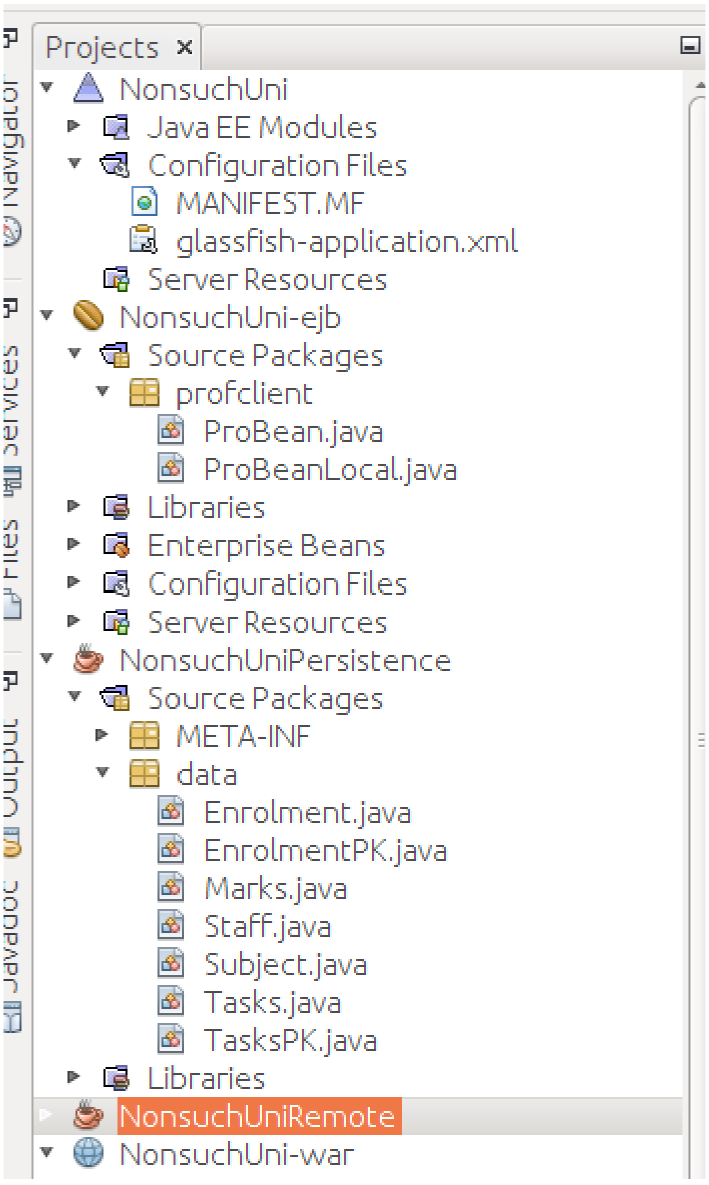
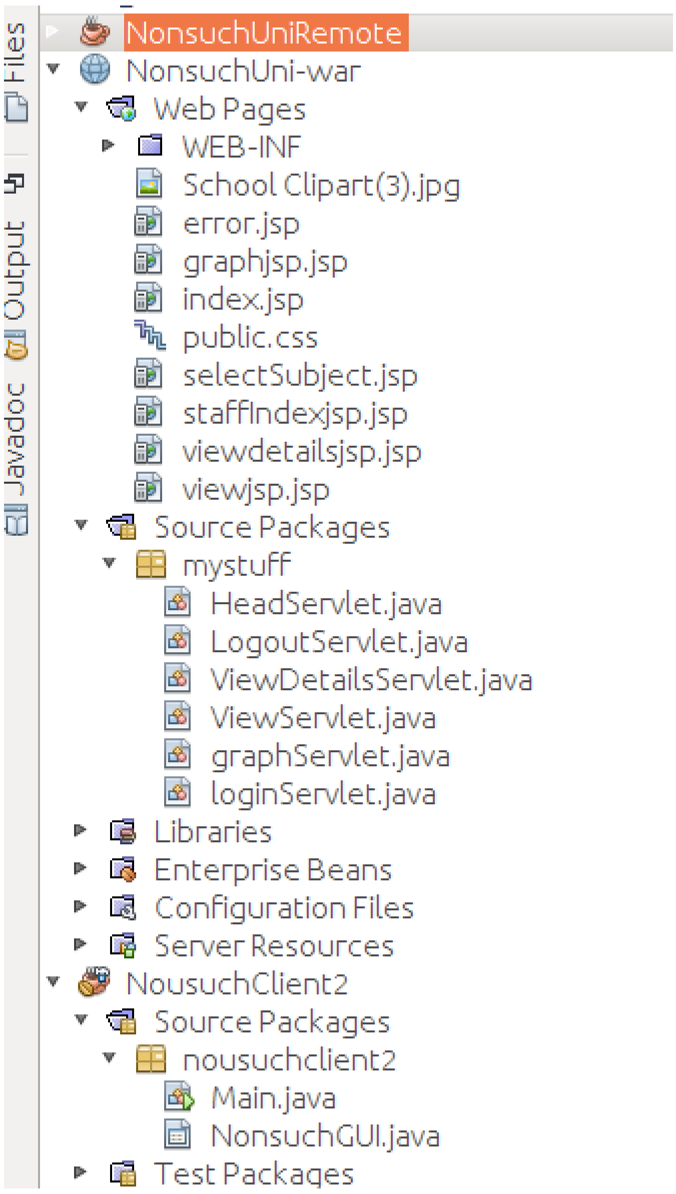
CSCI398 A4

Overview

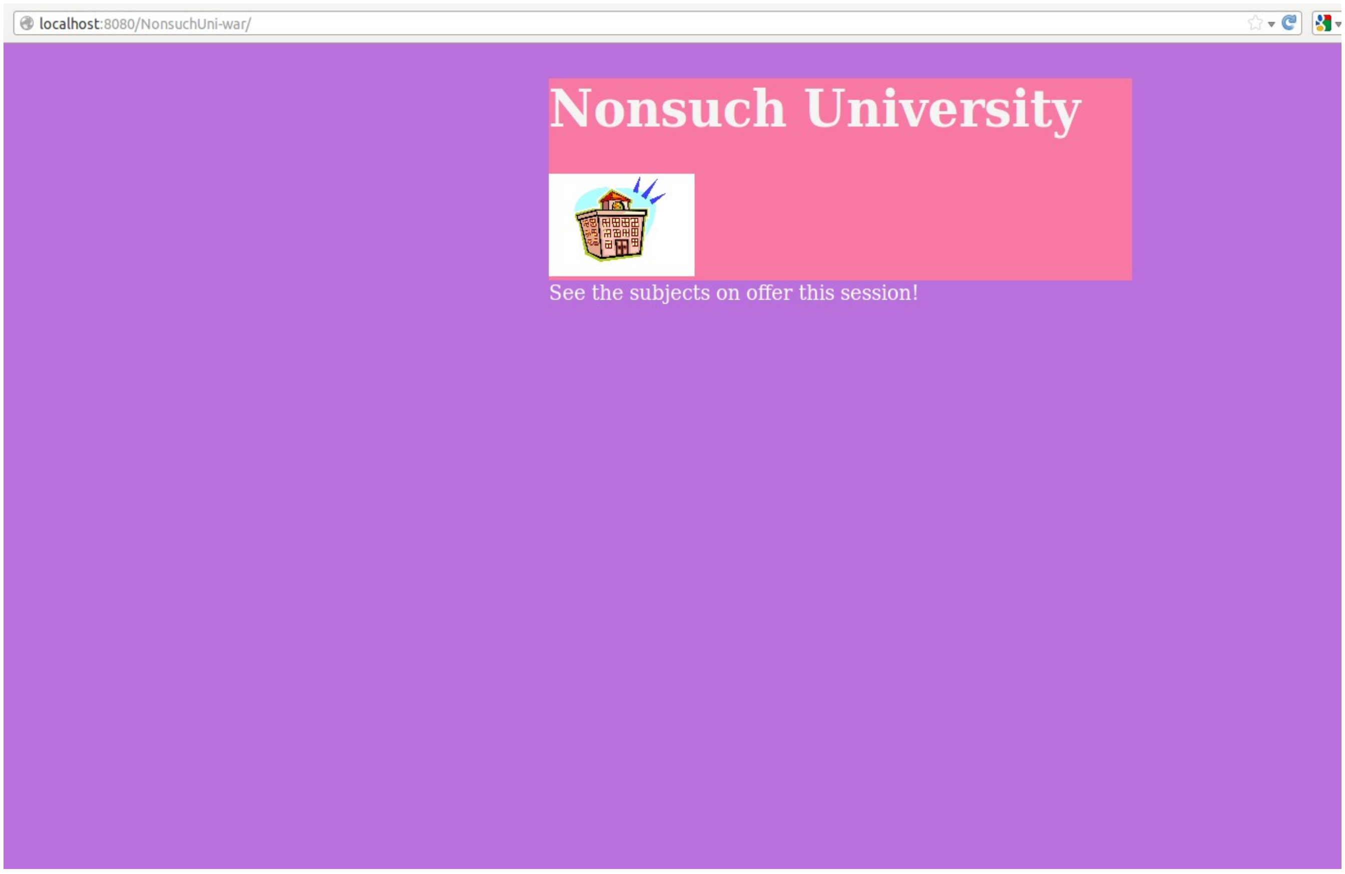


```
▼ NonsuchUni
    ▶ Java EE Modules
    ▼ Configuration Files
        MANIFEST.MF
        glassfish-application.xml
      Server Resources
▼ NonsuchUni-ejb
    ▼ Source Packages
        ▼ profclient
            ProBean.java
            ProBeanLocal.java
    ▶ Libraries
    ▶ Enterprise Beans
    ▶ Configuration Files
    ▶ Server Resources
▼ NonsuchUniPersistence
    ▼ Source Packages
        ▶ META-INF
        ▼ data
            Enrolment.java
            EnrolmentPK.java
            Marks.java
            Staff.java
            Subject.java
            Tasks.java
            TasksPK.java
    ▶ Libraries
▶ NonsuchUniRemote
▼ NonsuchUni-war
```



```
▶ NonsuchUniRemote
▼ NonsuchUni-war
    ▼ Web Pages
        ▶ WEB-INF
          School Clipart(3).jpg
          error.jsp
          graphjsp.jsp
          index.jsp
          public.css
          selectSubject.jsp
          staffIndexjsp.jsp
          viewdetailsjsp.jsp
          viewjsp.jsp
    ▼ Source Packages
        ▼ mystuff
            HeadServlet.java
            LogoutServlet.java
            ViewDetailsServlet.java
            ViewServlet.java
            graphServlet.java
            loginServlet.java
    ▶ Libraries
    ▶ Enterprise Beans
    ▶ Configuration Files
    ▶ Server Resources
▼ NousuchClient2
    ▼ Source Packages
        ▼ nousuchclient2
            Main.java
            NonsuchGUI.java
    ▶ Test Packages
```

NonsuchUni-war
using local bean and JSP and serlvet to present subject information
<u>1.index page that do not need to login</u>
  just one line that link to subjects list.


localhost:8080/NonsuchUni-war/

Nonsuch University

See the subjects on offer this session!

Code for this page:

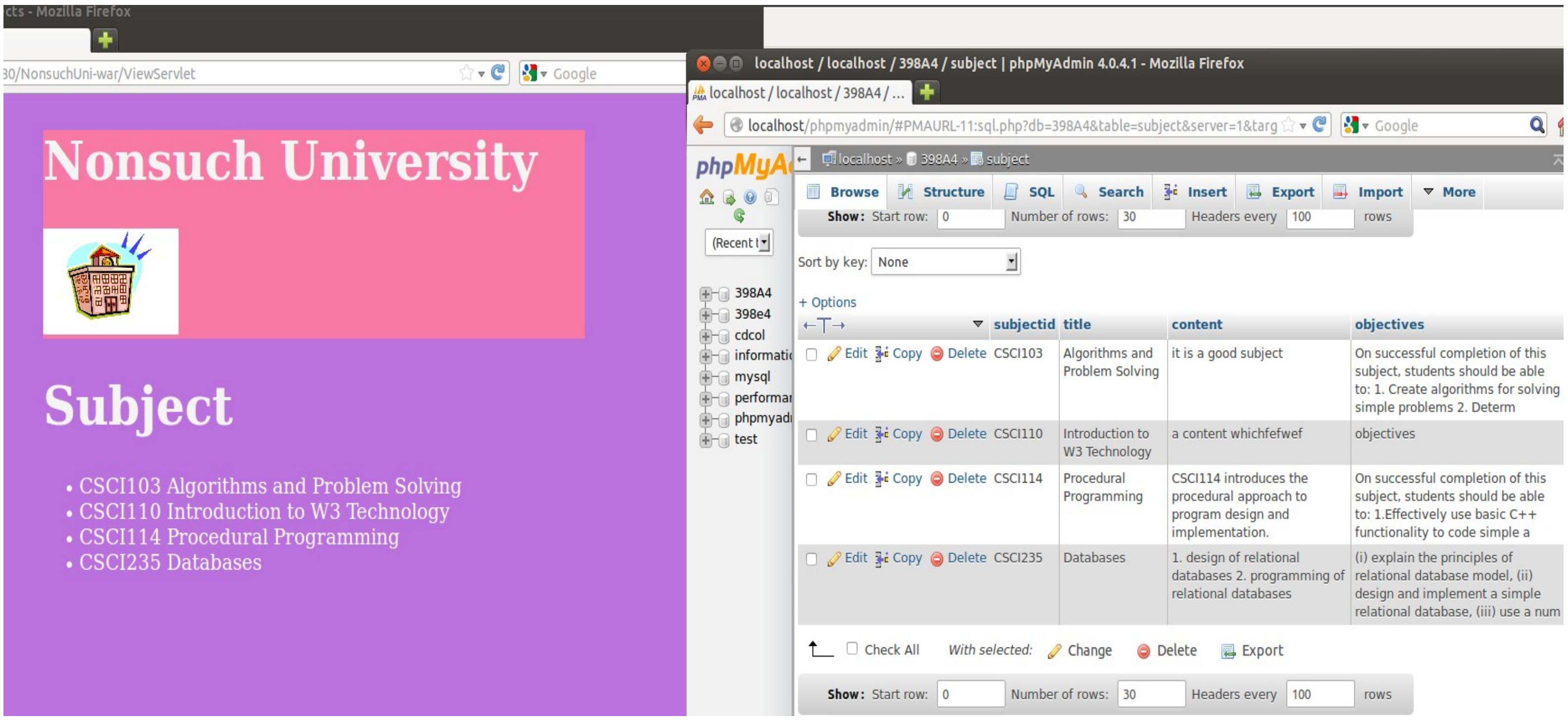
/home/user/NetBeansProjects/NonsuchUni/NonsuchUni-war/web/index.jsp

```
<%--
    Document   : index
    Created on : 18/10/2013, 11:11:09 AM
    Author     : user
--%>


<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <script type='text/javascript' src='public.js'></script>
        <link rel="stylesheet" type="text/css" href="public.css"/>
        <title>Welcome Page</title>
    </head>
    <div id='head'>
        <h1>Nonsuch University</h1>
        <img src="School Clipart(3).jpg" width="150" height="100" alt="School
Clipart(3)"/>
    </div>

    <body>

        <a href='ViewServlet'>See the subjects on offer this session!
</a><br/>

```

```
    </body>
</html>

```

2.subject list page

the image shows that all the information is matching the database named 298a4


cts - Mozilla Firefox
30/NonsuchUni-war/ViewServlet
Google
localhost / localhost / 398A4 / subject | phpMyAdmin 4.0.4.1 - Mozilla Firefox
localhost / localhost / 398A4 / ...
localhost/phpmyadmin/#PMAURL-11:sql.php?db=398A4&table=subject&server=1&targ
Google
phpMyA
(Recent t
398A4
398e4
cdcol
informati
mysql
performa
phpmyadi
test
localhost » 398A4 » subject
Browse
Structure
SQL
Search
Insert
Export
Import
More

Nonsuch University

Subject

- CSCI103 Algorithms and Problem Solving
- CSCI110 Introduction to W3 Technology
- CSCI114 Procedural Programming
- CSCI235 Databases

