

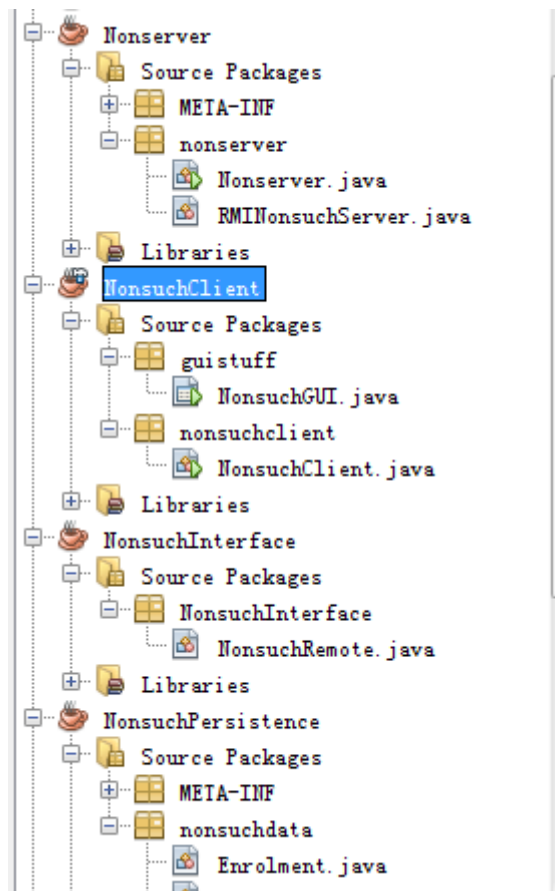
CSCI398 Assignment 1

Student Id: 4208869

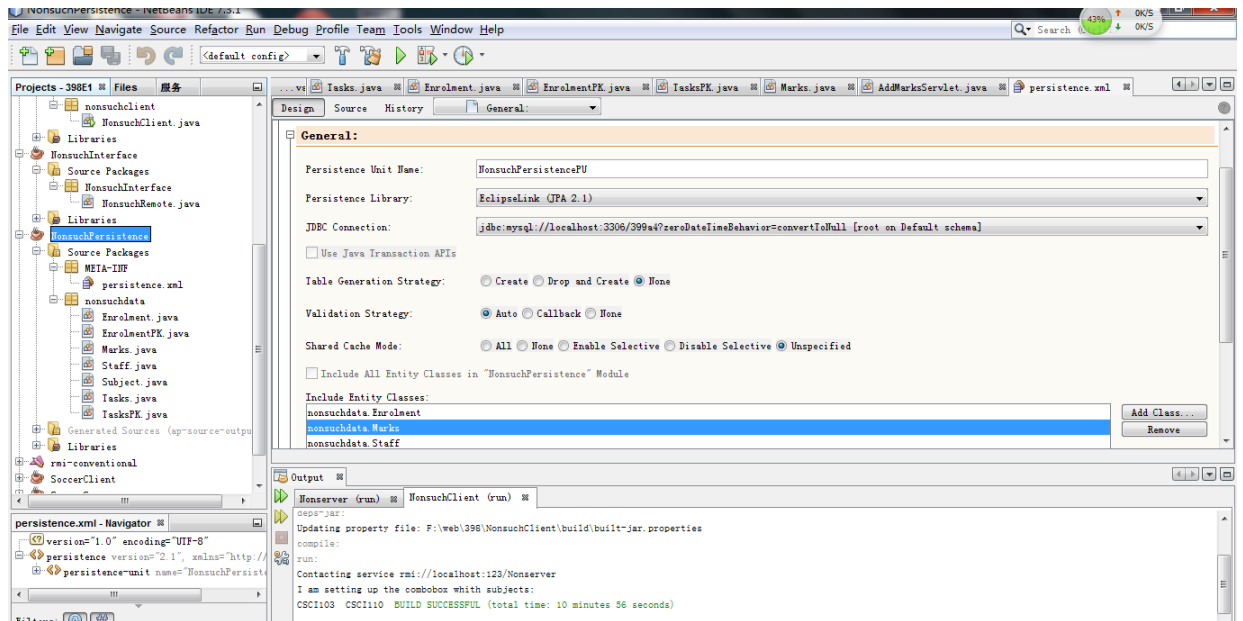
SOLS : dx869

Overview:

