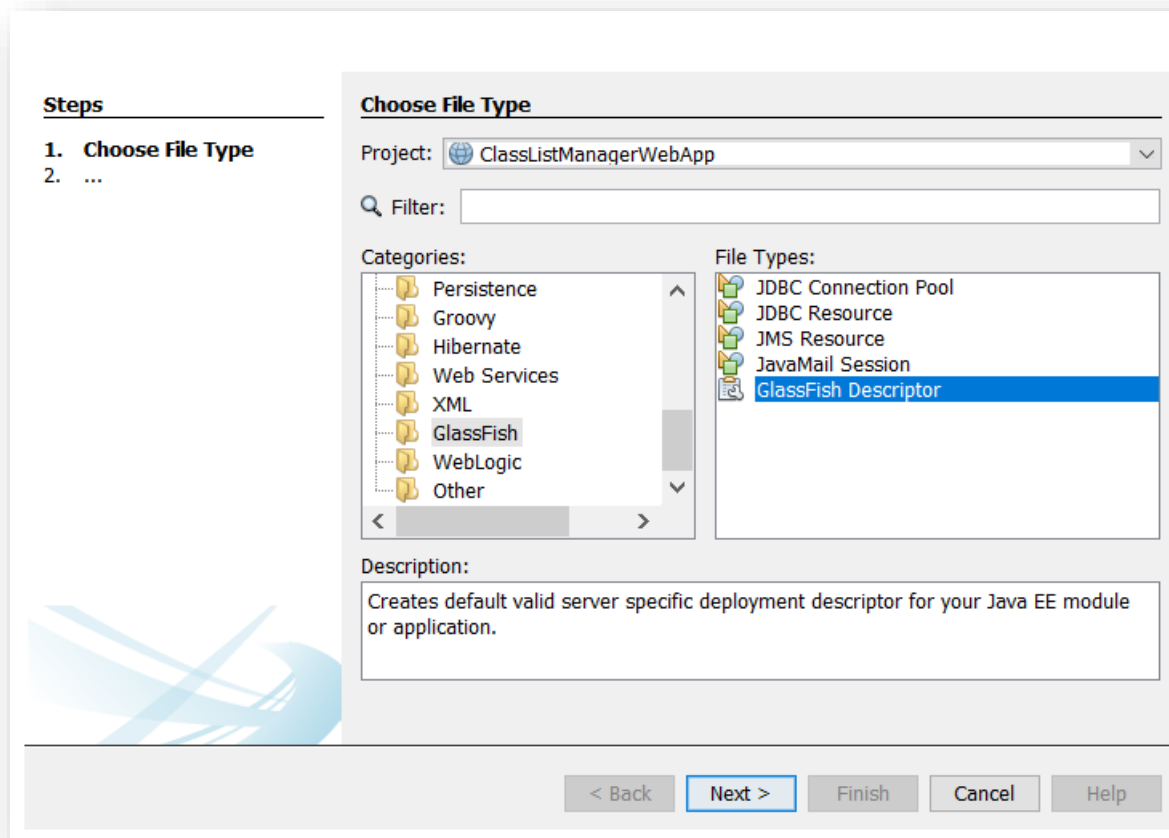
## 2.17.1 Map NetBeans roles and GlassFish users