Show: Start row: 0 Number of rows: 30 Headers every 100 rows

Sort by key: None

+ Options

	subjectid	title	content	objectives
Edit Copy Delete	CSCI103	Algorithms and Problem Solving	it is a good subject	On successful completion of this subject, students should be able to: 1. Create algorithms for solving simple problems 2. Determ
Edit Copy Delete	CSCI110	Introduction to W3 Technology	a content whichfefwef	objectives
Edit Copy Delete	CSCI114	Procedural Programming	CSCI114 introduces the procedural approach to program design and implementation.	On successful completion of this subject, students should be able to: 1.Effectively use basic C++ functionality to code simple a
Edit Copy Delete	CSCI235	Databases	1. design of relational databases 2. programming of relational databases	(i) explain the principles of relational database model, (ii) design and implement a simple relational database, (iii) use a num

Check All *With selected:* Change Delete Export

Show: Start row: 0 Number of rows: 30 Headers every 100 rows

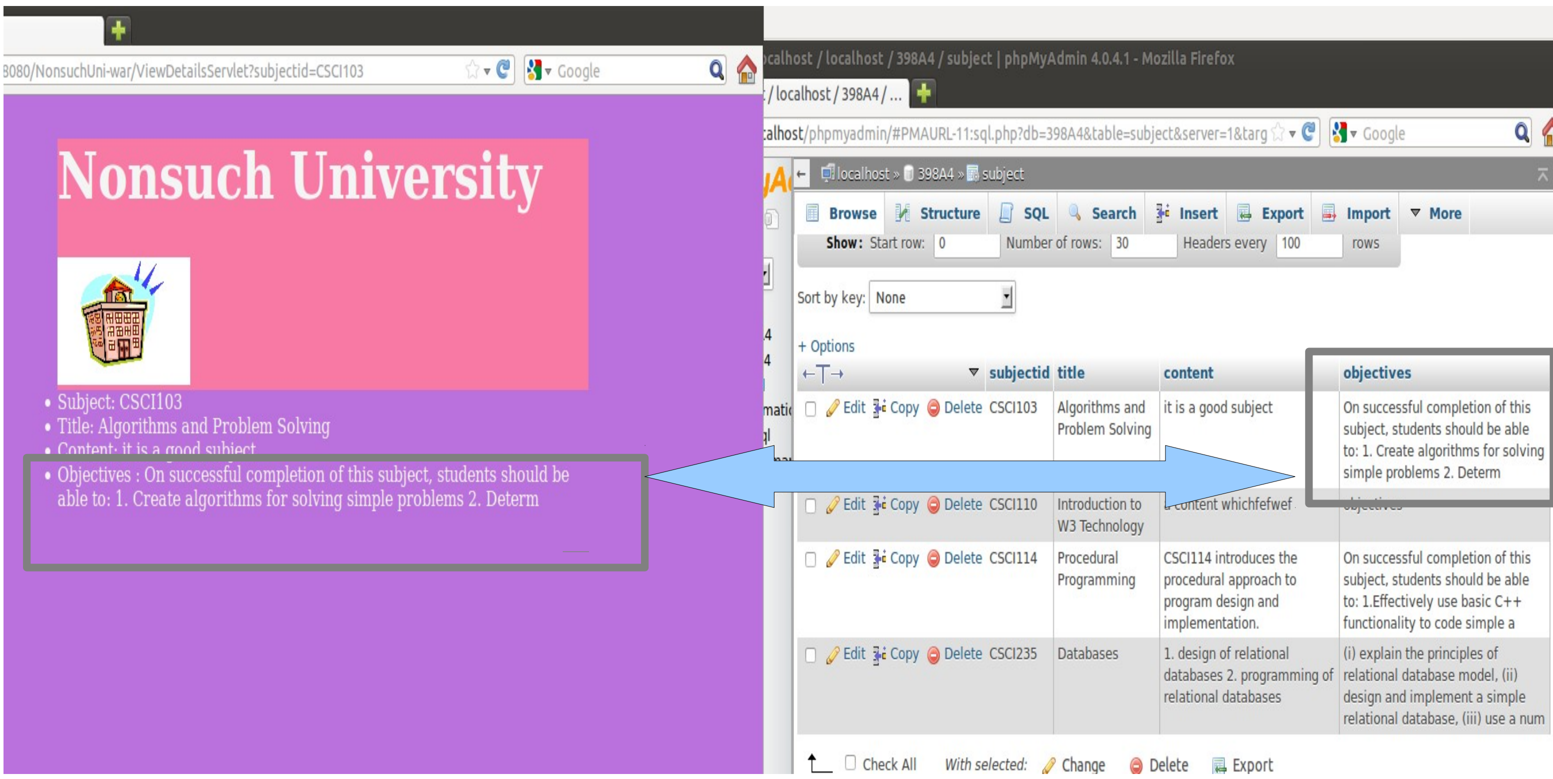
code

method listSubject() in local bean has been called.

/home/user/NetBeansProjects/NonsuchUni/NonsuchUni-war/src/java/mystuff/ViewServlet.java

```
public class ViewServlet extends HttpServlet {
    @EJB
    private ProBeanLocal proBean;

    /**
     * Processes requests for both HTTP
     * <code>GET</code> and
     * <code>POST</code> methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            List<Subject> results=proBean.listSubject();
            request.setAttribute("Subject",results);
            //forwrd to JSP
            RequestDispatcher dispatch=request.getRequestDispatcher("viewjsp.jsp");
            dispatch.forward(request,response);
        } catch(Exception e){

        }finally {
           // out.close();
        }
    }
```

3.Details page



8080/NonsuchUni-war/ViewDetailsServlet?subjectid=CSCI103

Nonsuch University

- Subject: CSCI103
- Title: Algorithms and Problem Solving
- Content: it is a good subject
- Objectives : On successful completion of this subject, students should be able to: 1. Create algorithms for solving simple problems 2. Determ

localhost / localhost / 398A4 / subject | phpMyAdmin 4.0.4.1 - Mozilla Firefox

localhost/phpmyadmin/#PMAURL-11:sql.php?db=398A4&table=subject&server=1&targ

localhost » 398A4 » subject

Show: Start row: 0 Number of rows: 30 Headers every 100 rows

Sort by key: None

	subjectid	title	content	objectives
Edit Copy Delete	CSCI103	Algorithms and Problem Solving	it is a good subject	On successful completion of this subject, students should be able to: 1. Create algorithms for solving simple problems 2. Determ
Edit Copy Delete	CSCI110	Introduction to W3 Technology	a content whichfefwef	objectives
Edit Copy Delete	CSCI114	Procedural Programming	CSCI114 introduces the procedural approach to program design and implementation.	On successful completion of this subject, students should be able to: 1.Effectively use basic C++ functionality to code simple a
Edit Copy Delete	CSCI235	Databases	1. design of relational databases 2. programming of relational databases	(i) explain the principles of relational database model, (ii) design and implement a simple relational database, (iii) use a num

code

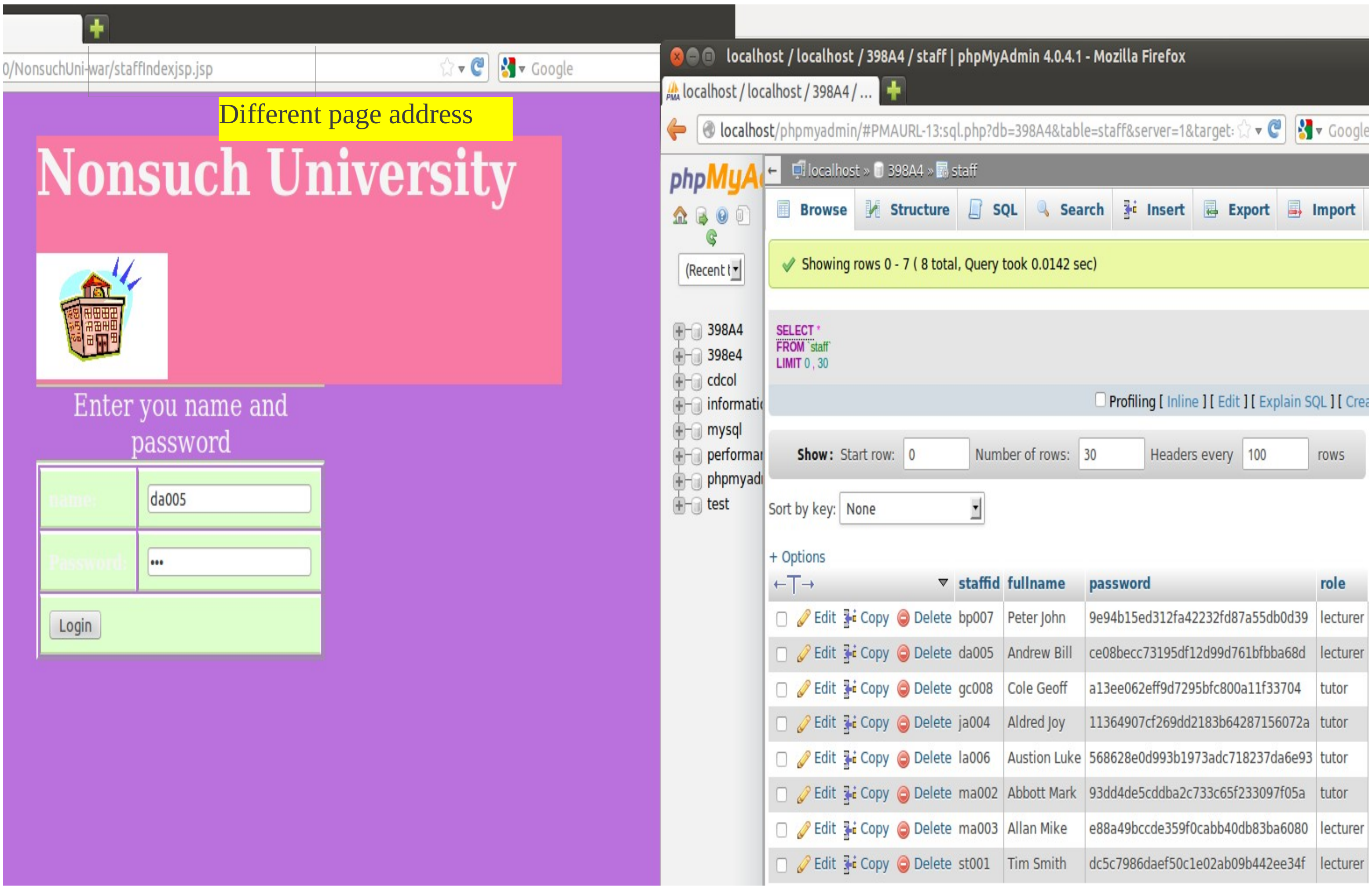
/home/user/NetBeansProjects/NonsuchUni/NonsuchUni-war/web/viewdetailsjsp.jsp

```
<%--
    Document   : viewdetailsjsp
    Created on : 19/10/2013, 3:48:17 PM
    Author     : user
--%>

<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <link rel="stylesheet" type="text/css" href="public.css"/>
        <title>Details</title>
    </head>

    <body>
        <div id='head'>
            <h1>Nonsuch University</h1>
            <img src="School Clipart(3).jpg" width="150" height="100"
alt="School Clipart(3)"/>
        </div>
        <c:forEach var="record" items="${requestScope.Subject}">
        <li>Subject: <c:out value="${record.subjectid}"/> </li>
        <li>Title: <c:out value="${record.title}"/></li>
        <li>Content: <c:out value="${record.content}"/></li>
        <li>Objectives : <c:out value="${record.objectives}"/></li>
        </c:forEach>
</body>
</html>
```

4.login page

pass word and name will be needed.

Pass word has been md5


0/NonsuchUni-war/staffIndexjsp.jsp
Google
Different page address
Nonsuch University
Enter you name and password
name:
da005
Password:
•••
Login
localhost / localhost / 398A4 / staff | phpMyAdmin 4.0.4.1 - Mozilla Firefox
localhost / localhost / 398A4 / ...
localhost/phpmyadmin/#PMAURL-13:sql.php?db=398A4&table=staff&server=1&target:
phpMyA
localhost » 398A4 » staff
Browse
Structure
SQL
Search
Insert
Export
Import
(Recent t
398A4
398e4
cdcol
informatic
mysql
performar
phpmyadi
test
Profiling [Inline] [Edit] [Explain SQL] [Crea
Show: Start row: 0
Number of rows: 30
Headers every 100 rows
Sort by key: None
+ Options

Showing rows 0 - 7 (8 total, Query took 0.0142 sec)

```
SELECT *
FROM `staff`
LIMIT 0 , 30
```

	staffid	fullname	password	role
Edit Copy Delete	bp007	Peter John	9e94b15ed312fa42232fd87a55db0d39	lecturer
Edit Copy Delete	da005	Andrew Bill	ce08becc73195df12d99d761bfbba68d	lecturer
Edit Copy Delete	gc008	Cole Geoff	a13ee062eff9d7295bfc800a11f33704	tutor
Edit Copy Delete	ja004	Aldred Joy	11364907cf269dd2183b64287156072a	tutor
Edit Copy Delete	la006	Austion Luke	568628e0d993b1973adc718237da6e93	tutor
Edit Copy Delete	ma002	Abbott Mark	93dd4de5cddba2c733c65f233097f05a	tutor
Edit Copy Delete	ma003	Allan Mike	e88a49bccde359f0cabb40db83ba6080	lecturer
Edit Copy Delete	st001	Tim Smith	dc5c7986daef50c1e02ab09b442ee34f	lecturer

code:

/home/user/NetBeansProjects/NonsuchUni/NonsuchUni-war/src/java/mystuff/ViewDetailsServlet.java

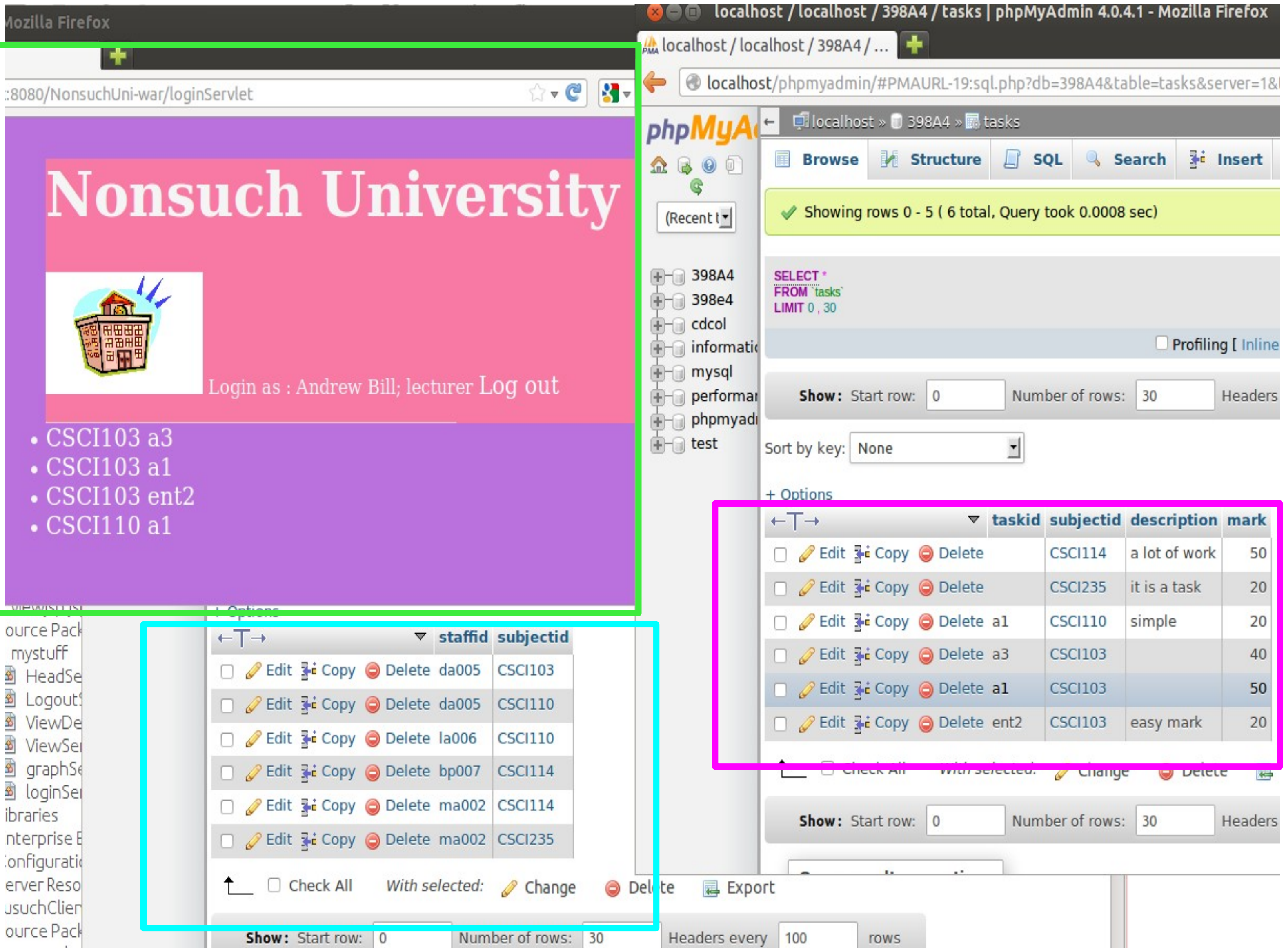
```java
public class ViewDetailsServlet extends
HttpServlet {

    @EJB
    private ProBeanLocal proBean;

    /**
     * Processes requests for both HTTP
     * <code>GET</code> and
     * <code>POST</code> methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {

            String subjectid = request.getParameter("subjectid");
            List<Subject> results = proBean.details(subjectid);
            request.setAttribute("Subject", results);
            //forwrd to JSP
            RequestDispatcher dispatch =
request.getRequestDispatcher("viewdetailsjsp.jsp");
            dispatch.forward(request, response);

        } catch{}finally {
            //out.close();
        }
    }
```