4 java files include: NonsuchPersistence , NonsuchInterface, NonsuchClient

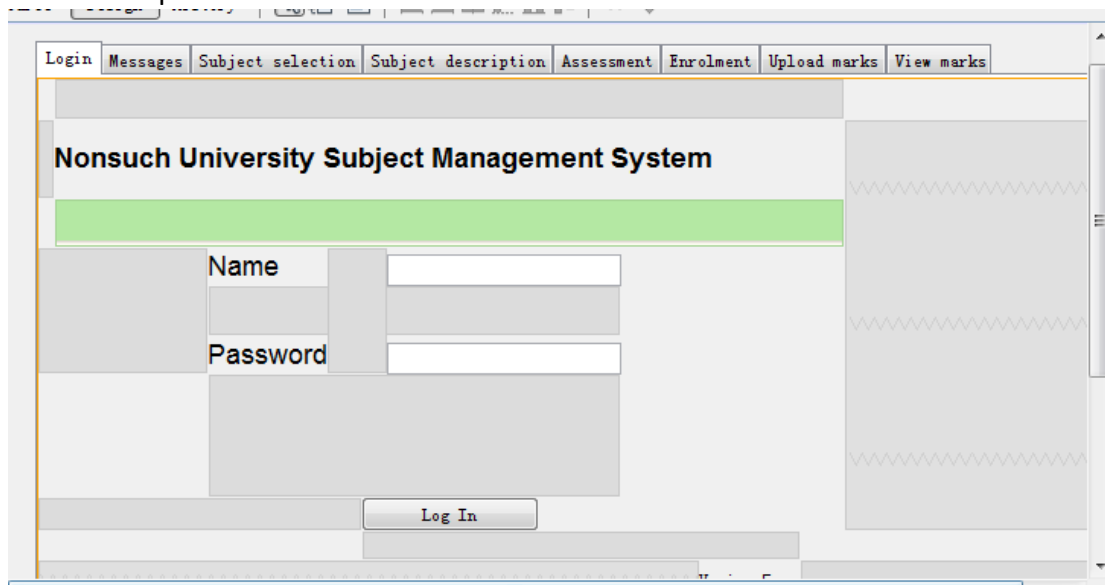


My persistence, using JPA , lazy relationship to access the database.



MY GUI

Same as requested.



Code & Display

Interface: just the an interface

Code:

F:\web\398\NonsuchInterface\src\NonsuchInterface\NonsuchRemote.java

```

1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5 package NonsuchInterface;
6
7 import java.rmi.Remote;
8 import java.rmi.RemoteException;
9 import java.util.Collection;
10 import nonsuchdata.*;
11
12 /**
13  *
14  * @author Administrator
15  */
16 public interface NonsuchRemote extends Remote {
17 //interface , put methods here and put code into Implement (RMINonsuchServer)
later.
18
19     public Staff staffInfo(String id) throws RemoteException;
20
21     public Collection<Subject> getSubjects(String id) throws RemoteException;
22
23     public void setContent(String content, String objectives, String subjectid)
throws RemoteException;
24
25     public Subject subjectInfo(String id) throws RemoteException;
26
27     public Tasks[] getTasks(String id) throws RemoteException;
28
29     public void deleteAlltasks(String subjectId) throws RemoteException;
30
31     public void updateTasks(String[][] tasks, String subjectId) throws
RemoteException;
32
33     public String[] enrolmentInfo(String subjectId) throws RemoteException;
34
35     public void deleteEnrolment(String studentId, String subjectId) throws
RemoteException;
36
37     public void addEnrolment(String update, String subjectId) throws
RemoteException;
38

```

```

39     public boolean enrolmented(String studentId, String subjectId) throws
RemoteException;
40
41     public int getTask(String taskid, String subjectid) throws RemoteException;
42
43     public void insertMark(String subjectId, String taskid, String studentId, int
mark) throws RemoteException;
44
45     public int searchMark(String studentId, String subjectId, String taskId)
throws RemoteException;
46     public void deleteTask(String taskid)throws RemoteException;
47     public void updateMark(String taskid,String studentid,int mark)throws
RemoteException;
48 }
49