### 2.17.1.1 Right-click on the project and select **New | Other**



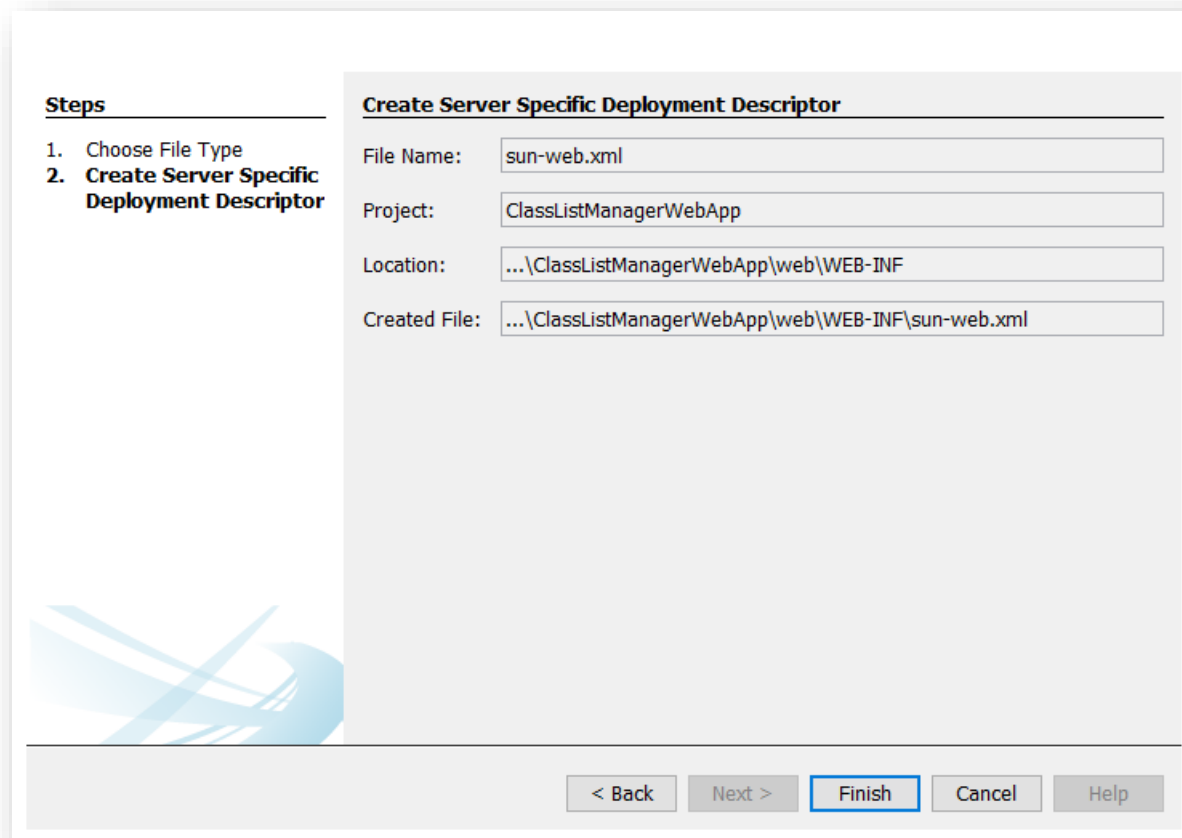
### 2.17.1.2

Select **GlassFish** under **Categories** and **GlassFish Descriptor** under **File Types**.



2.17.1.3

Click Next



**Steps**

1. Choose File Type
- 2. Create Server Specific Deployment Descriptor**

**Create Server Specific Deployment Descriptor**

File Name:

Project:

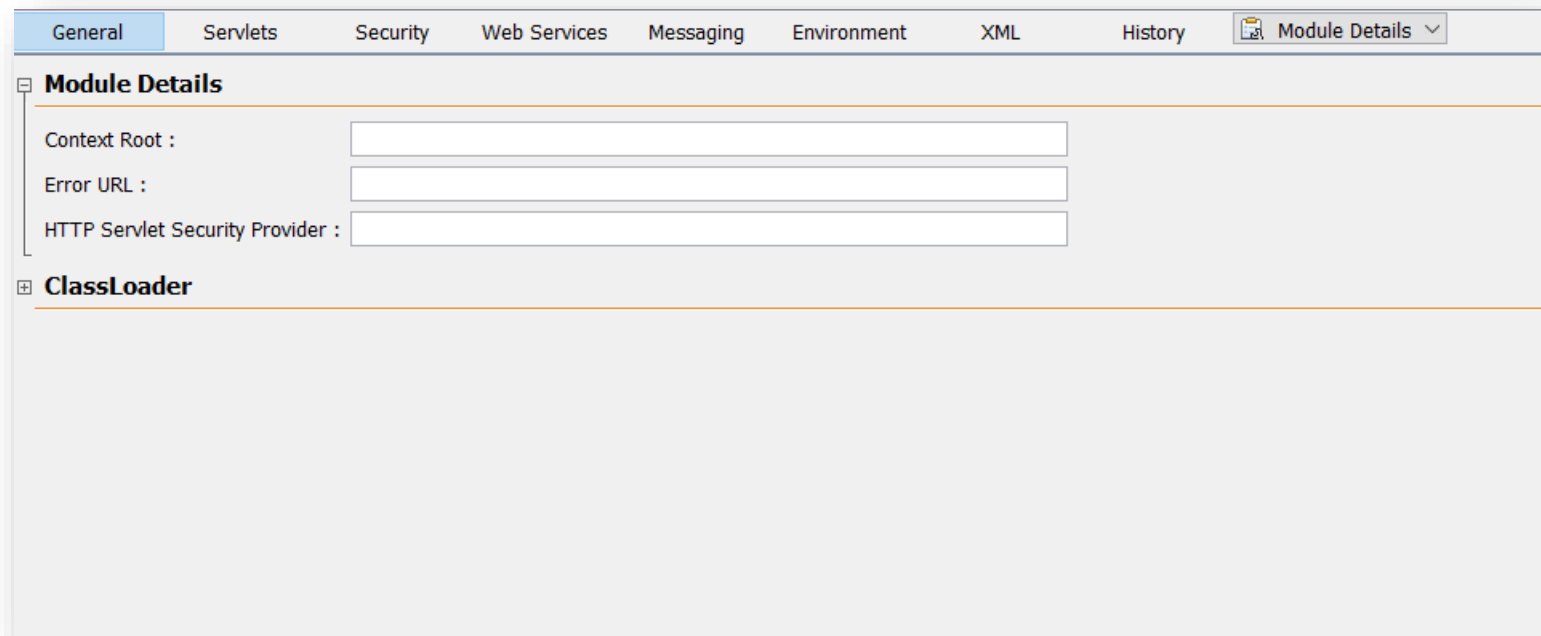
Location:

Created File:

< Back   Next >   **Finish**   Cancel   Help

2.17.1.4

Click **Finish**



The screenshot shows a configuration window with a tabbed interface. The 'Module Details' tab is selected, displaying three input fields for configuration. Below these fields is a section titled 'ClassLoader' which is currently collapsed.

Tab
General
Servlets
Security
Web Services
Messaging
Environment
XML
History
Module Details

**Module Details**

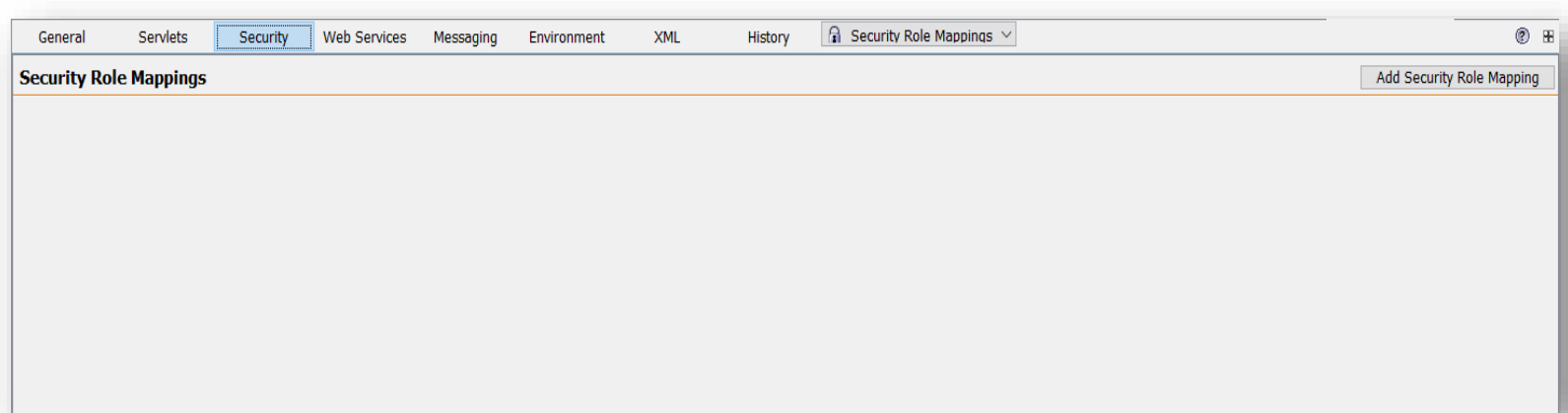
Context Root :