/home/user/NetBeansProjects/NonsuchUni/NonsuchUni-war/web/staffIndexjsp.jsp

```jsp
<%--
    Document   : staffIndexjsp
    Created on : 19/10/2013, 3:57:05 PM
    Author     : user
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <title>Log in</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <script type='text/javascript' src='public.js'></script>
        <link rel="stylesheet" type="text/css" href="public.css"/>
    </head>
    <body>
        <div id='head'>
```

```
            <h1>Nonsuch University</h1>
            <img src="School Clipart(3).jpg" width="150" height="100" alt="School Clipart(3)"/>
        </div>
        <form method=POST action="loginServlet">
            <table align="center" border='2'>
                <caption>Enter you name and password</caption>
                <tr>
                    <td align='right'><b>name:</b></td>
                    <td><input type='text' id="username"  size='15'>
                    </td>
                </tr>
                <tr>
                    <td align='right'><b>Password:</b></td>
                    <td ><input type='password' name='password'  size='15'>
                    </td>
                </tr>
                <tr>
                    <td colspan='2' Align='center'>
                        <input type='submit' value='Login'>

                </tr>
            </table>
        </form>
    </body>
</html>

```

2 task list page
only the subject that staff is teaching will be listed in this list
as you can see
CSCI235 CSCI 104 is also in the subject list but not shown here
CSCI103 CSCI110 are shown because da005 teach those two subjects.



Mozilla Firefox

:8080/NonsuchUni-war/loginServlet

Nonsuch University

Login as : Andrew Bill; lecturer Log out

- CSCI103 a3
- CSCI103 a1
- CSCI103 ent2
- CSCI110 a1

localhost / localhost / 398A4 / tasks | phpMyAdmin 4.0.4.1 - Mozilla Firefox

localhost/phpmyadmin/#PMAURL-19:sql.php?db=398A4&table=tasks&server=1&

localhost » 398A4 » tasks

Browse Structure SQL Search Insert

Showing rows 0 - 5 (6 total, Query took 0.0008 sec)

```
SELECT *
FROM `tasks`
LIMIT 0 , 30
```

Show: Start row: 0 Number of rows: 30 Headers

Sort by key: None

	taskid	subjectid	description	mark
Edit Copy Delete		CSCI114	a lot of work	50
Edit Copy Delete		CSCI235	it is a task	20
Edit Copy Delete	a1	CSCI110	simple	20
Edit Copy Delete	a3	CSCI103		40
Edit Copy Delete	a1	CSCI103		50
Edit Copy Delete	ent2	CSCI103	easy mark	20

	staffid	subjectid
Edit Copy Delete	da005	CSCI103
Edit Copy Delete	da005	CSCI110
Edit Copy Delete	la006	CSCI110
Edit Copy Delete	bp007	CSCI114
Edit Copy Delete	ma002	CSCI114
Edit Copy Delete	ma002	CSCI235

Check All With selected: Change Delete Export

Show: Start row: 0 Number of rows: 30 Headers every 100 rows

Code
data has been a hashmap that each subject and its task list pairs.
Serlvet

/home/user/NetBeansProjects/NonsuchUni/NonsuchUni-war/src/java/mystuff/loginServlet.java

```
public class loginServlet extends HttpServlet
{

    @EJB
    private ProBeanLocal proBean;

    /**
     * Processes requests for both HTTP
     * <code>GET</code> and
     * <code>POST</code> methods.
     *
```

```java
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {

            String pwd = request.getParameter("password");
            String name = request.getParameter("username");
            //Create MessageDigest object for MD5
            MessageDigest digest = MessageDigest.getInstance("MD5");
            //Update input string in message digest
            digest.update(pwd.getBytes(), 0, pwd.length());
            //Converts message digest value in base 16 (hex)
            String md5Pwd = new BigInteger(1, digest.digest()).toString(16);
            Staff staff = proBean.login(name, md5Pwd);
            if (staff == null) {
                RequestDispatcher dispatch =
request.getRequestDispatcher("error.jsp");
                dispatch.forward(request, response);
            } else {
                HashMap map = proBean.mapSubject();

                request.setAttribute("map", map);

                //forwrd to JSP
                RequestDispatcher dispatch =
request.getRequestDispatcher("selectSubject.jsp");
                dispatch.forward(request, response);
            }

        } catch (Exception ex) {
            ex.printStackTrace(out);
            //  Logger.getLogger(LogoutServlet.class
            //          .getName()).log(Level.SEVERE, null, ex);
        } finally {
            //  out.close();
        }
    }
```

/home/user/NetBeansProjects/NonsuchUni/NonsuchUni-war/web/selectSubject.jsp

```jsp
<%--
    Document   : sekectSubject
    Created on : 20/10/2013, 2:43:34 PM
    Author     : user
--%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>

    </head>
    <body>

        <jsp:include page="HeadServlet"/>
        <form method="POST" action="loginServlet">
            <c:forEach items="${map}" var="entry">
                <c:forEach items="${entry.value}" var="tasks">
                    <li><a href='graphServlet?subjectid=${entry.key}&task=${tasks.tasksPK.taskid}'>${entry.key}  ${tasks.tasksPK.taskid}</a></li>
                </c:forEach>
            </c:forEach>
        </form>
    </body>
</html>
```

JSP

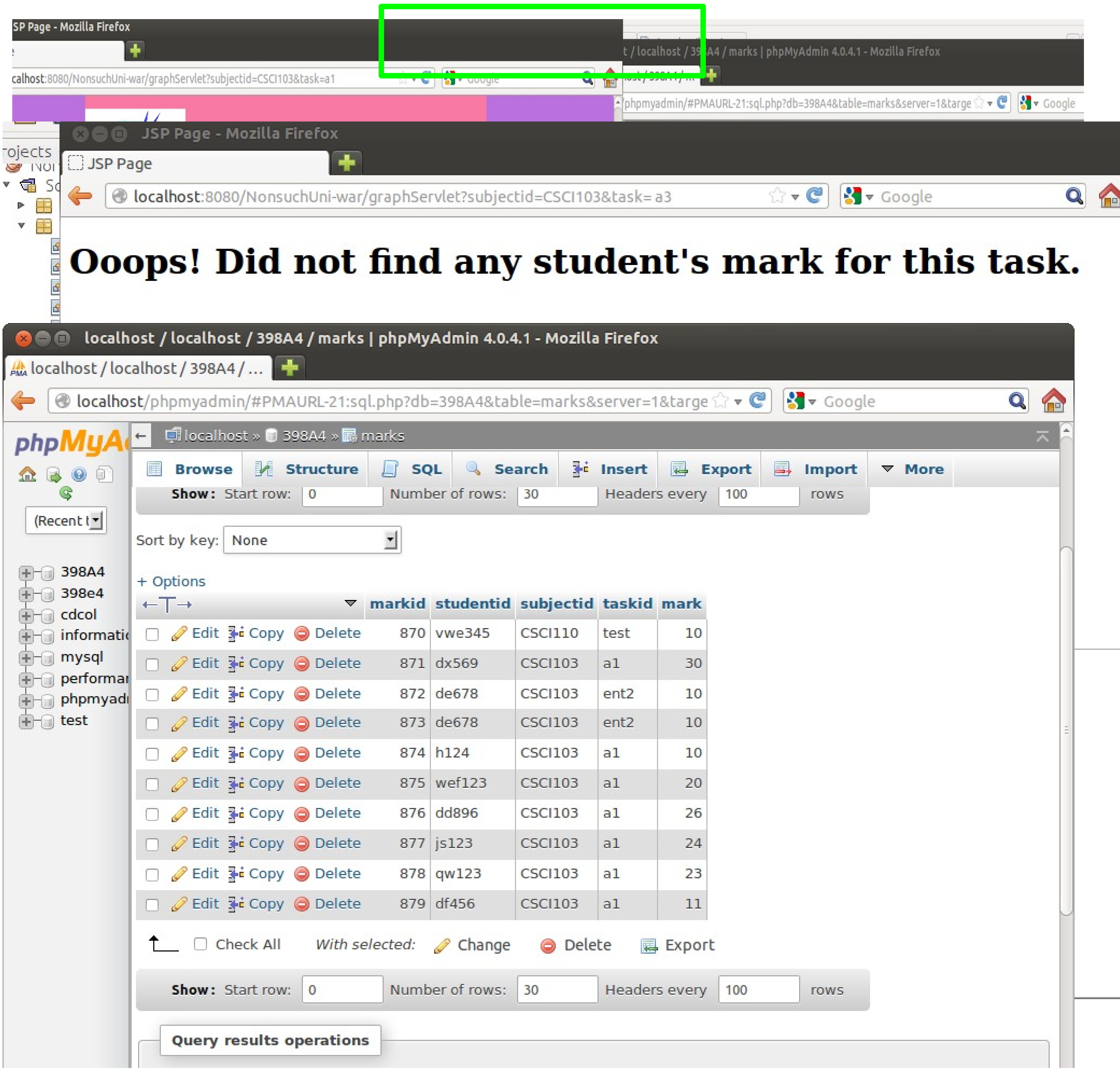
/home/user/NetBeansProjects/NonsuchUni/NonsuchUni-war/web/selectSubject.jsp

```
<%--
    Document   : sekectSubject
    Created on : 20/10/2013, 2:43:34 PM
    Author     : user
--%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>

    </head>
    <body>

        <jsp:include page="HeadServlet"/>
        <form method="POST" action="loginServlet">
            <c:forEach items="${map}" var="entry">
                <c:forEach items="${entry.value}" var="tasks">
                    <li><a href='graphServlet?subjectid=${entry.key}&task=${tasks.tasksPK.taskid}'>${entry.key}  ${tasks.tasksPK.taskid}</a></li>
                </c:forEach>
            </c:forEach>
        </form>
    </body>
</html>
```

5. Graph the data

if there are not marks in database a error page will shown
as shown below
1 subject id : csci103 and task: a3 do not has any mark inserted , so a erro page shown.
2 subject id csci103 and task a1 has data

the marks becomes to precentage



localhost:8080/NonsuchUni-war/graphServlet?subjectid=CSCI103&task=a3

Ooops! Did not find any student's mark for this task.

localhost / localhost / 398A4 / marks | phpMyAdmin 4.0.4.1 - Mozilla Firefox

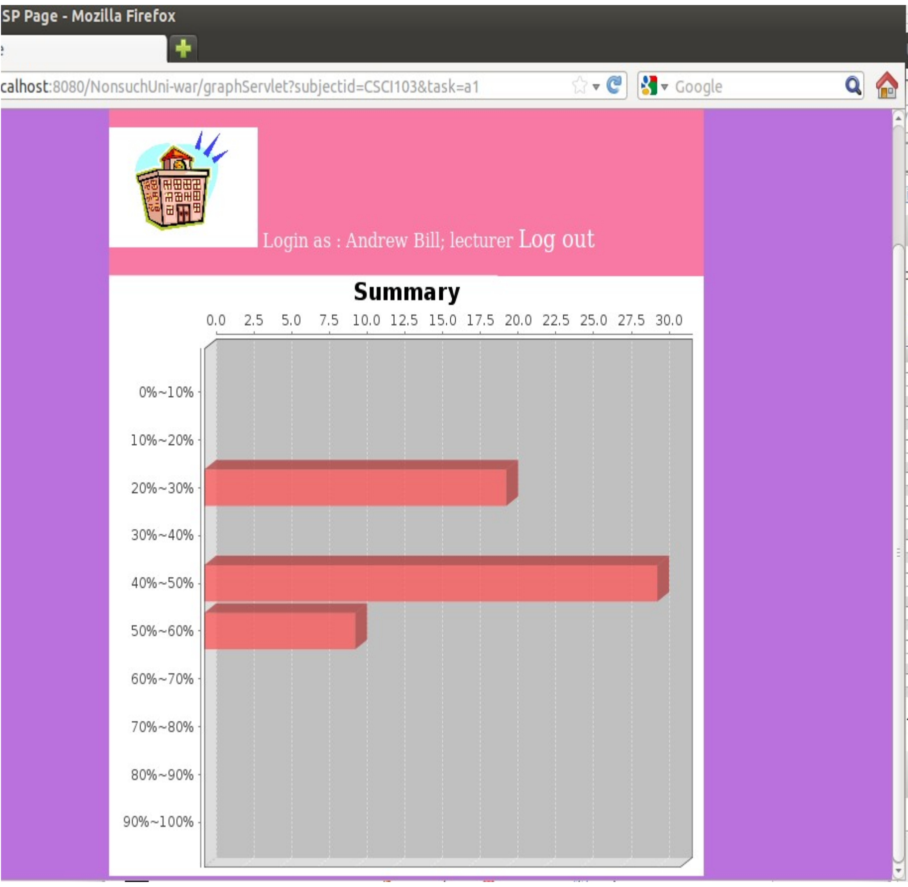
localhost/phpmyadmin/#PMAURL-21:sql.php?db=398A4&table=marks&server=1&targe

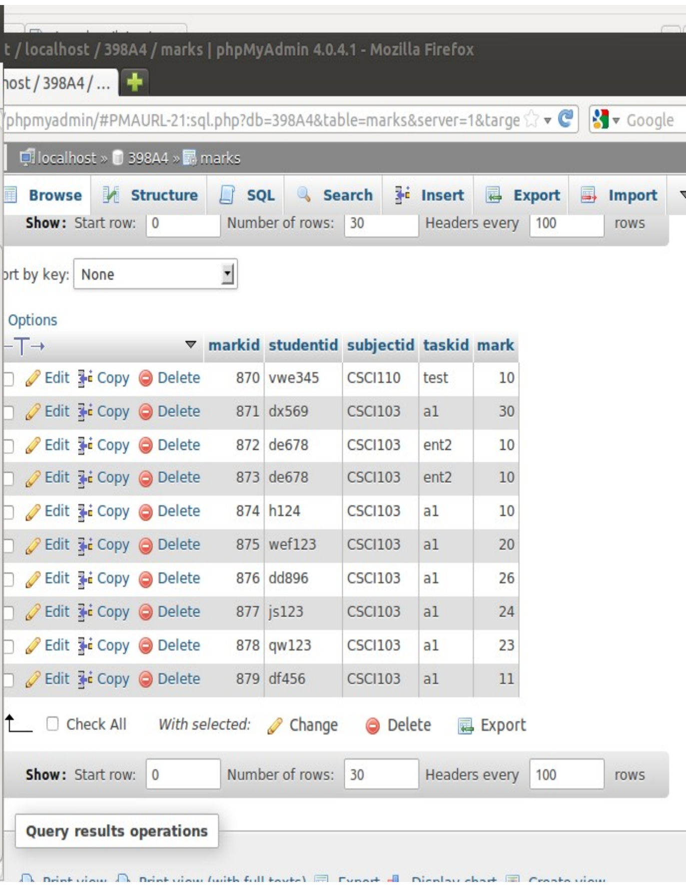
Show: Start row: 0 Number of rows: 30 Headers every 100 rows