```

Nonsuch server

Two java file include

Code

Nonserver.java just run a server waiting for client

```

F:\web\398\Nonserver\src\nonserver\Nonserver.java
1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5 package nonserver;
6
7 import java.rmi.RemoteException;
8 import java.rmi.registry.LocateRegistry;
9 import java.rmi.registry.Registry;
10
11 /**
12  *
13  * @author Administrator
14  */
15 public class Nonserver {

```

```

16
17     private static final int myPort=123;
18     /**
19      * @param args the command line arguments
20      */
21     public static void main(String[] args) {
22         RMINonsuchServer g=null;
23         try{
24             g=new RMINonsuchServer();
25         }catch (Exception re){
26             System.out.println("unable to start up server");
27             System.out.println(re);
28             System.exit(1);
29         }
30         String bindName="Nonserver";
31         Registry rmiNamingService=null;
32         try{
33             // rmi registry , so client can send stub files.
34             rmiNamingService=LocateRegistry.createRegistry(myPort);
35         }catch(RemoteException re){
36             System.out.println(re);
37             System.exit(1);
38         }
39         System.out.println("About to bind :"+ bindName);
40         try{
41             rmiNamingService.rebind(bindName,g);
42         }catch(Exception e){
43             System.out.println(e);
44             System.exit(1);
45         }
46         System.out.println("yes it is working");
47     }
48 }
49 }
50

```

Code

RMINonsuchServer

All the method than is works with database, get data and transfor to client. Or delete or update data as requested.

```

1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5  package nonserver;
6
7  import NonsuchInterface.NonsuchRemote;
8  import java.rmi.RemoteException;
9  import java.rmi.server.UnicastRemoteObject;
10 import java.util.Collection;
11 import java.util.List;
12 import javax.persistence.EntityManager;
13 import javax.persistence.EntityManagerFactory;
14 import javax.persistence.Persistence;
15 import javax.persistence.Query;
16 import nonsuchdata.*;
17
18 /**
19  *
20  * @author Administrator this is the implement all the data methods from
nonsuch
21  * remote will be coded in here.
22  */
23 //linking with RMI rn-time library involves use of class nicastRemoteObject.
24 public class RMINonsuchServer extends UnicastRemoteObject implements
NonsuchRemote {
25
26     private EntityManagerFactory emf;
27     private EntityManager em;
28
29     public RMINonsuchServer() throws RemoteException {
30         try {
31             emf = Persistence.createEntityManagerFactory("NonserverPU");
32             em = emf.createEntityManager();
33
34             // put code here which is already mentioned in nonsuchremote.
35         } catch (Exception e) {
36             System.out.println(e);
37             System.exit(1);
38         }
39
40     }
41 // get staff information just for log in

```

```

42 // @parameter staffid;
43 // @return Staff or null if not find.
44
45     @Override
46     public Staff staffInfo(String id) throws RemoteException {
47
48         Query q = em.createNamedQuery("Staff.findByStaffid");
49         q.setParameter("staffid", id);
50         List<Staff> staff = (List<Staff>) q.getResultList();
51         if (staff.isEmpty()) {
52             em.clear();
53             return null;
54         } else {
55             for (Staff f : staff) {
56                 em.clear();
57                 return f;
58             }
59             return null;
60         }
61     }
62
63
64 }
65
66     @Override
67     public Subject subjectInfo(String id) throws RemoteException {
68         Query q = em.createNamedQuery("Subject.findBySubjectid");
69         q.setParameter("subjectid", id);
70         List<Subject> subject = (List<Subject>) q.getResultList();
71         for (Subject s1 : subject) {
72             return s1;
73         }
74         em.clear();
75         return null;
76     }
77
78     @Override
79     public int getTask(String taskid, String subjectid) throws RemoteException
80 {
81     int mark = 0;
82     Query query = em.createQuery("SELECT t FROM Tasks t WHERE
83 t.tasksPK.subjectid = :subjectid AND t.tasksPK.taskid = :taskid");
84     query.setParameter("subjectid", subjectid);
85     query.setParameter("taskid", taskid);