Error URL :

HTTP Servlet Security Provider :

**ClassLoader**

#### 2.17.1.5 Click on the Security tab



### 2.17.1.6 Click on Add Security Role Mapping

The screenshot shows a web application configuration interface with tabs for General, Servlets, Security, Web Services, Messaging, Environment, XML, and History. The Security tab is active, and a dropdown menu shows 'role1'. The main area is titled 'Security Role Mappings' and contains a section for 'role1'. At the top right of this section is a button labeled 'Add Security Role Mapping'. Below the title, there is a 'Security Role Name' field containing 'role1' and a 'Remove' button. The 'Principals Assigned to this Role' section includes a table with columns 'Principal Name' and 'Class Name', and buttons for 'Add Principal...', 'Edit Principal...', and 'Remove Principal(s)'. The 'Groups Assigned to this Role' section includes a table with a 'Group Name' column and buttons for 'Add Group...', 'Edit Group...', and 'Remove Group(s)'.

General Servlets **Security** Web Services Messaging Environment XML History role1

**Security Role Mappings** Add Security Role Mapping

role1 Remove

Security Role Name : role1

Principals Assigned to this Role

Principal Name	Class Name
----------------	------------

Add Principal...  
Edit Principal...  
Remove Principal(s)

Groups Assigned to this Role

Group Name
------------

Add Group...  
Edit Group...  
Remove Group(s)



2.17.1.7 Name the security role as teacher

General Servlets **Security** Web Services Messaging Environment XML

### Security Role Mappings

**teacher**

Security Role Name :