Sort by key: None

markid	studentid	subjectid	taskid	mark
870	vwe345	CSCI110	test	10
871	dx569	CSCI103	a1	30
872	de678	CSCI103	ent2	10
873	de678	CSCI103	ent2	10
874	h124	CSCI103	a1	10
875	wef123	CSCI103	a1	20
876	dd896	CSCI103	a1	26
877	js123	CSCI103	a1	24
878	qw123	CSCI103	a1	23
879	df456	CSCI103	a1	11

Show: Start row: 0 Number of rows: 30 Headers every 100 rows




t / localhost / 398A4 / marks | phpMyAdmin 4.0.4.1 - Mozilla Firefox
host / 398A4 / ...
phpmyadmin/#PMAURL-21:sql.php?db=398A4&table=marks&server=1&targe
localhost » 398A4 » marks
Browse
Structure
SQL
Search
Insert
Export
Import

Show: Start row: 0 Number of rows: 30 Headers every 100 rows

rt by key: None

Options

	markid	studentid	subjectid	taskid	mark
Edit Copy Delete	870	vwe345	CSCI110	test	10
Edit Copy Delete	871	dx569	CSCI103	a1	30
Edit Copy Delete	872	de678	CSCI103	ent2	10
Edit Copy Delete	873	de678	CSCI103	ent2	10
Edit Copy Delete	874	h124	CSCI103	a1	10
Edit Copy Delete	875	wef123	CSCI103	a1	20
Edit Copy Delete	876	dd896	CSCI103	a1	26
Edit Copy Delete	877	js123	CSCI103	a1	24
Edit Copy Delete	878	qw123	CSCI103	a1	23
Edit Copy Delete	879	df456	CSCI103	a1	11

Check All *With selected:* Change Delete Export

Show: Start row: 0 Number of rows: 30 Headers every 100 rows

Query results operations

code

/home/user/NetBeansProjects/NonsuchUni/NonsuchUni-war/src/java/mystuff/graphServlet.java

```
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        int [] data;
        try {
            String subjectid=request.getParameter("subjectid");
            String tasks=request.getParameter("task");
            data=proBean.graph(subjectid, tasks);
            boolean flag=false;
            for(int i=0;i<data.length;i++)
            {
                if(data[i]!=0)
                {flag=true;}
            }
            if(flag==false)
            {
             RequestDispatcher dispatch =
```

```java
request.getRequestDispatcher("error.jsp");
            dispatch.forward(request, response);
            }
            else{
           HttpSession session = request.getSession(true);
            session.setAttribute("data",data);
          session.setAttribute("subjectid",subjectid);

            //forwrd to JSP
            RequestDispatcher dispatch = request.getRequestDispatcher("graphjsp.jsp");
            dispatch.forward(request, response);
            }

        }catch(Exception e){} finally {
           // out.close();
        }
    }
```

/home/user/NetBeansProjects/NonsuchUni/NonsuchUni-war/web/graphjsp.jsp

```jsp
<%--
    Document   : graphjsp
    Created on : 21/10/2013, 8:43:47 AM
    Author     : user
--%>

<%@page language="java" contentType="text/html" pageEncoding="UTF-8"%>
<%@page
import="javax.servlet.http.HttpServlet,javax.servlet.http.HttpSession,org.jfree.chart.ChartFactory, org.jfree.chart.JFreeChart, org.jfree.chart.plot.PlotOrientation, org.jfree.chart.servlet.ServletUtilities, org.jfree.data.category.DefaultCategoryDataset"
%>
<%
            HttpSession session2 = request.getSession(true);
            int []data=(int []) session2.getAttribute("data");
            DefaultCategoryDataset dataset=new DefaultCategoryDataset();
            for(int i=0;i<data.length;i++){
                dataset.addValue(data[i]*10, "yes",i*10+"%~"+(i+1)*10+"%");
            }
            JFreeChart chart=ChartFactory.createBarChart3D("Summary",
                    "", "", dataset, PlotOrientation.HORIZONTAL, false, false, false);
            String filename=ServletUtilities.saveChartAsPNG(chart, 600, 500, null, session2);
            String graphURL=request.getContextPath()+"/DisplayChart?filename="+filename;


%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
        <jsp:include page="HeadServlet"/>
        <img src="<%= graphURL%>">

    </body>
</html>

```

/home/user/NetBeansProjects/NonsuchUni/NonsuchUni-war/web/error.jsp

```
<%--
    Document   : error
    Created on : 22/10/2013, 7:41:50 PM
    Author     : user
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <h1>Ooops! Did not find any student's mark for this task.</h1>
    </body>
</html>

```

Generate code works for all pages

/home/user/NetBeansProjects/NonsuchUni/NonsuchUni-war/web/public.css

```
/*
    Document   : public
    Created on : May 23, 2013, 6:04:06 PM
    Author     : rebecca
    Description:
        Purpose of the stylesheet follows.
*/

root {
    display: block;
}
body{
    width:600px;
    margin:0px auto;
    color: whitesmoke;
    background-color:#BA71DD;
}
#head{
    background-color: #F879A4;
}
#foot{
    background-color: #B7AF02;
    }
a{
    font-size: 20px;
    color:#EFFFFF;
}
```

```
a:link, a:visited, a:hover, a:focus, a:active{
  color:white;
  text-decoration: none;
}
h1{
    font-size:50px;
}


table caption {
font-size: 24px;
border-bottom: 2px solid #B3DE94;
border-top: 2px solid #B3DE94;
}
table, td, th {
margin: 0;
padding: 0;
vertical-align: middle;
text-align:left;
}
tbody td, tbody th {
background-color: #DFC;
border-bottom: 2px solid #B3DE94;
border-top: 3px solid #FFFFFF;
padding: 9px;
}
tfoot td, tfoot th {
font-weight: bold;
padding: 4px 8px 6px 9px;
text-align:center;
}
thead th {
font-size: 14px;
font-weight: bold;
line-height: 19px;
padding: 0 8px 2px;
text-align:center;
}
```

/home/user/NetBeansProjects/NonsuchUni/NonsuchUni-war/src/java/mystuff/HeadServlet.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package mystuff;

import data.Staff;
import java.io.IOException;
import java.io.PrintWriter;
import javax.ejb.EJB;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import profclient.ProBeanLocal;

/**
 *
 * @author user
 */
```

```java
public class HeadServlet extends HttpServlet {
    @EJB
    private ProBeanLocal proBean;

    /**
     * Processes requests for both HTTP
     * <code>GET</code> and
     * <code>POST</code> methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<meta http-equiv=\"Content-Type\" content=\"text/html; charset=UTF-8\">");
            out.println("<script type='text/javascript' src='public.js'></script>");
            out.println("<link rel=\"stylesheet\" type=\"text/css\" href=\"public.css\"/>");
            out.println("<div id='head'>");
            out.println("<h1>Nonsuch University</h1>");
            out.println("<img src=\"School Clipart(3).jpg\" width=\"150\" height=\"100\" alt=\"School Clipart(3)\"/>");


                out.println("Login as : ");
                Staff rs=proBean.staffInfo();
                String role=rs.getRole();
                String fullname=rs.getFullname();
                out.println(fullname+"; "+role);
                out.println("<a href='LogoutServlet' >Log out </a><br/>");


out.println("_________________________________________________<br/>");
            out.println("</div>");
        } catch (Exception ex) {
            ex.printStackTrace(out);
          //  Logger.getLogger(LogoutServlet.class
          //          .getName()).log(Level.SEVERE, null, ex);
        }finally {

          //  out.close();
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP
     * <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP
     * <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>
}

```

/home/user/NetBeansProjects/NonsuchUni/NonsuchUni-war/src/java/mystuff/LogoutServlet.java

```
package mystuff;

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

import java.io.IOException;
import java.io.PrintWriter;
import javax.ejb.EJB;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import profclient.ProBeanLocal;

/**
 *
 * @author user
 */
```

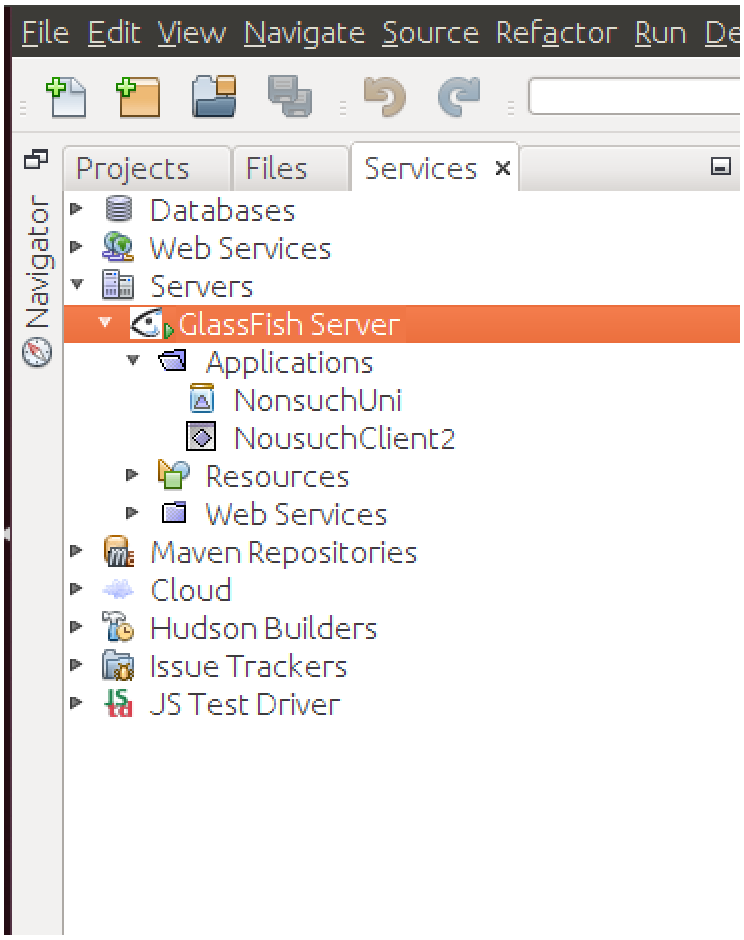
```java
public class LogoutServlet extends HttpServlet {
    @EJB
    private ProBeanLocal proBean;

    /**
     * Processes requests for both HTTP
     * <code>GET</code> and
     * <code>POST</code> methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {

            /* TODO output your page here. You may use following sample code. */
            proBean.Logout();
            out.println("Log out!");