```

```

84         List<Tasks> task = query.getResultList();
85         for (Tasks t : task) {
86             mark = t.getMark();
87         }
88         return mark;
89     }
90
91     @Override
92     public void insertMark(String subjectId, String taskid, String studentId, int
mark) throws RemoteException {
93         Marks aMark = new Marks();
94         if (!em.getTransaction().isActive()) {
95             em.getTransaction().begin();
96         }
97         aMark.setMark(mark);
98         aMark.setStudentid(studentId);
99         aMark.setTaskid(taskid);
100        aMark.setSubjectid(subjectId);
101        em.persist(aMark);
102        em.getTransaction().commit();
103        em.clear();
104
105
106
107    }
108
109    @Override
110    public int searchMark(String studentId, String subjectId, String taskId)
throws RemoteException {
111        int mark = 0;
112        Query query = em.createQuery("SELECT m FROM Marks m WHERE
m.subjectid = :subjectid AND m.taskid = :taskid AND m.studentid = :studentid");
113        query.setParameter("subjectid", subjectId);
114        query.setParameter("taskid", taskId);
115        query.setParameter("studentid", studentId);
116        List<Marks> marks = query.getResultList();
117        if (marks.isEmpty()) {
118            return 101;
119        }
120        for (Marks m : marks) {
121            mark = m.getMark();;
122        }
123        return mark;
124

```



```

125
126     }
127
128     @Override
129     public void deleteTask(String taskid) throws RemoteException {
130         if (!em.getTransaction().isActive()) {
131             em.getTransaction().begin();
132         }
133         Query query = em.createNamedQuery("Tasks.findByTaskid");
134         query.setParameter("taskid", taskid);
135         List<Tasks> list = query.getResultList();
136         if (!list.isEmpty()) {
137             for (Tasks t : list) {
138                 em.remove(t);
139             }
140             em.getTransaction().commit();
141             em.clear();
142         }
143         Query query1 = em.createNamedQuery("Marks.findByTaskid");
144         query1.setParameter("taskid", taskid);
145
146         List<Marks> marks = query1.getResultList();
147         if (!marks.isEmpty()) {
148             for (Marks m : marks) {
149                 em.remove(m);
150             }
151             em.getTransaction().commit();
152             em.clear();
153         }
154
155     }
156
157     @Override
158     public void deleteEnrolment(String studentId, String subjectId) throws
RemoteException {
159         if (!em.getTransaction().isActive()) {
160             em.getTransaction().begin();
161         }
162         Query query = em.createQuery("SELECT e FROM Enrolment e
WHERE e.enrolmentPK.studentid = :studentid AND e.enrolmentPK.subjectid
= :subjectid");
163         query.setParameter("subjectid", subjectId);
164         query.setParameter("studentid", studentId);
165         List<Enrolment> list = query.getResultList();

```

```

166
167     for (Enrolment e : list) {
168         em.remove(e);
169     }
170     em.getTransaction().commit();
171     em.clear();
172 }
173
174 @Override
175 public boolean enrolmented(String studentId, String subjectId) throws
RemoteException {
176     boolean flag = false;
177     Query q = em.createNamedQuery("Enrolment.findBySubjectid");
178     q.setParameter("subjectid", subjectId);
179     List<Enrolment> enrolment = (List<Enrolment>) q.getResultList();
180     int num = enrolment.size();
181     if (num != 0) {
182         for (Enrolment e : enrolment) {
183             if (e.getEnrolmentPK().getStudentid().equals(studentId)) {
184                 flag = true;
185             }
186         }
187     }
188
189     return flag;
190 }
191
192
193 @Override
194 public String[] enrolmentInfo(String subjectId) throws RemoteException {
195
196     Query q = em.createNamedQuery("Enrolment.findBySubjectid");
197     q.setParameter("subjectid", subjectId);
198     List<Enrolment> enrolment = (List<Enrolment>) q.getResultList();
199     int num = enrolment.size();
200     int i = 0;
201     if (num == 0) {
202         em.clear();
203         return null;
204     }
205     String[] studentId = new String[num];
206     for (Enrolment e : enrolment) {
207         studentId[i] = e.getEnrolmentPK().getStudentid().toString();
208         i++;