Principals Assigned to this Role

Principal Name	Class Name
----------------	------------

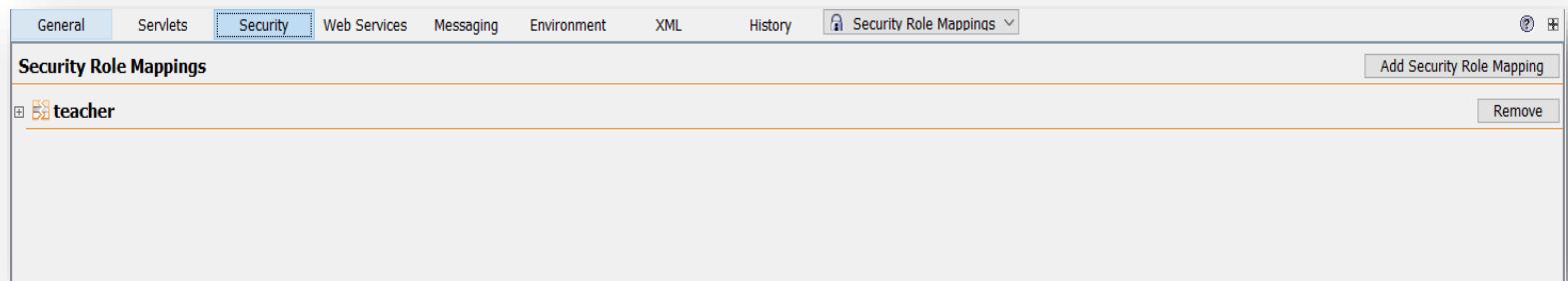
Groups Assigned to this Role

2.17.1.8 Click on Add group and name the group **teacher**.

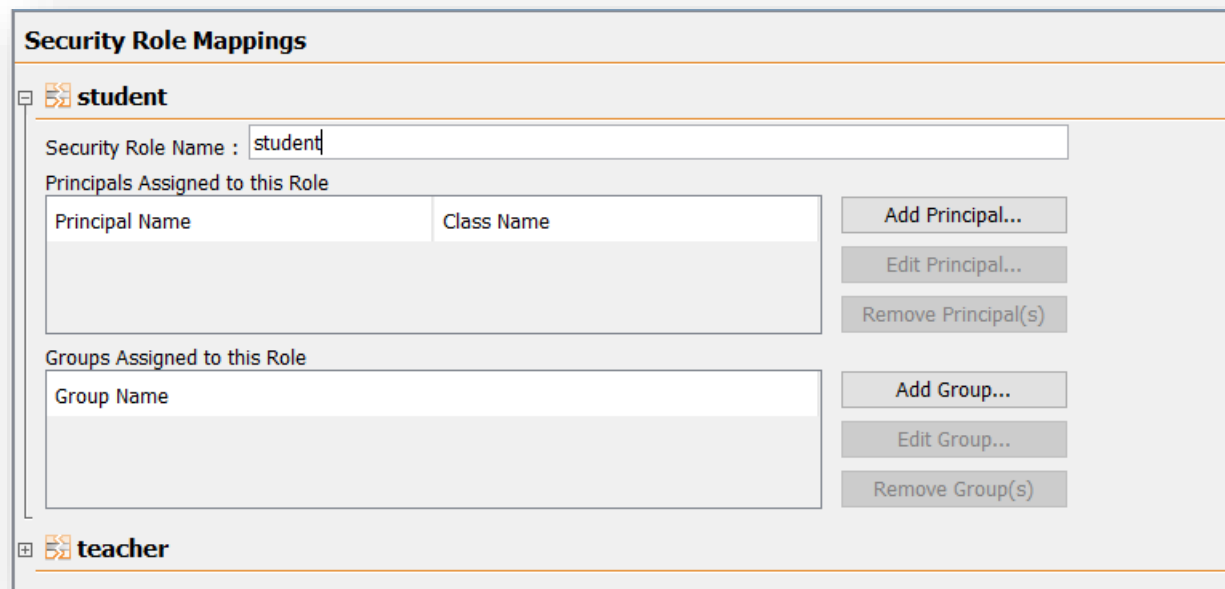
The screenshot shows the 'Security Role Mappings' configuration page for a role named 'teacher'. The page has a tabbed interface with 'Security' selected. Below the tabs, the role name 'teacher' is displayed. A text field for 'Security Role Name' contains 'teacher'. Under 'Principals Assigned to this Role', there is an empty table with columns 'Principal Name' and 'Class Name', and buttons for 'Add Principal...', 'Edit Principal...', and 'Remove Principal(s)'. Under 'Groups Assigned to this Role', there is a table with one row containing 'teacher' in the 'Group Name' column, which is highlighted in blue. To the right of this table are buttons for 'Add Group...', 'Edit Group...', and 'Remove Group(s)'.

Security Role Mappings	
teacher	
Security Role Name : teacher	
Principals Assigned to this Role	
Principal Name	Class Name
Add Principal...	
Edit Principal...	
Remove Principal(s)	
Groups Assigned to this Role	
Group Name	
teacher	
Add Group...	
Edit Group...	
Remove Group(s)	

2.17.1.9 Add another role, the student. Click on the Security tab of sun-web.xml





2.17.1.10 Click on **Add Security Role Mapping** button and name the role **student**.



2.17.1.11 Click on the Add Group button and name the group student.

**Security Role Mappings**

  **student**

Security Role Name :

Principals Assigned to this Role

Principal Name	Class Name

Add Principal...

Edit Principal...

Remove Principal(s)



Groups Assigned to this Role

Group Name
student

Add Group...

Edit Group...

Remove Group(s)