        } finally {
            out.close();
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP
     * <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP
     * <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>
}

```

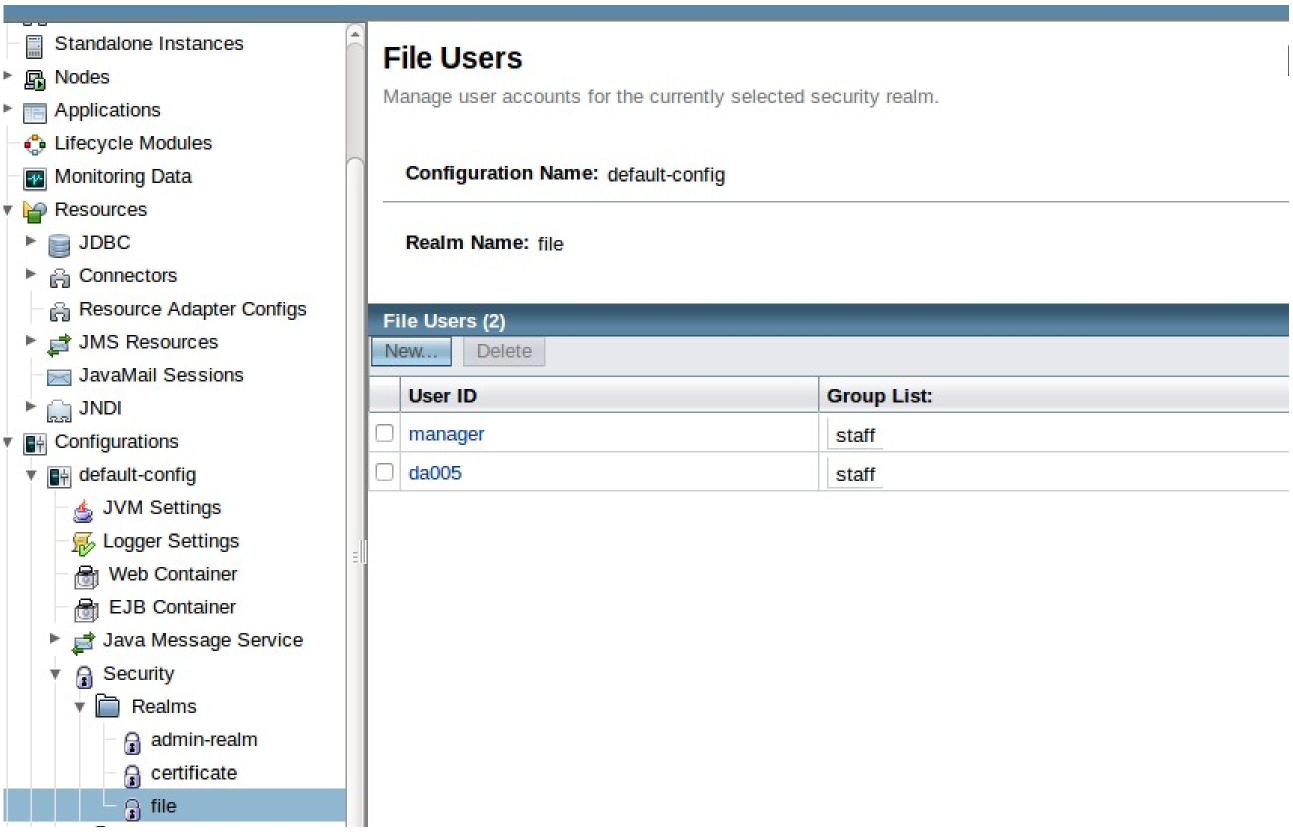
client project

log in part

first of all, NonsuchUni and NonsuchClient2 has been delopyed.



Glassfish Domain shows there are two users in group: staff



File Users

Manage user accounts for the currently selected security realm.

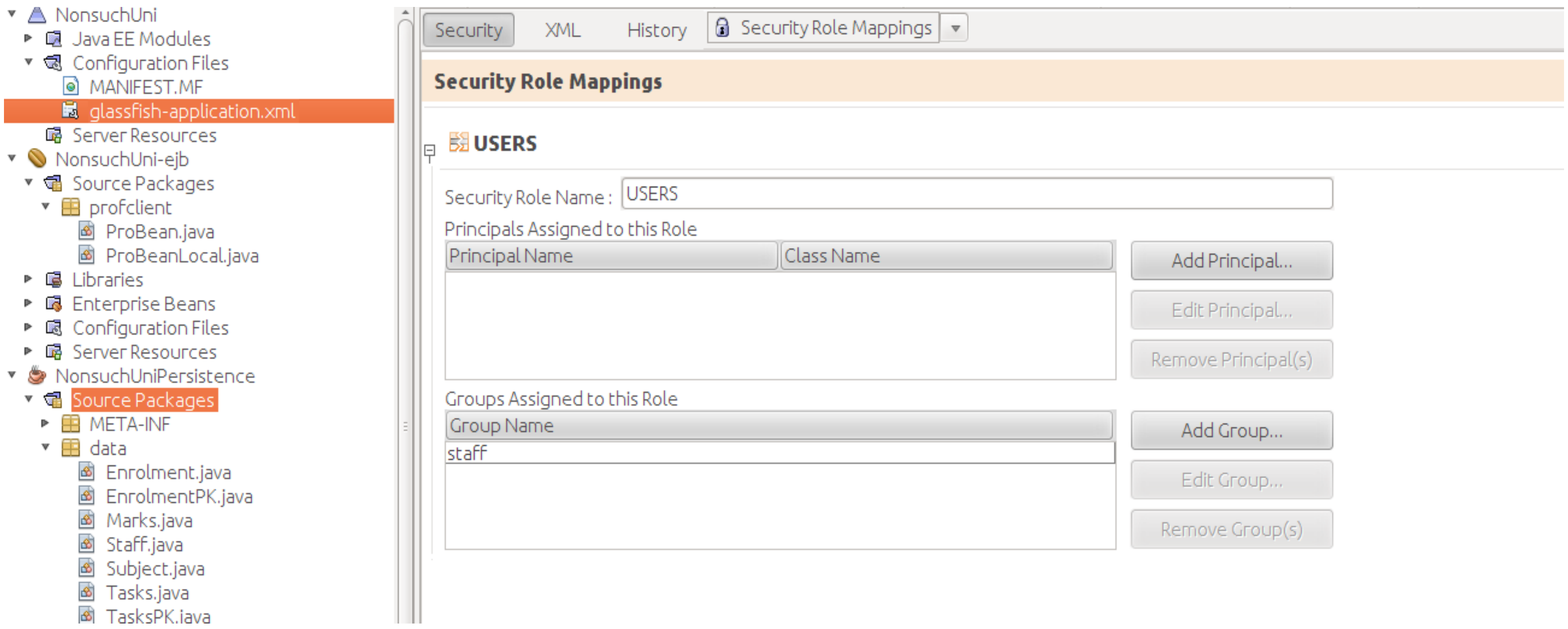
Configuration Name: default-config

Realm Name: file

File Users (2)

User ID	Group List:
manager	staff
da005	staff

in xml USERS has added.



Security Role Mappings

USERS

Security Role Name : USERS

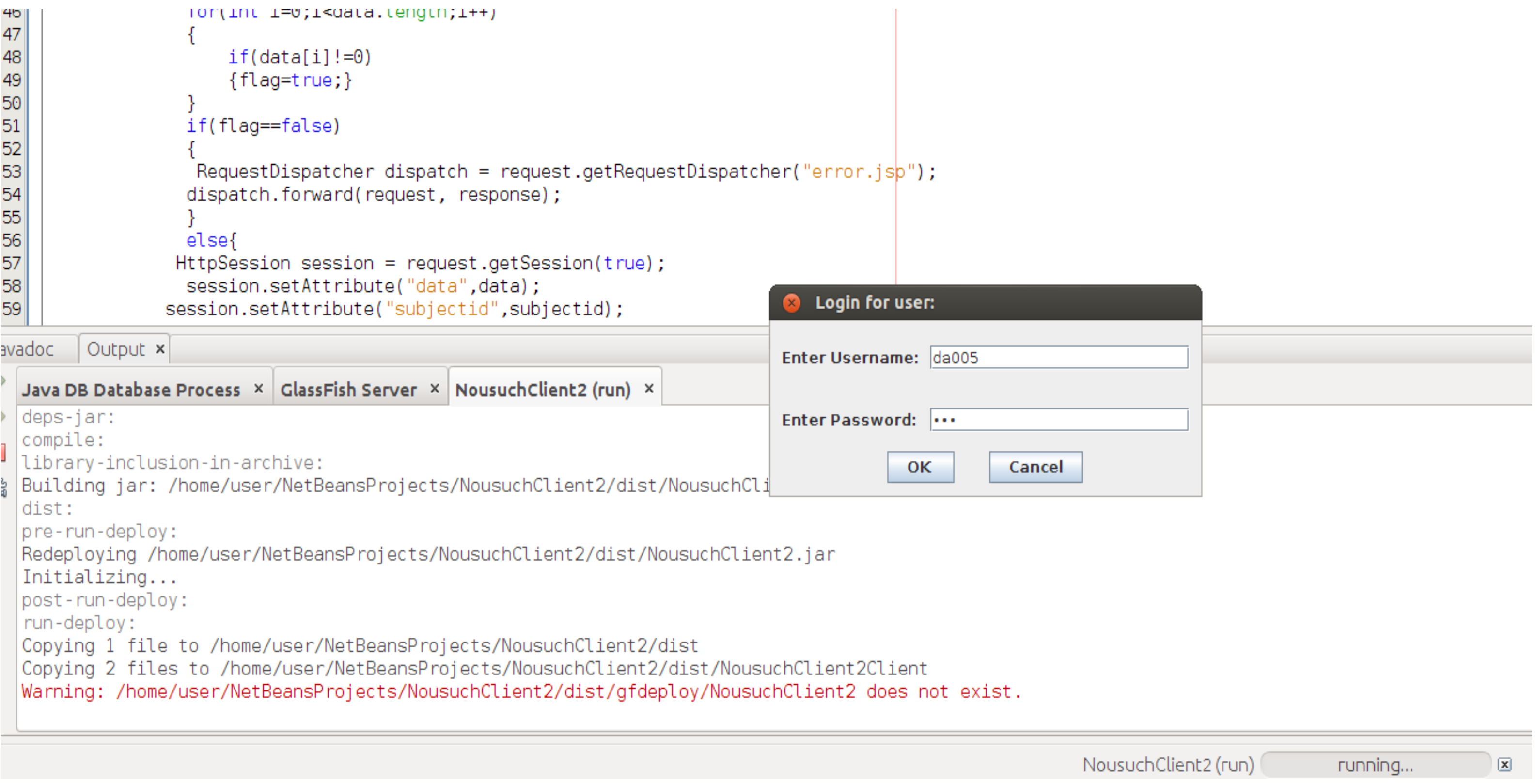
Principals Assigned to this Role

Principal Name	Class Name

Groups Assigned to this Role

Group Name
staff

Log in window



```
            for(int i=0;i<data.length;i++)
            {
                if(data[i]!=0)
                {flag=true;}
            }
            if(flag==false)
            {
             RequestDispatcher dispatch = request.getRequestDispatcher("error.jsp");
            dispatch.forward(request, response);
            }
            else{
           HttpSession session = request.getSession(true);
            session.setAttribute("data",data);
          session.setAttribute("subjectid",subjectid);
```

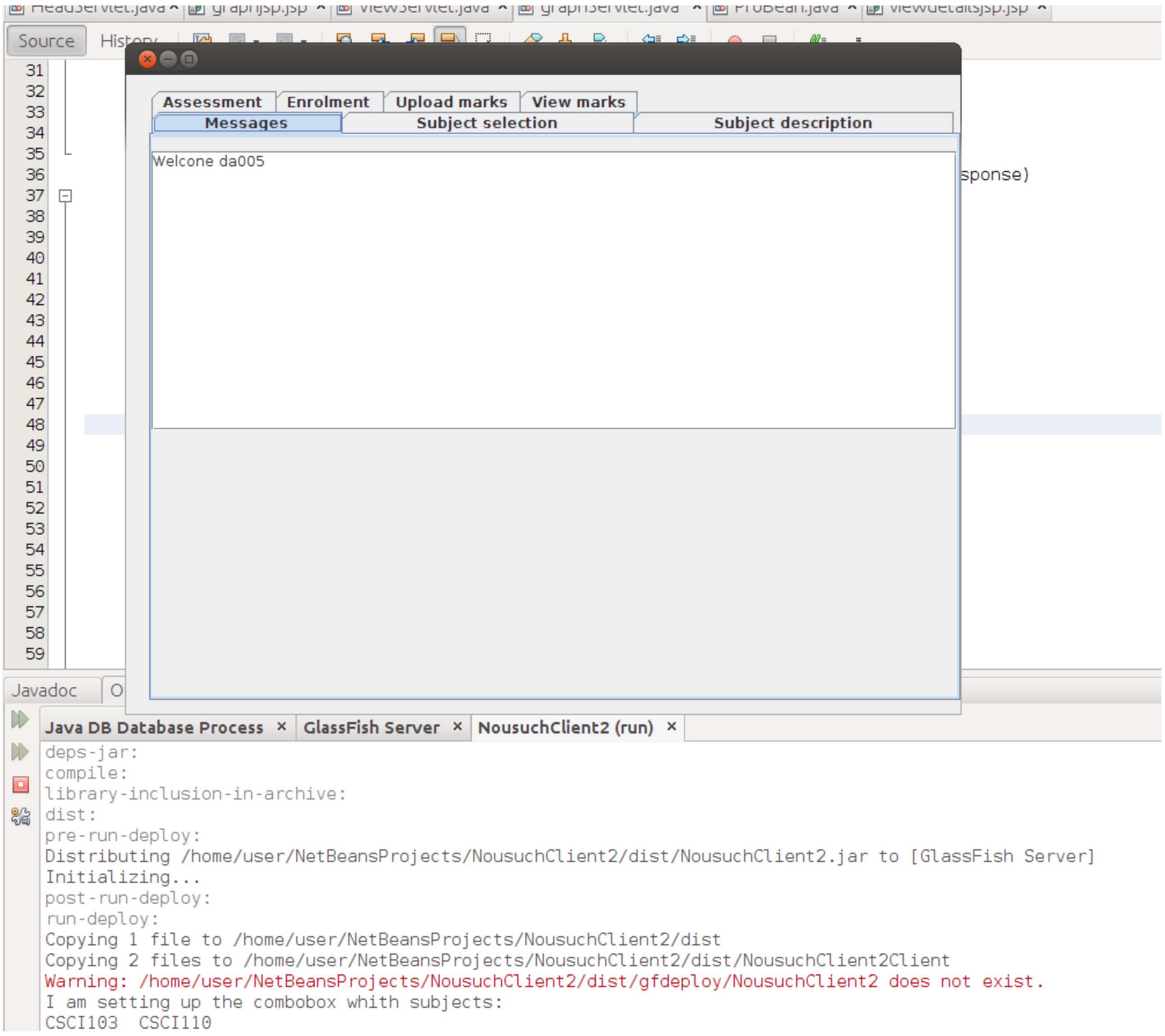
```
deps-jar:
compile:
library-inclusion-in-archive:
Building jar: /home/user/NetBeansProjects/NousuchClient2/dist/NousuchCli
dist:
pre-run-deploy:
Redeploying /home/user/NetBeansProjects/NousuchClient2/dist/NousuchClient2.jar
Initializing...
post-run-deploy:
run-deploy:
Copying 1 file to /home/user/NetBeansProjects/NousuchClient2/dist
Copying 2 files to /home/user/NetBeansProjects/NousuchClient2/dist/NousuchClient2Client
Warning: /home/user/NetBeansProjects/NousuchClient2/dist/gfdeploy/NousuchClient2 does not exist.
```

Login for user:

Enter Username: da005

Enter Password: •••

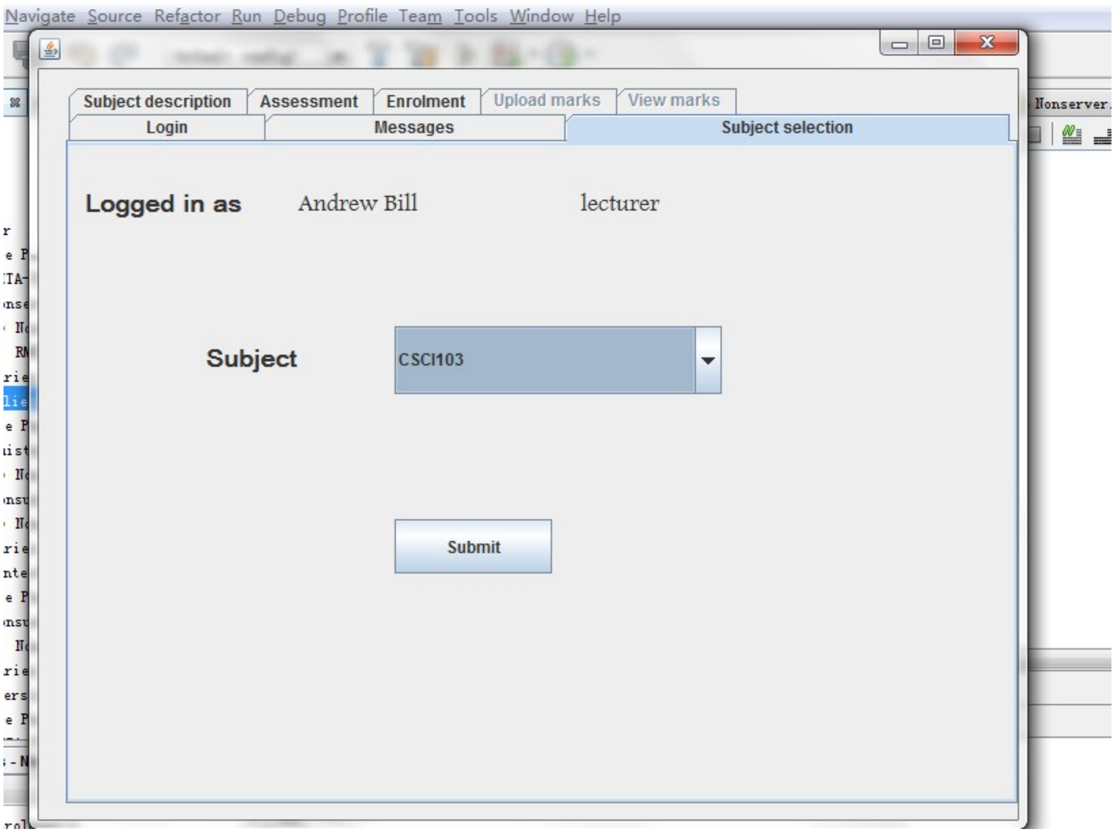
new pannel



Welcone da005

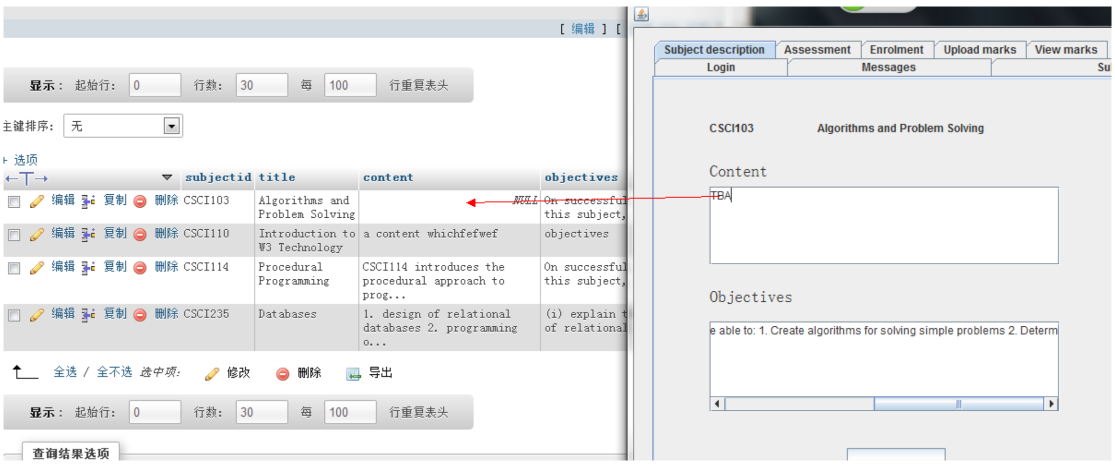
```
deps-jar:
compile:
library-inclusion-in-archive:
dist:
pre-run-deploy:
Distributing /home/user/NetBeansProjects/NousuchClient2/dist/NousuchClient2.jar to [GlassFish Server]
Initializing...
post-run-deploy:
run-deploy:
Copying 1 file to /home/user/NetBeansProjects/NousuchClient2/dist
Copying 2 files to /home/user/NetBeansProjects/NousuchClient2/dist/NousuchClient2Client
Warning: /home/user/NetBeansProjects/NousuchClient2/dist/gfdeploy/NousuchClient2 does not exist.
I am setting up the combobox whith subjects:
CSCI103  CSCI110
```

the rest of the shot will be the same as A1



Logged in as Andrew Bill lecturer

Subject: CSCI103



显示： 起始行: 0 行数: 30 每 100 行重复表头

主键排序: 无

	subjectid	title	content	objectives
编辑 复制 删除	CSCI103	Algorithms and Problem Solving	NULL	On successful this subject,
编辑 复制 删除	CSCI110	Introduction to W3 Technology	a content whichfefwef	objectives
编辑 复制 删除	CSCI114	Procedural Programming	CSCI114 introduces the procedural approach to prog...	On successful this subject,
编辑 复制 删除	CSCI235	Databases	1. design of relational databases 2. programming o...	(i) explain t of relational

显示： 起始行: 0 行数: 30 每 100 行重复表头

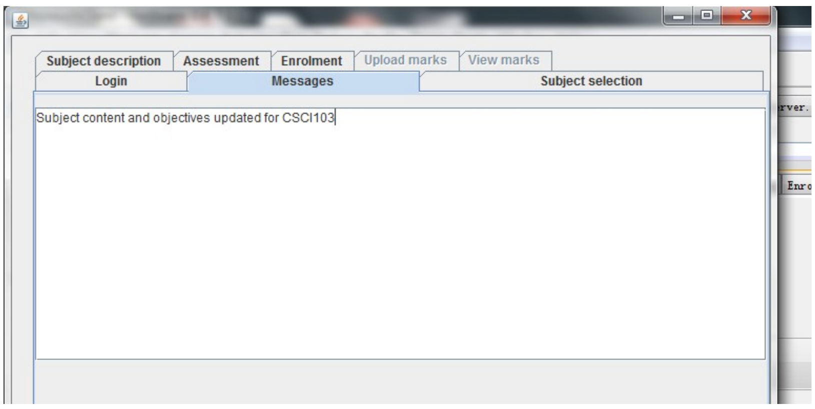
CSCI103 Algorithms and Problem Solving

Content

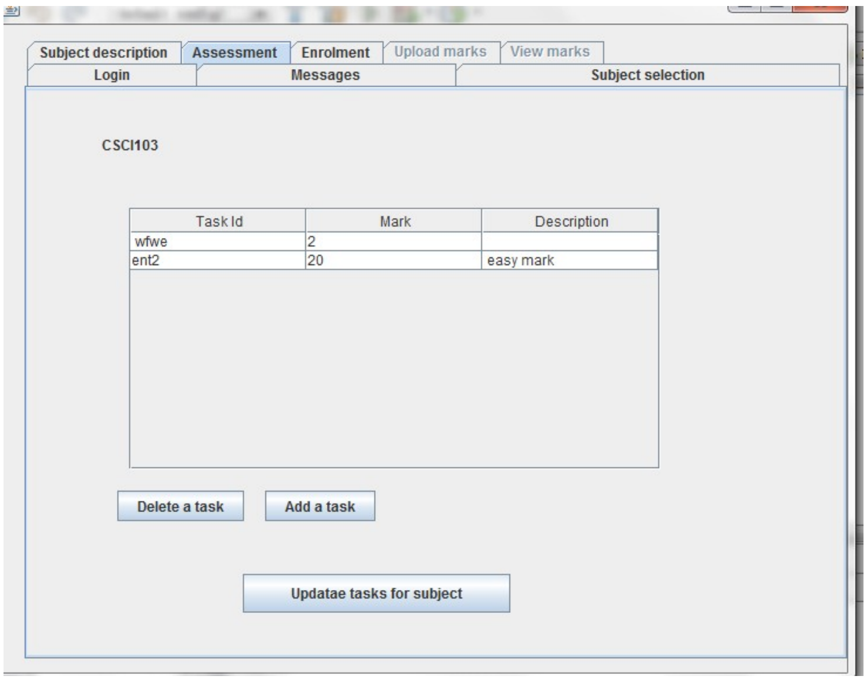
TBA

Objectives

e able to: 1. Create algorithms for solving simple problems 2. Determ

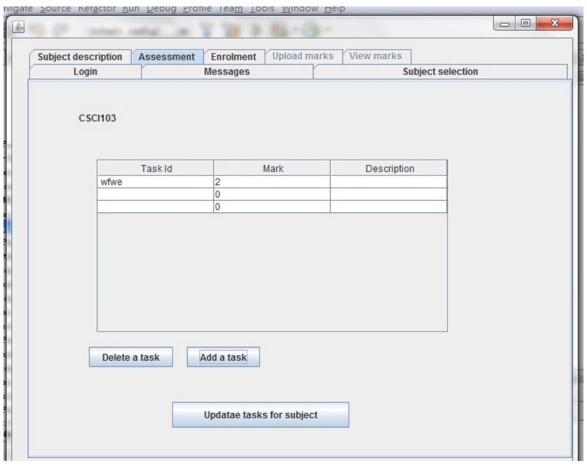


Subject content and objectives updated for CSCI103



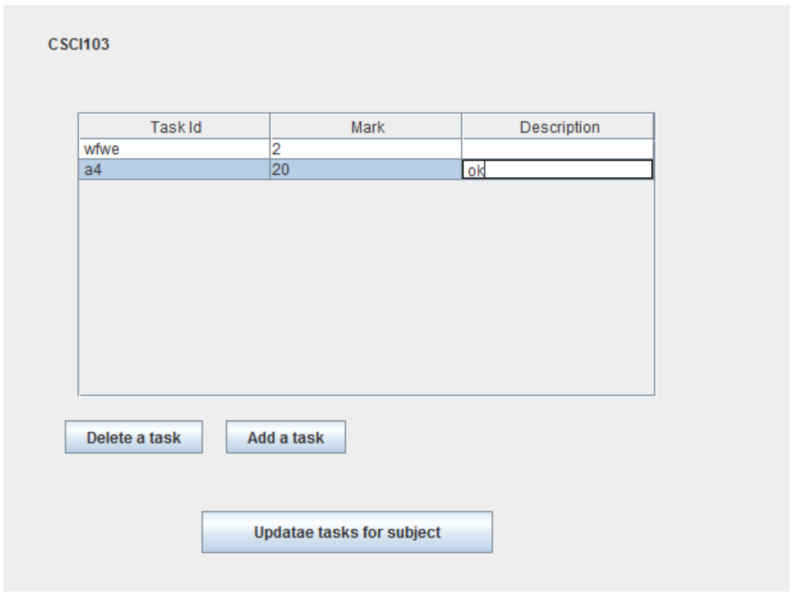
CSCI103

Task Id	Mark	Description
wfwe	2	
ent2	20	easy mark



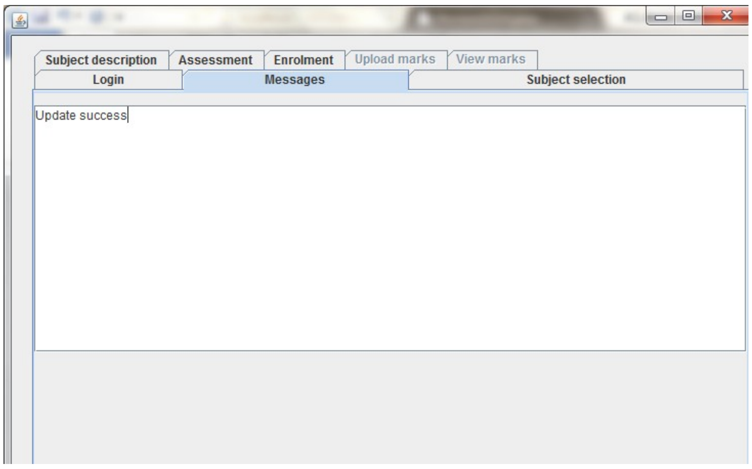
CSCI103

Task Id	Mark	Description
wfwe	2	
	0	
	0	

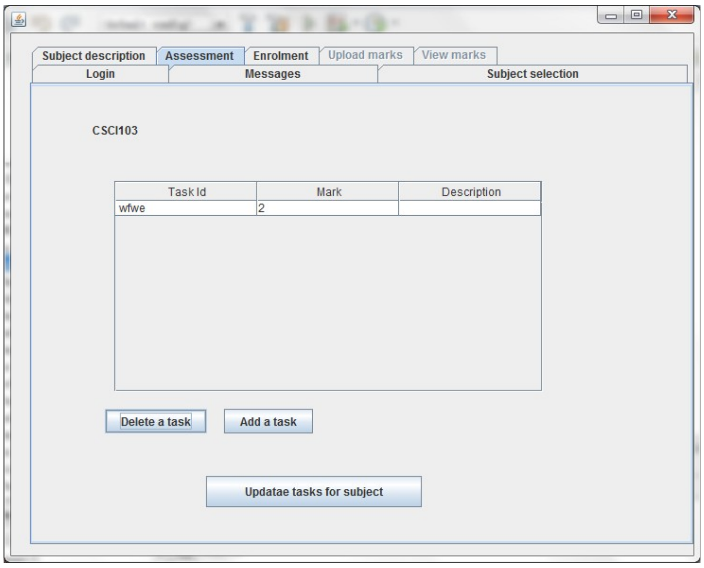


CSCI103

Task Id	Mark	Description
wfwe	2	
a4	20	ok

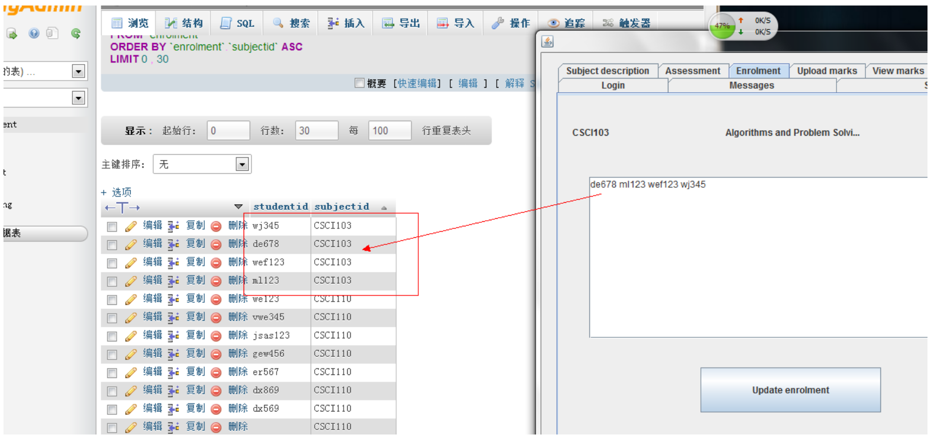