```

```

209         }
210         em.clear();
211         return studentId;
212     }
213
214     @Override
215     public Collection<Subject> getSubjects(String id) throws
RemoteException {
216         Staff staff = em.find(Staff.class, id);
217         Collection<Subject> results = staff.getSubjectCollection();
218         return results;
219     }
220
221     //get tasks for assessemnt table;
222     @Override
223     public Tasks[] getTasks(String id) throws RemoteException {
224         Query q = em.createNamedQuery("Tasks.findBySubjectid");
225         q.setParameter("subjectid", id);
226         List<Tasks> tasks = (List<Tasks>) q.getResultList();
227         Tasks[] t = new Tasks[tasks.size()];
228         tasks.toArray(t);
229         em.clear();
230         return t;
231     }
232
233     @Override
234     public void setContent(String content, String objectives, String subjectid)
throws RemoteException {
235         //set description of one subject.
236         Subject subject = em.find(Subject.class, subjectid);
237         if (!em.getTransaction().isActive()) {
238             em.getTransaction().begin();
239         }
240         subject.setContent(content);
241         subject.setObjectives(objectives);
242         em.getTransaction().commit();
243         em.clear();
244     }
245
246     @Override
247     public void updateMark(String taskid, String studentid, int mark) throws
RemoteException {
248
249         if (!em.getTransaction().isActive()) {

```

```

250         em.getTransaction().begin();
251     }
252     Query query = em.createQuery("SELECT m FROM Marks m WHERE
m.studentid = :studentid AND m.taskid = :taskid");
253     query.setParameter("studentid", studentid);
254     query.setParameter("taskid", taskid);
255     List<Marks> marks = query.getResultList();
256     for (Marks m : marks) {
257
258         m.setMark(mark);
259     }
260     em.getTransaction().commit();
261     em.clear();
262
263 }
264
265 @Override
266 public void deleteAlltasks(String subjectId) throws RemoteException {
267     if (!em.getTransaction().isActive()) {
268         em.getTransaction().begin();
269     }
270     Query q = em.createNamedQuery("Tasks.findBySubjectid");
271     q.setParameter("subjectid", subjectId);
272     List<Tasks> tasks = (List<Tasks>) q.getResultList();
273     Tasks[] t = new Tasks[tasks.size()];
274     tasks.toArray(t);
275     for (int j = 0; j < tasks.size(); j++) {
276         em.remove(t[j]);
277     }
278     em.getTransaction().commit();
279     em.clear();
280 }
281
282 @Override
283 public void addEnrolment(String updated, String subjectId) throws
RemoteException {
284     if (!em.getTransaction().isActive()) {
285         em.getTransaction().begin();
286     }
287     Enrolment e = new Enrolment();
288     EnrolmentPK ePK = new EnrolmentPK();
289     ePK.setStudentid(updated);
290     ePK.setSubjectid(subjectId);
291     e.setEnrolmentPK(ePK);

```

```

292         em.persist(e);
293         em.getTransaction().commit();
294         em.clear();
295     }
296
297     @Override
298     public void updateTasks(String[][] tasks, String subjectId) throws
RemoteException {
299
300         int num = tasks.length;
301         for (int i = 0; i < num; i++) {
302             if (!em.getTransaction().isActive()) {
303                 em.getTransaction().begin();
304             }
305             Tasks t = new Tasks();
306             TasksPK pk = new TasksPK();
307             pk.setSubjectid(subjectId);
308             pk.setTaskid(tasks[i][0]);
309             t.setTasksPK(pk);
310             int mark = Integer.parseInt(tasks[i][1]);
311             t.setMark(mark);
312             t.setDescription(tasks[i][2]);
313             em.persist(t);
314             em.getTransaction().commit();
315             em.clear();
316         }
317
318     }
319 }
320

```

Nonsuch client

NonsuchClient

code

just show the GUI and get connection with server.

F:\web\398\NonsuchClient\src\nonsuchclient\NonsuchClient.java

1 /*

```

2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package nonsuchclient;
6
7  import NonsuchInterface.NonsuchRemote;
8  import guistuff.NonsuchGUI;
9  import java.net.MalformedURLException;
10 import java.rmi.Naming;
11 import java.rmi.NotBoundException;
12 import java.rmi.RemoteException;
13 import java.util.logging.Level;
14 import java.util.logging.Logger;
15 import nonserver.Nonserver;
16
17 /**
18  *
19  * @author dx869
20  */
21 public class NonsuchClient {
22
23     //set up
24     private static final String hostName="localhost";
25     private static final String SoccerServiceport="123";
26     private static final String principal="Nonserver";
27     private static NonsuchRemote myServer;
28
29     private static void launchGUI() {
30         java.awt.EventQueue.invokeLater(new Runnable() {
31             @Override
32             public void run() {
33                 //show the GUI
34                 new NonsuchGUI().setVisible(true);
35             }
36         });
37     }
38     /**
39      * @param args the command line arguments
40      */
41     public static void main(String[] args) {
42
43
44         //connect to remot