  **teacher**

2.18 Modify StudentFacade to state the roles allowed to perform certain functionalities.

```
@Stateless
public class StudentFacade extends AbstractFacade<Student> implements StudentFacadeLocal {

    @PersistenceContext(unitName = "ClassListManagerWebAppPU")
    private EntityManager em;

    @Override
    protected EntityManager getEntityManager() {
        return em;
    }

    public StudentFacade() {
        super(Student.class);
    }

    @RolesAllowed("teacher")
    @Override
    public void createStudent(Student student) {
        create(student);
    }
}
```

```

@RolesAllowed("teacher")
@Override
public void editStudent(Student student) {
    edit(student);
}

@RolesAllowed("teacher")
@Override
public void removeStudent(Student student) {
    remove(student);
}

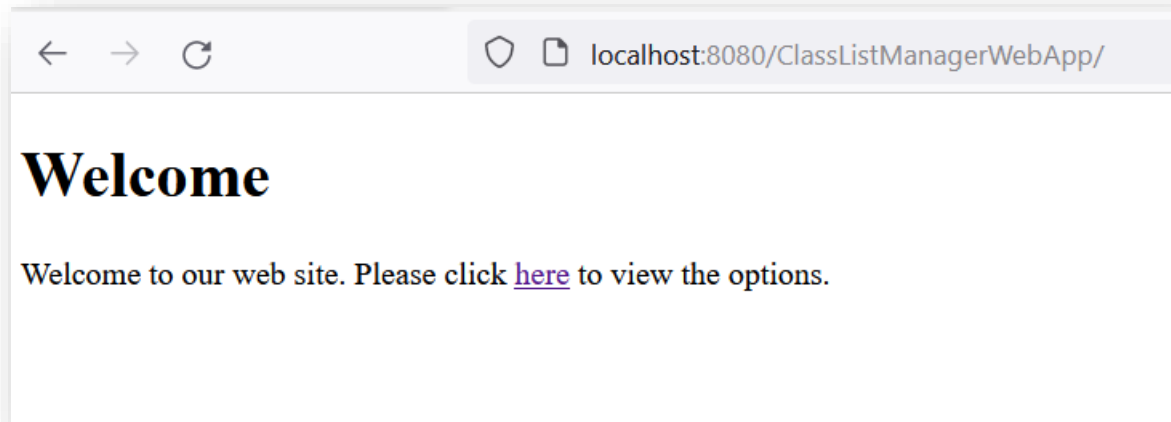
@RolesAllowed("teacher")
@Override
public Student findStudent(Object id) {
    Student targetStudent = find(id);
    return targetStudent;
}

@RolesAllowed({"teacher", "student"})
@Override
public List<Student> findAllStudent() {
    List<Student> list = findAll();
    return list;
}

```

#### D. Test secured web application

1. Run the application



2. Click on the link

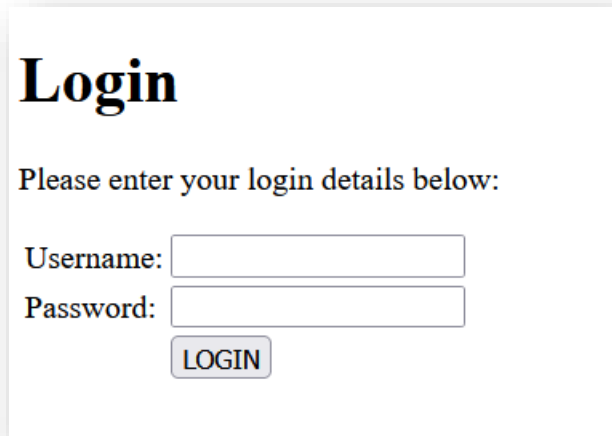
## Menu options

Please select one of the following options or click [here](#) to go back to the welcome page:

1. [Add a student](#)
2. [Find a student](#)
3. [Remove a student](#)
4. [Edit a student](#)
5. [View all students](#)



3. Add student.
  - 3.1 Click on the **Add a student** link. The container sees that the user is attempting to access a protected resource and is not authenticated yet, hence the login page.

A login form with a white background and a subtle drop shadow. It features a large, bold, black serif title 'Login' at the top left. Below the title is the instruction 'Please enter your login details below:'. This is followed by two input fields: 'Username:' and 'Password:', each with a corresponding text box. At the bottom right of the form is a button labeled 'LOGIN' in all caps, with a light blue gradient and a thin border.

# Login

Please enter your login details below:

Username:

Password:

### 3.2 Fill-in incorrect login details.

## Login

Please enter your login details below:

Username:

Password:

A login error page is returned.

## Login error

Your login details are invalid. Please click [here](#) to go back to the menu page.

3.3 Enter correct login details.



**Login**

Please enter your login details below:

Username:

Password:

The expected form is returned. Why? The user is authenticated and authorised to access the form.