```
Update success
```



CSCI103

Task Id	Mark	Description
wfwe	2	



```
ORDER BY `enrolment`.`subjectid` ASC
LIMIT 0 , 30
```

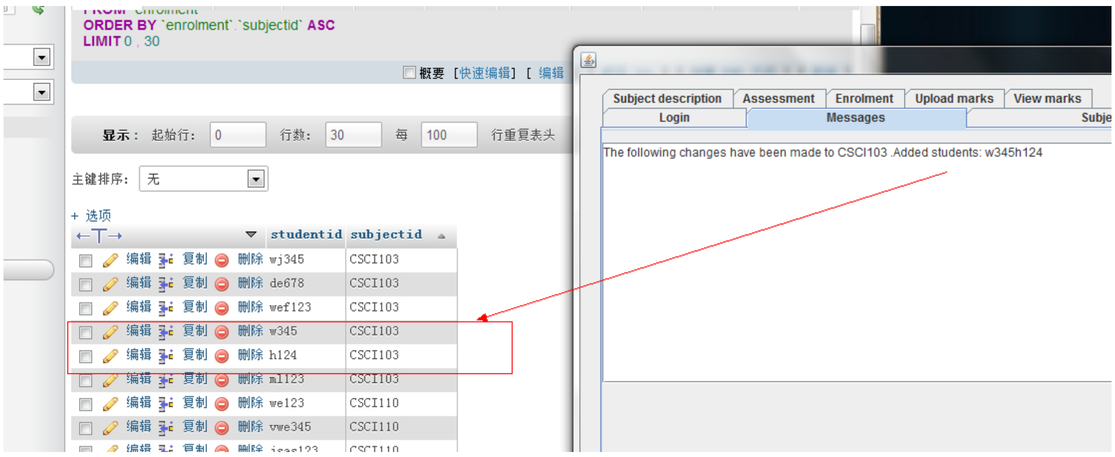
显示： 起始行： 0 行数： 30 每 100 行重复表头

主键排序： 无

	studentid	subjectid
编辑 复制 删除	wj345	CSCI103
编辑 复制 删除	de678	CSCI103
编辑 复制 删除	wef123	CSCI103
编辑 复制 删除	ml123	CSCI103
编辑 复制 删除	we123	CSCI110
编辑 复制 删除	vwe345	CSCI110
编辑 复制 删除	jsas123	CSCI110
编辑 复制 删除	gew456	CSCI110
编辑 复制 删除	er567	CSCI110
编辑 复制 删除	dx869	CSCI110
编辑 复制 删除	dx569	CSCI110
编辑 复制 删除		CSCI110

CSCI103 Algorithms and Problem Solvi...