```

```

45         String
serviceName="rmi://" + hostName + ":" + SoccerServiceport + "/" + principal;
46         System.out.println("Contacting service " + serviceName);
47
48         try{
49             myServer=(NonsuchRemote) Naming.lookup(serviceName);
50         }catch( MalformedURLException  mfurle){
51             System.out.println(mfurle);
52             System.exit(1);
53         } catch (NotBoundException ex) {
54
55             Logger.getLogger(NonsuchClient.class.getName()).log(Level.SEVERE, null, ex);
56             } catch (RemoteException ex) {
57
58             Logger.getLogger(NonsuchClient.class.getName()).log(Level.SEVERE, null, ex);
59             }
60         }
61         launchGUI();
62         // oops  return a brand new RMINonsuchServer
63         public static NonsuchRemote getServer() {
64             return myServer;
65         }
66
67 }
68
69
70

```

GUI

Just put the part of code here

1.get ready

```

37 public class NonsuchGUI extends javax.swing.JFrame {
38
39     private static NonsuchRemote myServer;
40     private static String staffid;
41     private static String subjectId;
42     private static Subject subject;
43     private DefaultTableModel dtm;
44     private DefaultTableModel mt;
45
46     /**
47      * Creates new form NonsuchGUI
48      */
49     public NonsuchGUI() {
50         initComponents();
51         //get myServer from nonsuch client.
52         myServer = (NonsuchRemote) NonsuchClient.getServer();
53         loginPane.setEnabledAt(2, false);
54         loginPane.setEnabledAt(3, false);
55         loginPane.setEnabledAt(4, false);
56         loginPane.setEnabledAt(5, false);
57         loginPane.setEnabledAt(6, false);
58         loginPane.setEnabledAt(7, false);
59     }