## Add student

Please enter the details of the student below or [here](#) to go back to the menu page:

Name:	<input type="text"/>
Age	<input type="text"/>
Student number:	<input type="text"/>
<input type="button" value="ADD"/>	

### 3.4 Enter student details.

## Add student

Please enter the details of the student below or [here](#) to go back to the menu page:

Name:	<input type="text" value="Mulumba"/>
Age	<input type="text" value="23"/>
Student number:	<input type="text" value="666"/>
<input type="button" value="ADD"/>	

## Add student outcome

The student has been successfully added. Please click [here](#) to go back to the menu page.

3.5 Add another student.

## Add student

Please enter the details of the student below or [here](#) to go back to the menu page:

Name:

Age

Student number:

## Add student

Please enter the details of the student below or [here](#) to go back to the menu page:

Name:	<input type="text" value="Tumelo"/>
Age	<input type="text" value="23"/>
Student number:	<input type="text" value="555"/>
	<input type="button" value="ADD"/>



localhost:8080/ClassListManagerWebApp/AddStudentServlet.do

## Add student outcome

The student has been successfully added. Please click [here](#) to go back to the menu page.

4. View all the students

## Class list

**Name:** Vuyisile Memani

**Age:** 100

**Student number:** 111

**Name:** Thandeka

**Age:** 20

**Student number:** 222

**Name:** Thato

**Age:** 15

**Student number:** 333

**Name:** Khomo

**Age:** 12

**Student number:** 444

**Name:** Tumelo

**Age:** 23

**Student number:** 555

**Name:** Mulumba

**Age:** 23

**Student number:** 666

Click [here](#) to go back to the menu page.



5. Find a student

## Find student

Please enter the student number below:

Student number:

## Find student outcome

The student has been found. Below are the details of the student.

**Name:** Mulumba

**Age:** 23

**Student number:** 666

Click [here](#) to go back to the menu page.

6. Edit student

## Edit student

Please enter the student details below:

New student age

Student number:

## Edit student outcome

The student has been successfully edited. Click [here](#) to go back to the main page.

## Find student outcome

The student has been found. Below are the details of the student.

**Name:** Mulumba

**Age:** 100

**Student number:** 666

Click [here](#) to go back to the menu page.

7. Remove student

## Remove student

Please enter the student number below:

Student number:

## Remove student outcome

The student has been successfully removed. Please click [here](#) to go back to the menu page.

## Class list

**Name:** Vuyisile Memani

**Age:** 100

**Student number:** 111

**Name:** Thandeka

**Age:** 20

**Student number:** 222

**Name:** Thato

**Age:** 15

**Student number:** 333

**Name:** Khomo

**Age:** 12

**Student number:** 444

**Name:** Tumelo

**Age:** 23

**Student number:** 555

Click [here](#) to go back to the menu page.

8. Test the student. Get to the menu page.

## Menu options

Please select one of the following options or click [here](#) to go back to the welcome page:

1. [Add a student](#)
2. [Find a student](#)
3. [Remove a student](#)
4. [Edit a student](#)
5. [View all students](#)

9. Click on the Add a student link.

## Login

Please enter your login details below:

Username:

Password:

LOGIN

10. Enter correct details of the student.

# Login

Please enter your login details below:

Username:

Password:

## HTTP Status 403 - Forbidden

---

type	Status report
message	Forbidden
description	Access to the specified resource has been forbidden.

---

GlassFish Server Open Source Edition 4.1



11. Write a JSP file to handle the 403 exception.

```
<%--
    Document    : userforbidden
    Created on  : 28 May 2022, 10:22:52 PM
    Author      : MemaniV
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>User Forbidden Page</title>
    </head>
    <body>
        <h1>Access denied</h1>
        <p>You are not allowed to access the resource. Please click <a href="menu.html">here</a>
    </body>
</html>
```

```
<security-role>
  <description>A student can only view all the
  <role-name>student</role-name>
</security-role>
<error-page>
  <error-code>403</error-code>
  <location>/userforbidden.jsp</location>
</error-page>
</web-app>
```