```
de678 ml123 wef123 wj345
```



```
ORDER BY `enrolment`.`subjectid` ASC
LIMIT 0 , 30
```

概要 [快速编辑] [编辑

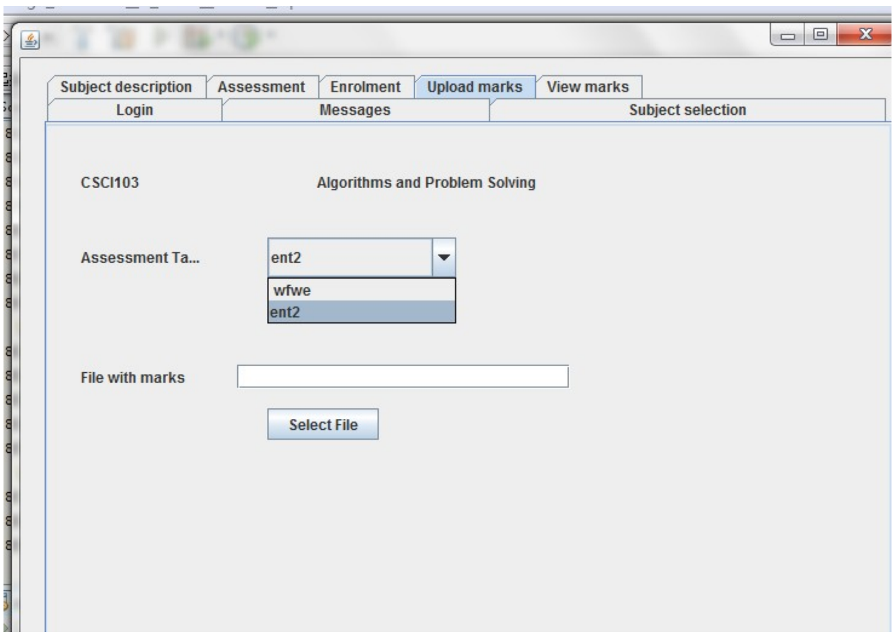
显示： 起始行： 0 行数： 30 每 100 行重复表头

主键排序： 无

+ 选项

	studentid	subjectid
编辑 复制 删除	wj345	CSCI103
编辑 复制 删除	de678	CSCI103
编辑 复制 删除	wef123	CSCI103
编辑 复制 删除	w345	CSCI103
编辑 复制 删除	h124	CSCI103
编辑 复制 删除	ml123	CSCI103
编辑 复制 删除	we123	CSCI110
编辑 复制 删除	vwe345	CSCI110
编辑 复制 删除	jsas123	CSCI110

The following changes have been made to CSCI103 .Added students: w345h124

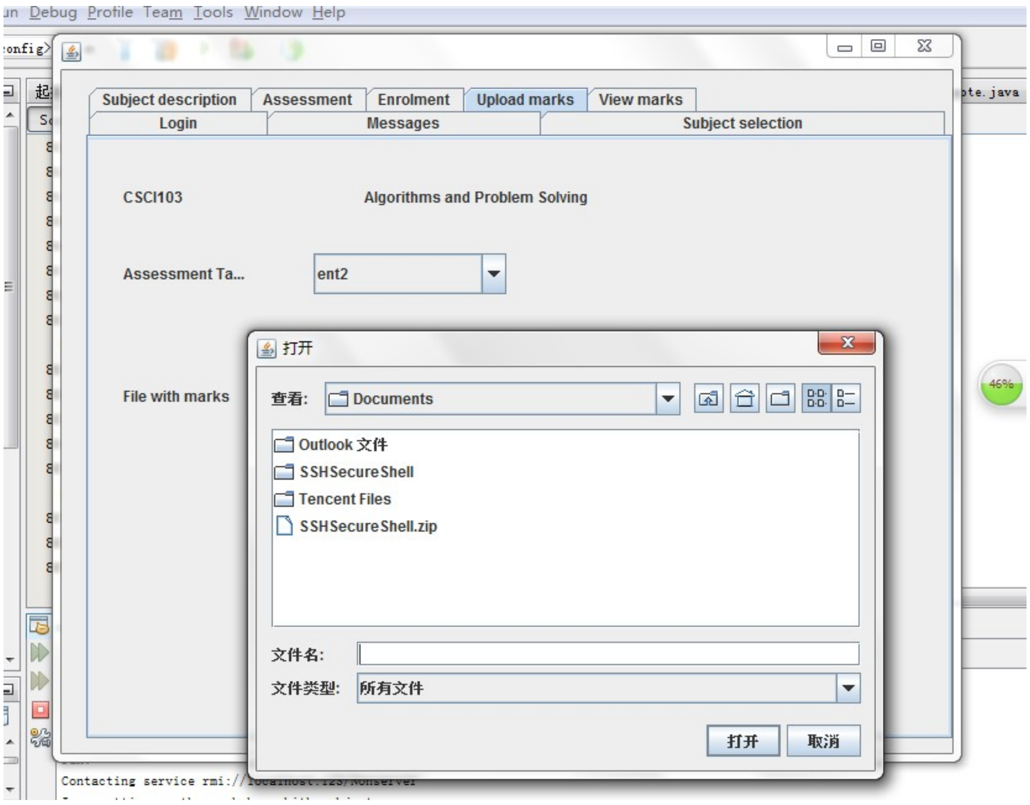


CSCI103 Algorithms and Problem Solving

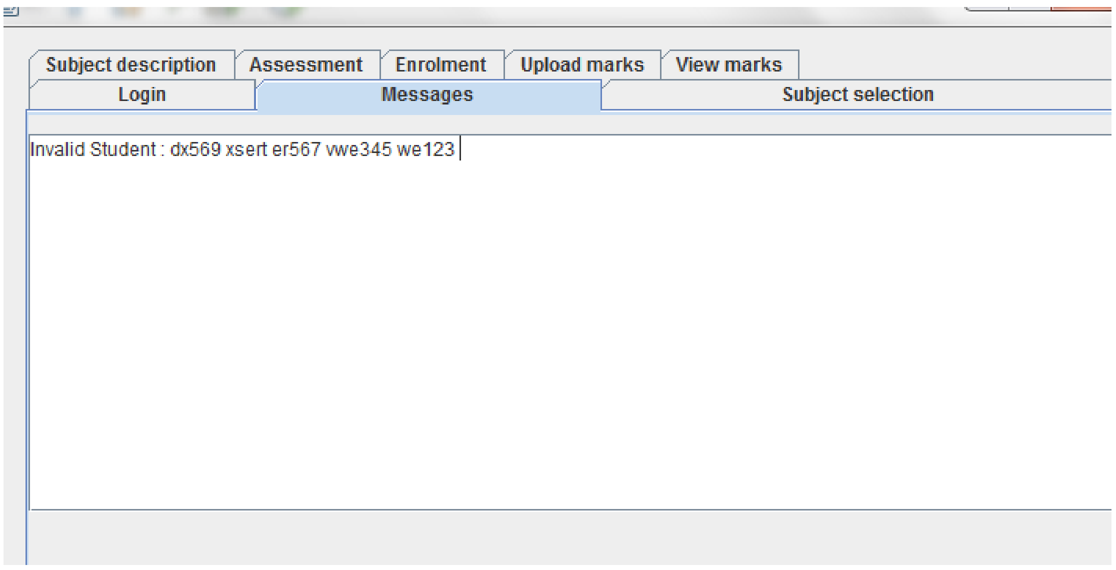
Assessment Ta... ent2

wfwe
ent2

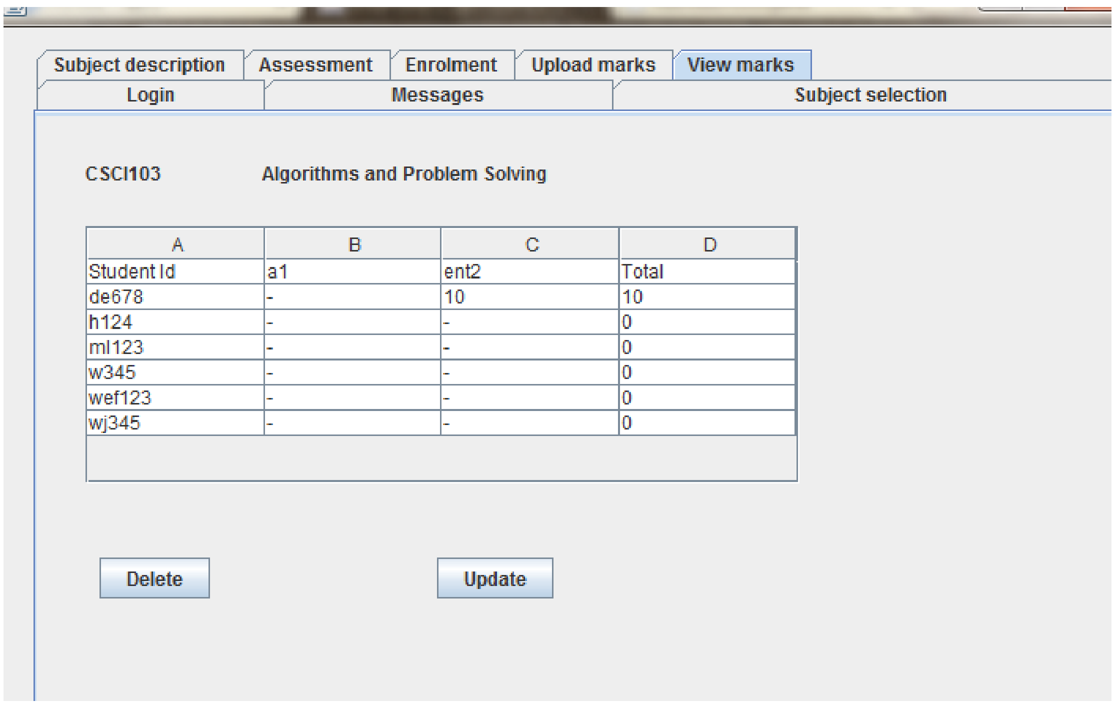
File with marks



```
Contacting service rmi://localhost:123/NonServer
```



```
Invalid Student : dx569 xsert er567 vwe345 we123
```



CSCI103 Algorithms and Problem Solving

A	B	C	D
Student Id	a1	ent2	Total
de678	-	10	10
h124	-	-	0
ml123	-	-	0
w345	-	-	0
wef123	-	-	0
wj345	-	-	0

Code

/home/user/NetBeansProjects/NousuchClient2/src/java/nousuchclient2/Main.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package nousuchclient2;

import javax.ejb.EJB;
import profclient.ProBeanRemote;

/**
 *
 * @author user
 */
public class Main {
    @EJB
    public static ProBeanRemote proBean;

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {

             while(proBean.loginGUI()==null){}
                //show the GUI

                launchGUI(proBean.getStaff(),proBean);

```

```java
    }
        private static void launchGUI(final String s,final ProBeanRemote pr){
        java.awt.EventQueue.invokeLater(new Runnable(){
            @Override
            public void run(){
                //show the GUI
                new NonsuchGUI(s,pr).setVisible(true);
            }
        });
    }
}

```

/home/user/NetBeansProjects/NousuchClient2/src/java/nousuchclient2/NonsuchGUI.java

```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package nousuchclient2;

import ...
/**
 *
 * @author dx869
 */
public class NonsuchGUI extends javax.swing.JFrame {


    private static ProBeanRemote proBean;
    private static String staffid;
    private static String subjectId;
    private static Subject subject;
    private DefaultTableModel dtm;
    private DefaultTableModel mt;

    /**
     * Creates new form NonsuchGUI
     */
    public NonsuchGUI(String s, ProBeanRemote pb) {
        initComponents();
        proBean=pb;
        staffid = s;
        String login = "Welcone " + staffid;
        message.setText(login);
        loginPane.setSelectedComponent(Messages);
        // set up the select subject panel

        role.setText(s);
        //find subject.

        Collection<Subject> subjects=proBean.staffSubjects(s);

        if (subjects != null) {
            int a = subjects.size();
            Subject[] sArray = new Subject[a];
            subjects.toArray(sArray);
            String[] array = new String[a];
            System.out.println("I am setting up the combobox whith
subjects: ");
            for (int i = 0; i < a; i++) {
                array[i] = sArray[i].getSubjectid();
                System.out.print(array[i] + "  ");

            }
            selectSubject.setModel(new javax.swing.DefaultComboBoxModel(array));
        }

    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {}// </editor-fold>

    private void ngloginAncestorMoved(javax.swing.event.AncestorEvent evt) {
        // TODO add your handling code here:
    }

    private void selectSubjectActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void selectSubjectVetoableChange(java.beans.PropertyChangeEvent evt)throws java.beans.PropertyVetoException {
        // TODO add your handling code here:
    }

    private void selectedActionPerformed(java.awt.event.ActionEvent evt) {
        //get the subject id then settle the rest
        Object o = selectSubject.getSelectedItem();
        subjectId = o.toString();

        subject = proBean.details(subjectId).get(0);

        String id = subjectId;
        String title = subject.getTitle();
        String content = subject.getContent();
        String objectives = subject.getObjectives();

        if (content == null) {
            content = "TBA";
        }
        if (objectives == null) {
            content = "TBA";
        }
        subjectidLabel.setText(id);
        subjectnameLabel.setText(title);
        ContentText.setText(content);
        objectiveText.setText(objectives);
        loginPane.setEnabledAt(6, false);
        setAssessment();
```

```java
        setEnrolment();
        setUploadMarks();



    }
    // initalize the upload marks pane

    private void setUploadMarks() {
        String title = subject.getTitle();
        subjectidLabel3.setText(subjectId);
        subjectnameLabel3.setText(title);
        List<Tasks> tasks = proBean.taskBysubjectid(subjectId);
        Tasks[] t = new Tasks[tasks.size()];
        tasks.toArray(t);
        if (tasks == null) {
        } else {
            int a = t.length;
            String[] array = new String[a];
            for (int i = 0; i < a; i++) {
                array[i] = t[i].getTasksPK().getTaskid();
            }
            selectMarkBox.setModel(new javax.swing.DefaultComboBoxModel(array));
        }
    }

    //initalize the enrolment panel
    private void setEnrolment() {
        String title = subject.getTitle();
        subjectidLabel4.setText(subjectId);
        subjectnameLabel4.setText(title);
        String[] studentId = proBean.enrolmentInfo(subjectId);
        if (studentId == null) {
            enrolmentText.setText("");
        } else {
            String enrolmentString = "";
            for (int i = 0; i < studentId.length; i++) {
                enrolmentString = enrolmentString + studentId[i] + " ";
            }
            enrolmentText.setText(enrolmentString);
        }
    }
    // initalize the assessment panel

    private void setAssessment() {
        // set label to correct value
        String title = subject.getTitle();
        subjectidLabel2.setText(subjectId);
        subjectnameLabel2.setText(title);

        // get the tasks in subject

        List<Tasks> t = proBean.taskBysubjectid(subjectId);
        Tasks tasks[] = new Tasks[t.size()];
        t.toArray(tasks);
        //display the tasks;
        Vector<String> row;
        dtm = new DefaultTableModel();
        dtm.addColumn("Task Id");
        dtm.addColumn("Mark");
        dtm.addColumn("Description");
        for (int j = 0; j < tasks.length; j++) {
            row = new Vector();
            row.add(tasks[j].getTasksPK().getTaskid());
            row.add(tasks[j].getMark().toString());
            row.add(tasks[j].getDescription());
            dtm.addRow(row);
        }
        taskTable.setModel(dtm);



    }
    private void updateSubjectActionPerformed(java.awt.event.ActionEvent evt) {

        String s = "Subject content and objectives updated for " + subjectId;
        message.setText(s);
        loginPane.setSelectedComponent(Messages);
        //objectiveText.setText(sArray[i].getObjectives());
    }

    private void deletAtaskActionPerformed(java.awt.event.ActionEvent evt) {
        int selIndex = taskTable.getSelectedRow();

        dtm.removeRow(selIndex);
        // TODO add your handling code here:
    }

    private void addAtaskActionPerformed(java.awt.event.ActionEvent evt) {
        Vector<String> row;
        row = new Vector();
        row.add(" ");
        row.add("0");
        row.add(" ");
        dtm.addRow(row);
        // TODO add your handling code here:
    }

    private void updateTasksActionPerformed(java.awt.event.ActionEvent evt) {
        //checking the data
        //add tasks into database
        //show message in the mesage panel.
        int rowCount = dtm.getRowCount();
        boolean flag = true;
        String tasks[][] = new String[rowCount][3];

        //check if the task id or mark is legal;
        for (int i = 0; i < rowCount; i++) {
            tasks[i][0] = dtm.getValueAt(i, 0).toString();
            for (int m = 0; m < i; m++) {
                if (tasks[i][0].equals(tasks[m][0])) {
                    flag = false;
                    break;
                }
            }

            // check if marks is integer use regular
            tasks[i][1] = dtm.getValueAt(i, 1).toString();
            Pattern pattern = Pattern.compile("[0-9]+");
            if (!pattern.matcher(tasks[i][1]).matches()) {
                flag = false;
                break;
            }
            tasks[i][2] = dtm.getValueAt(i, 2).toString();
        }
        if (flag != true) {
            message.setText("Update failed");
            loginPane.setSelectedComponent(Messages);
        } else {
            proBean.deleteAlltasks(subjectId);
            proBean.updateTasks(tasks, subjectId);

        }
        message.setText("Update success");
        loginPane.setSelectedComponent(Messages);


    }

    private void updateEnrolmentActionPerformed(java.awt.event.ActionEvent evt) {

        String s = enrolmentText.getText();
        String[] old = proBean.enrolmentInfo(subjectId);
        String[] updated = s.split(" ");
        String add = "";
        String delete = "";


        // find the which one has been deleted from old studentid array;
        //call function to delete this studentid from database
        //make string for message panel to show
        if (old == null) {
            add = s;
            for (int i = 0; i < updated.length; i++) {
                proBean.addEnrolment(updated[i], subjectId);
            }
        } else {
            for (int i = 0; i < old.length; i++) {
                boolean flag = false;
                for (int j = 0; j < updated.length; j++) {
                    if (old[i].equals(updated[j])) {
                        flag = true;
                    }
                }
                if (flag == false) {
                    delete = delete + old[i];
                    proBean.deleteEnrolment(old[i], subjectId);
                }
            }
            // find the which one has been added from updated studentid array;
            //call function to add this studentid from database
            //make string for message panel to show
            for (int i = 0; i < updated.length; i++) {
                boolean flag = false;
                for (int j = 0; j < old.length; j++) {
                    if (updated[i].equals(old[j])) {
                        flag = true;
                    }
                }
                if (flag == false) {
                    add = add + updated[i];
                    proBean.addEnrolment(updated[i], subjectId);

                }
            }
        }

        //show the message
        String aMessage = "";
        if (add == null && delete == null) {
            aMessage = "There is no change for your update";
        } else {
            aMessage = "The following changes have been made to " + subjectId + " .";
            if (!add.equals("")) {
                aMessage = aMessage + "Added students: " + add;
            }
            if (!delete.equals("")) {
                aMessage = aMessage + "Removed students: " + delete;
            }
        }
        message.setText(aMessage);
        loginPane.setSelectedComponent(Messages);
        setViewMarks();



        // the rest two panel is available now
        loginPane.setEnabledAt(6, true);
        loginPane.setEnabledAt(7, true);

        setUploadMarks();

    }
    public void setViewMarks() {
        String title = subject.getTitle();
        subjectidLabel5.setText(subjectId);
        subjectnameLabel5.setText(title);
        String[] studentId = proBean.enrolmentInfo(subjectId);
        String nothing = "-";
        //display the tasks;

        List<Tasks> tasklist = proBean.taskBysubjectid(subjectId);
        Tasks[] tasks = new Tasks[tasklist.size()];
        tasklist.toArray(tasks);
        mt = new DefaultTableModel();
        mt.setColumnCount((tasks.length + 2));
        mt.setRowCount((studentId.length + 1));
        markTable.setModel(mt);
        mt.setValueAt("Student Id", 0, 0);
        mt.setValueAt("Total", 0, (tasks.length + 1));
        // first lay of for-loop is find the taskid and put to title.
        // second lay of for-loop is when tasks id id known, get the student mars set as a row

        //set the sutdent id in first column.
        int j = 1;
        for (int i = 0; i < studentId.length; i++) {
            mt.setValueAt(studentId[i], (i + 1), 0);
        }
        for (Tasks t : tasks) {
            mt.setValueAt(t.getTasksPK().getTaskid(), 0, j);

            for (int i = 0; i < studentId.length; i++) {
                int mark = proBean.searchMark(studentId[i], subjectId, t.getTasksPK().getTaskid());
                if (mark == 101) {
                    mt.setValueAt(nothing, (i + 1), j);
                } else {
                    mt.setValueAt(mark, (i + 1), j);
                }
            }
            j++;
        }

        for (int n = 1; n < studentId.length + 1; n++) {
            int total = 0;
            for (int m = 1; m < (tasks.length + 1); m++) {
                String totalS;
                totalS = mt.getValueAt(n, m).toString();
                if (!totalS.equals("-")) {
                    total = Integer.parseInt(totalS);
                }
            }
            mt.setValueAt(total, n, tasks.length + 1);
        }


        markTable.setModel(mt);



    }
//get file and read file deal with file...
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        InputStream is = null;
        String invalidStudent = "Invalid Student : ";
        Object o = selectMarkBox.getSelectedItem();
        String taskid = o.toString();
        boolean exit;
        int mark;

        //Create a file chooser
        final JFileChooser fc = new JFileChooser();
        int returnVal = fc.showOpenDialog(loginPane);
        File file = fc.getSelectedFile();

        //change text field 's value
        fileName.setText(file.getPath());
        try {
            //open and read the file
            is = new FileInputStream(file);
        } catch (FileNotFoundException ex) {
            Logger.getLogger(NonsuchGUI.class.getName()).log(Level.SEVERE, null, ex);
        }
        InputStreamReader isr = new InputStreamReader(is);
        BufferedReader reader = new BufferedReader(isr);
        try {
            String text = reader.readLine();
            while (text != null) {
                String[] pairs = text.split(" ");