```

Login

Screen Shot

Subject description Assessment Enrolment Upload marks View marks

Login Messages Subject selection

Nonsuch University Subject Management System

Name

Password

Vesion 5

Subject description Assessment Enrolment Upload marks View marks

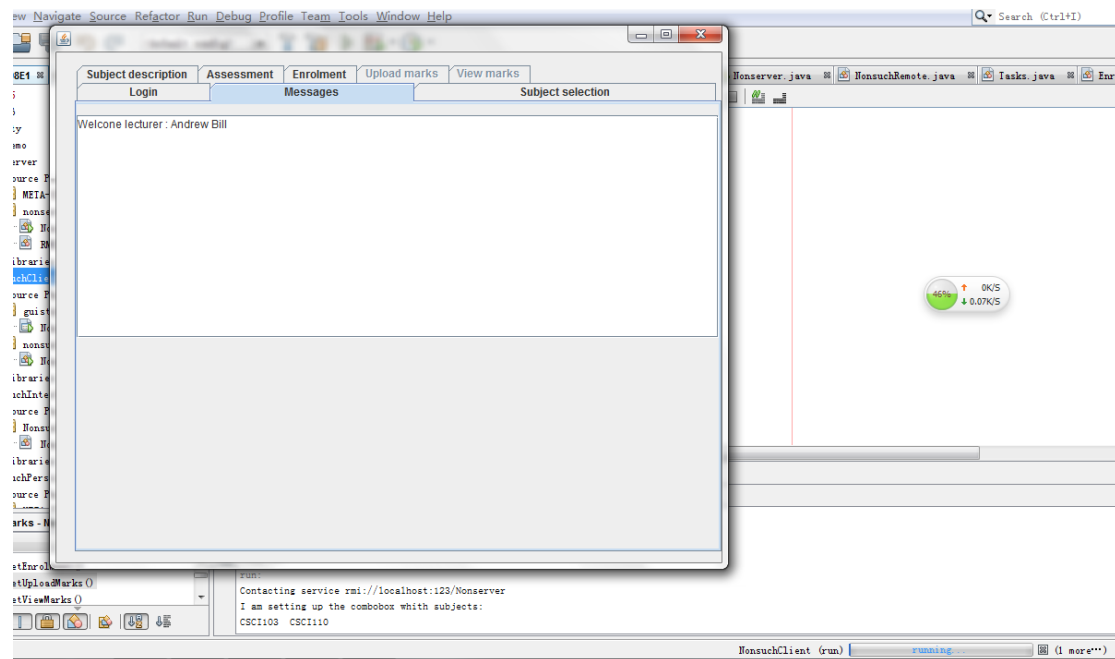
Login Messages Subject selection

Please enter name or password

Subject description Assessment Enrolment Upload marks View marks

Login Messages Subject selection

Sorry , can not find id



Code

```

647     private void loginActionPerformed(java.awt.event.ActionEvent evt) {
648
649         //check the log in vale if it is works then the rest panel will be
available;
650
651         String name = userName.getText().trim();
652         String pwd = password.getText().trim();
653
654         if (name.isEmpty() || pwd.isEmpty()) {
655             String emptymessage = "Please enter name or password";
656             message.setText(emptymessage);
657             loginPane.setSelectedComponent(Messages);
658         }
659         try {
660             Staff staff = myServer.staffInfo(name);
661             if (staff != null) {
662                 //Create MessageDigest object for MD5
663                 MessageDigest digest =
MessageDigest.getInstance("MD5");
664                 //Update input string in message digest
665                 digest.update(pwd.getBytes(), 0, pwd.length());
666                 //Converts message digest value in base 16 (hex)
667                 String md5Pwd = new BigInteger(1,
digest.digest()).toString(16);
668
669                 // after login
670                 //the rest pane will available

```

```

671         if (staff.getPassword().equals(md5Pwd)) {
672             loginPane.setEnabledAt(2, true);
673             loginPane.setEnabledAt(3, true);
674             loginPane.setEnabledAt(4, true);
675             loginPane.setEnabledAt(5, true);
676             // a message shows in message panel;
677             String login = "Welcone " + staff.getRole() + " : " +
staff.getFullname();
678             message.setText(login);
679             loginPane.setSelectedComponent(Messages);
680             // set up the select subject panel
681             fullname.setText(staff.getFullname());
682             role.setText(staff.getRole());
683             staffid = staff.getStaffid();
684
685             //find subject.
686             Collection<Subject> subjects =
staff.getSubjectCollection();
687             int a = subjects.size();
688             Subject[] sArray = new Subject[a];
689             subjects.toArray(sArray);
690             String[] array = new String[a];
691             System.out.println("I am setting up the combobox
with subjects: ");
692             for (int i = 0; i < a; i++) {
693                 array[i] = sArray[i].getSubjectid();
694                 System.out.print(array[i]+" ");
695
696             }
697             selectSubject.setModel(new
javax.swing.DefaultComboBoxModel(array));
698
699         }
700
701     } else {
702         String nameError = "Sorry , can not find id";
703         message.setText(nameError);
704         loginPane.setSelectedComponent(Messages);
705     }
706 } catch (RemoteException ex) {
707
Logger.getLogger(NonsuchGUI.class.getName()).log(Level.SEVERE, null, ex);
708 } catch (NoSuchAlgorithmException ex) {

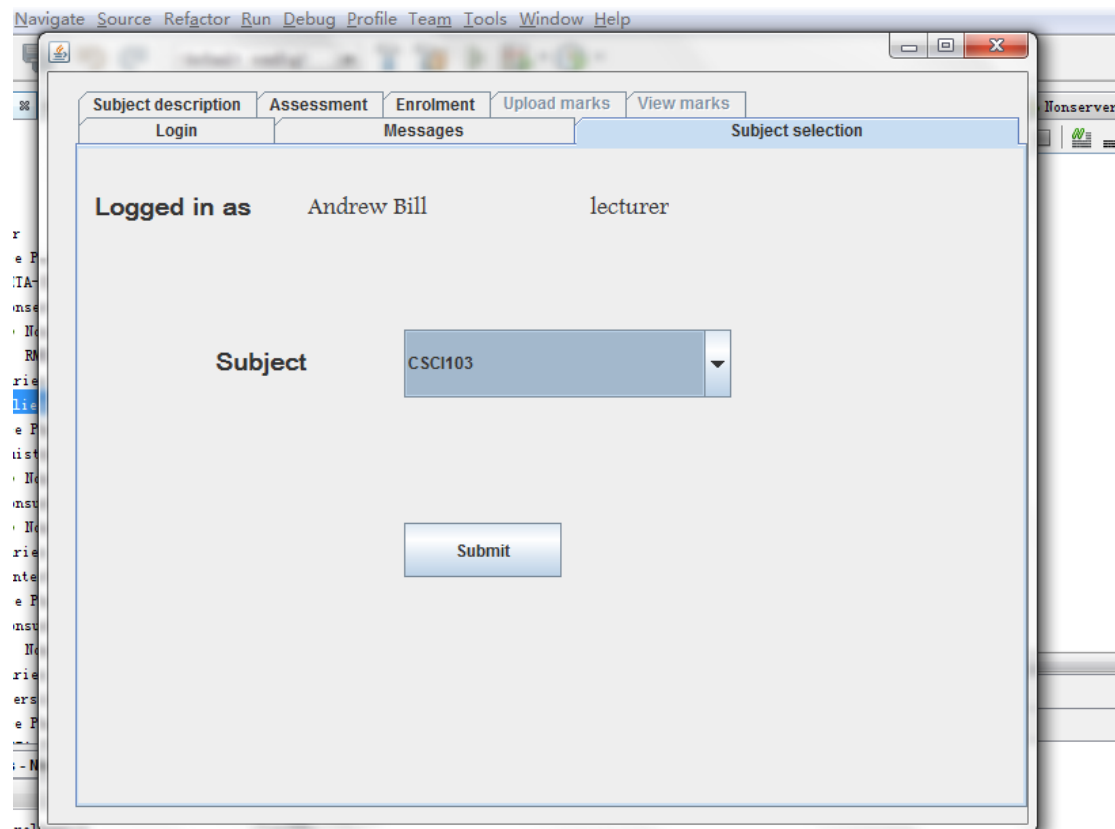
```

```

709
Logger.getLogger(NonsuchGUI.class.getName()).log(Level.SEVERE, null, ex);
710     }
711
712
713
714
715
716
717
718     }