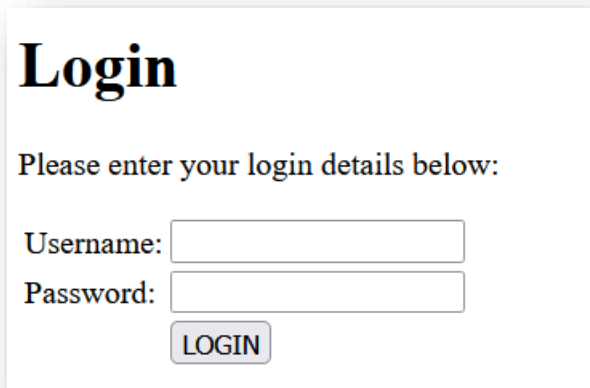
12. Get to the menu again

## Menu options

Please select one of the following options or click [here](#) to go back to the welcome page:

1. [Add a student](#)
2. [Find a student](#)
3. [Remove a student](#)
4. [Edit a student](#)
5. [View all students](#)

13. Click on the Add a student link.



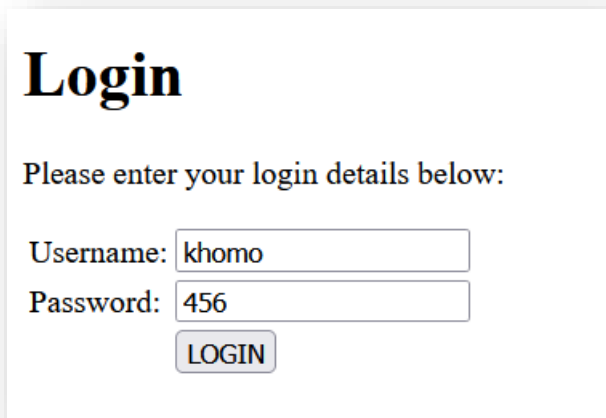
**Login**

Please enter your login details below:

Username:

Password:

14. Enter correct login details



**Login**

Please enter your login details below:

Username:

Password:

## Access denied

You are not allowed to access the resource. Please click [here](#) to go back to the menu page.

**A student is not allowed to add into the database.**

15. Click on the link.

## Menu options

Please select one of the following options or click [here](#) to go back to the welcome page:

1. [Add a student](#)
2. [Find a student](#)
3. [Remove a student](#)
4. [Edit a student](#)
5. [View all students](#)

16. Click on the Find a student link.

## Access denied

You are not allowed to access the resource. Please click [here](#) to go back to the menu page.

**A student is not allowed to search the database.**

17. Click on the link.

## Menu options

Please select one of the following options or click [here](#) to go back to the welcome page:

1. [Add a student](#)
2. [Find a student](#)
3. [Remove a student](#)
4. [Edit a student](#)
5. [View all students](#)

18. Click on the Remove a student link.

## Access denied

You are not allowed to access the resource. Please click [here](#) to go back to the menu page.

**A student is not allowed to remove from the database.**

19. Click on the link.

## Menu options

Please select one of the following options or click [here](#) to go back to the welcome page:

1. [Add a student](#)
2. [Find a student](#)
3. [Remove a student](#)
4. [Edit a student](#)
5. [View all students](#)

20. Click on the Edit a student link.

## Access denied

You are not allowed to access the resource. Please click [here](#) to go back to the menu page.

**A student is not allowed to edit the database.**

21. Click on the link.

## Menu options

Please select one of the following options or click [here](#) to go back to the welcome page:

1. [Add a student](#)
2. [Find a student](#)
3. [Remove a student](#)
4. [Edit a student](#)
5. [View all students](#)

22. Click on View all students link.

## Get all students

Click on the button below to get all the students.

GET ALL STUDENTS



23. Click on the button

## Class list

**Name:** Vuyisile Memani

**Age:** 100

**Student number:** 111

**Name:** Thandeka

**Age:** 20

**Student number:** 222

**Name:** Thato

**Age:** 15

**Student number:** 333

**Name:** Khomo

**Age:** 12

**Student number:** 444

**Name:** Tumelo

**Age:** 23

**Student number:** 555

Click [here](#) to go back to the menu page.