```

```java

                //check if the sutdent is exit and enroled in this subject
                exit = proBean.enrolmented(pairs[0], subjectId);
                if (exit) {
                    mark = Integer.parseInt(pairs[1]);

                    //check if the mark is less than the total mark in the task.
                    if (mark <= proBean.getTask(taskid, subjectId)) {
                        proBean.insertMark(subjectId, taskid, pairs[0], mark);
                    } else {
                        invalidStudent = invalidStudent + pairs[0] + " ";
                    }
                } else {
                    invalidStudent = invalidStudent + pairs[0] + " ";
                }
                text = reader.readLine();
            }
            message.setText(invalidStudent);
            loginPane.setSelectedComponent(Messages);

        } catch (IOException ex) {
            Logger.getLogger(NonsuchGUI.class.getName()).log(Level.SEVERE, null, ex);
        }


        //



    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

        int columncount = markTable.getSelectedColumn();
        String taskid = markTable.getValueAt(0, columncount).toString();

        proBean.deleteTask(taskid);
        TableColumnModel columnModel = markTable.getColumnModel();
        TableColumn tableColumn = columnModel.getColumn(columncount);
        columnModel.removeColumn(tableColumn);

    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        int i = 0;

        int row[] = new int[markTable.getRowCount()];
        int col[] = new int[markTable.getColumnCount()];

        //when it is editing then get the rownumber and column number , stop editing.
        if (markTable.isEditing()) {
            row[i] = markTable.getEditingRow();
            col[i] = markTable.getEditingColumn();
            markTable.getCellEditor().stopCellEditing();


            //get task id & student id & mark;
            String taskid = markTable.getValueAt(0, col[i]).toString();
            String studentid = markTable.getValueAt(row[i], 0).toString();
            String markS = markTable.getValueAt(row[i], col[i]).toString();

            int mark = Integer.parseInt(markS);


            //call remote method to update


            proBean.updateMark(taskid, studentid, subjectId, mark);

        }


        // TODO add your handling code here:
    }
    /**
     * @param args the command line arguments
     */
    /* public static void main(String args[]) {
     /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */


     /* Create and display the form */
    /*    java.awt.EventQueue.invokeLater(new Runnable() {
     public void run() {
     new NonsuchGUI().setVisible(true);
     }
     });
     }*/
```

/home/user/NetBeansProjects/NonsuchUni/NonsuchUni-ejb/src/java/profclient/ProBean.java

```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package profclient;

import ...

/**
 *
 * @author user
 */
@Stateless
public class ProBean implements ProBeanRemote, ProBeanLocal {

    @Resource
    private SessionContext sctx;
    @PersistenceContext(unitName = "NonsuchUni-ejbPU")
    private EntityManager em;
    private Staff staff;

    @Override
    public List<Subject> listSubject() {
        Query q = em.createNamedQuery("Subject.findAll");
        List<Subject> results = q.getResultList();
        return results;
    }

    @Override
    public List<Subject> viewDetails() {
        return null;
    }

    @Override
    public List<Subject> details(String String) {
        Query q = em.createNamedQuery("Subject.findBySubjectid");
        q.setParameter("subjectid", String);
        List<Subject> results = q.getResultList();
        return results;

    }

    @Override
    public Staff login(String name, String pwd) {
        Query q = em.createNamedQuery("Staff.findByStaffid");
        q.setParameter("staffid", "da005");
        staff = (Staff) q.getSingleResult();

        return staff;
    }

    @Override
    public String getMD5(String str) {
        try {
            //Create MessageDigest object for MD5
            MessageDigest digest = MessageDigest.getInstance("MD5");
            //Update input string in message digest
            digest.update(str.getBytes(), 0, str.length());
            //Converts message digest value in base 16 (hex)
            String md5Pwd = new BigInteger(1, digest.digest()).toString(16);

            return md5Pwd;
        } catch (NoSuchAlgorithmException ex) {
            Logger.getLogger(ProBean.class.getName()).log(Level.SEVERE,
null, ex);
        }
        return null;
    }

    @Override
    public Staff staffInfo() {

        return staff;
    }

    @Override
    public boolean Logout() {
        staff = null;
        return true;
    }

    @Override
    public Collection<Subject> staffSubject() {

        if (staff != null) {
            Collection<Subject> results = staff.getSubjectCollection();
            return results;
        } else {
            return null;
        }
    }

    @Override
    public HashMap mapSubject() {
        if (staff == null) {
            return null;
        }
        HashMap map = new HashMap();

        List<Subject> subjects = (List<Subject>) staff.getSubjectCollection();

        for (Subject s : subjects) {
            List<Tasks> tasks = (List<Tasks>) s.getTasksCollection();
            map.put(s.getSubjectid(), tasks);
        }
        return map;

    }

    @Override
    public int[] graph(String subjectid, String task) {

        try {
            Query q = em.createQuery("select m from Marks m where m.subjectid =:subjectid and m.taskid=:taskid");
            q.setParameter("taskid", task);
            q.setParameter("subjectid", subjectid);

            List<Marks> marks = (List<Marks>) q.getResultList();
            em.clear();

            int a[] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
           Query q2 = em.createQuery("SELECT t FROM Tasks t WHERE t.tasksPK.subjectid=:subjectid AND t.tasksPK.taskid = :taskid");
            q2.setParameter("subjectid", subjectid);
            q2.setParameter("taskid", task);
            Tasks t = (Tasks) q2.getSingleResult();
            int mark = t.getMark();

            for (Marks m: marks) {
                int studentMark = m.getMark();
                int level = studentMark * 10 / mark;
                a[level] = a[level] + 1;
            }


            return a;
        } catch (Exception e) {
            return null;
        }
    }

    @Override
    public List<Tasks> taskBysubjectid(String subjectid) {
        Query q = em.createNamedQuery("Tasks.findBySubjectid");
        q.setParameter("subjectid", subjectid);
        List<Tasks> tasks = (List<Tasks>) q.getResultList();

        return tasks;
    }

    @Override
    @RolesAllowed({"USERS"})
    public String[] enrolmentInfo(String subjectId) {
        Query q = em.createNamedQuery("Enrolment.findBySubjectid");
        q.setParameter("subjectid", subjectId);
        List<Enrolment> enrolment = (List<Enrolment>) q.getResultList();
        int num = enrolment.size();
        int i = 0;
        if (num == 0) {
            em.clear();
            return null;
        }
        String[] studentId = new String[num];
        for (Enrolment e : enrolment) {
            studentId[i] = e.getEnrolmentPK().getStudentid().toString();
            i++;
        }
        em.clear();
        return studentId;
    }

    @Override
    @RolesAllowed({"USERS"})
    public void deleteAlltasks(String subjectid) {
        if (!em.getTransaction().isActive()) {
            em.getTransaction().begin();
        }
        Query q = em.createNamedQuery("Tasks.findBySubjectid");
        q.setParameter("subjectid", subjectid);
        List<Tasks> tasks = (List<Tasks>) q.getResultList();
        Tasks[] t = new Tasks[tasks.size()];
        tasks.toArray(t);
        for (int j = 0; j < tasks.size(); j++) {
            em.remove(t[j]);
        }
        em.getTransaction().commit();
        em.clear();
    }

    @Override
    @RolesAllowed({"USERS"})
    public void updateTasks(String[][] tasks, String subjectId) {
        int num = tasks.length;
        for (int i = 0; i < num; i++) {
            if (!em.getTransaction().isActive()) {
                em.getTransaction().begin();
            }
            Tasks t = new Tasks();
            TasksPK pk = new TasksPK();
            pk.setSubjectid(subjectId);
            pk.setTaskid(tasks[i][0]);
            t.setTasksPK(pk);
            int mark = Integer.parseInt(tasks[i][1]);
            t.setMark(mark);
            t.setDescription(tasks[i][2]);
            em.persist(t);
            em.getTransaction().commit();
            em.clear();
        }
    }

    @Override
    @RolesAllowed({"USERS"})
    public void addEnrolment(String updated, String subjectId) {
        if (!em.getTransaction().isActive()) {
            em.getTransaction().begin();
        }
        Enrolment e = new Enrolment();
        EnrolmentPK ePK = new EnrolmentPK();
        ePK.setStudentid(updated);
        ePK.setSubjectid(subjectId);
        e.setEnrolmentPK(ePK);
        em.persist(e);
        em.getTransaction().commit();
        em.clear();
    }

    @Override
    @RolesAllowed({"USERS"})
    public void deleteEnrolment(String studentId, String subjectId) {
        if (!em.getTransaction().isActive()) {
            em.getTransaction().begin();
        }
        Query query = em.createQuery("SELECT e FROM Enrolment e WHERE e.enrolmentPK.studentid = :studentid AND e.enrolmentPK.subjectid = :subjectid");
        query.setParameter("subjectid", subjectId);
        query.setParameter("studentid", studentId);
        List<Enrolment> list = query.getResultList();

        for (Enrolment e : list) {
            em.remove(e);
        }
        em.getTransaction().commit();
        em.clear();
    }

    @Override
    @RolesAllowed({"USERS"})
    public int searchMark(String studentId, String subjectId, String taskId) {
        int mark = 0;
        Query query = em.createQuery("SELECT m FROM Marks m WHERE m.subjectid = :subjectid AND m.taskid = :taskid AND m.studentid = :studentid");
        query.setParameter("subjectid", subjectId);
        query.setParameter("taskid", taskId);
        query.setParameter("studentid", studentId);
        List<Marks> marks = query.getResultList();
        if (marks.isEmpty()) {
            return 101;
        }
        for (Marks m : marks) {
            mark = m.getMark();;
        }
        return mark;

    }

    @Override
    @RolesAllowed({"USERS"})
    public boolean enrolmented(String studentId, String subjectId) {
        boolean flag = false;
        Query q = em.createNamedQuery("Enrolment.findBySubjectid");
        q.setParameter("subjectid", subjectId);
        List<Enrolment> enrolment = (List<Enrolment>) q.getResultList();
        int num = enrolment.size();
        if (num != 0) {
            for (Enrolment e : enrolment) {
                if (e.getEnrolmentPK().getStudentid().equals(studentId)) {
                    flag = true;
                }
            }
        }

        return flag;
    }

    @Override
    public int getTask(String taskid, String subjectid) {
        int mark = 0;
        Query query = em.createQuery("SELECT t FROM Tasks t WHERE
t.tasksPK.subjectid = :subjectid AND t.tasksPK.taskid = :taskid");
        query.setParameter("subjectid", subjectid);
        query.setParameter("taskid", taskid);
        List<Tasks> task = query.getResultList();
        for (Tasks t : task) {
            mark = t.getMark();
        }
        return mark;
    }

    @Override
    @RolesAllowed({"USERS"})
    public void insertMark(String subjectId, String taskid, String
studentId, int mark) {
        Marks aMark = new Marks();
        if (!em.getTransaction().isActive()) {
            em.getTransaction().begin();
        }
        aMark.setMark(mark);
        aMark.setStudentid(studentId);
        aMark.setTaskid(taskid);
        aMark.setSubjectid(subjectId);
        em.persist(aMark);
        em.getTransaction().commit();
        em.clear();
    }

    @Override
    @RolesAllowed({"USERS"})
    public void deleteTask(String taskid) {
        if (!em.getTransaction().isActive()) {
            em.getTransaction().begin();
        }
        Query query = em.createNamedQuery("Tasks.findByTaskid");
        query.setParameter("taskid", taskid);
        List<Tasks> list = query.getResultList();
        if (!list.isEmpty()) {
            for (Tasks t : list) {
                em.remove(t);
            }
            em.getTransaction().commit();
            em.clear();
        }
        Query query1 = em.createNamedQuery("Marks.findByTaskid");
        query1.setParameter("taskid", taskid);

        List<Marks> marks = query1.getResultList();
        if (!marks.isEmpty()) {
            for (Marks m : marks) {
                em.remove(m);
            }
            em.getTransaction().commit();
            em.clear();
        }
    }

    @Override
    @RolesAllowed({"USERS"})
    public void updateMark(String taskid, String studentid, String subjectid, int mark) {
        if (!em.getTransaction().isActive()) {
            em.getTransaction().begin();
        }
        Query query = em.createQuery("SELECT m FROM Marks m WHERE m.studentid = :studentid  AND m.taskid = :taskid");
        query.setParameter("studentid", studentid);
        query.setParameter("taskid", taskid);
        List<Marks> marks = query.getResultList();
        if (!marks.isEmpty()) {
            System.out.println(marks.size());
            for (Marks m : marks) {
                System.out.println(m.getTaskid());
                m.setMark(mark);
            }

        } else {
            Marks mNew = new Marks();
            mNew.setMark(mark);
            mNew.setStudentid(studentid);
            mNew.setSubjectid(subjectid);
            mNew.setTaskid(taskid);
            em.persist(mNew);
        }
        em.getTransaction().commit();
        em.clear();
    }

    @Override
    @RolesAllowed({"USERS"})
    public String loginGUI() {
        try {
            Principal cp = sctx.getCallerPrincipal();
            return cp.getName();
        } catch (Exception e) {
            return null;
        }


    }

    @Override
```

```java
    @RolesAllowed({"USERS"})
    public String getStaff() {
        Principal cp = sctx.getCallerPrincipal();
        Query q = em.createNamedQuery("Staff.findByStaffid");
        q.setParameter("staffid", cp.getName());
        staff = (Staff) q.getSingleResult();
        return cp.getName();
    }

    @Override
    public Collection<Subject> staffSubjects(String staffid) {
        Query q = em.createNamedQuery("Staff.findByStaffid");
        q.setParameter("staffid", staffid);
        staff = (Staff) q.getSingleResult();
        Collection<Subject> subjects = staff.getSubjectCollection();

        return subjects;
    }
}
```