```

Selection page



Code

```

private void selectedActionPerformed(java.awt.event.ActionEvent evt) {
737     //get the subject id then settle the rest
738     Object o = selectSubject.getSelectedItemAt();
739     subjectId = o.toString();
740     try {
741         subject = myServer.subjectInfo(subjectId);
742     } catch (RemoteException ex) {

```

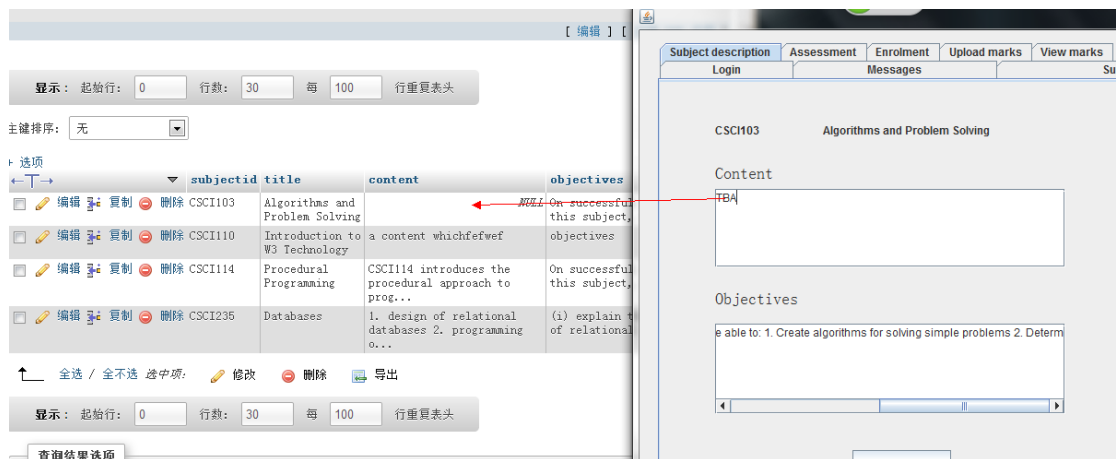
```

743
Logger.getLogger(NonsuchGUI.class.getName()).log(Level.SEVERE, null, ex);
744     }
745     String id = subjectId;
746     String title = subject.getTitle();
747     String content = subject.getContent();
748     String objectives = subject.getObjectives();
749
750     if (content == null) {
751         content = "TBA";
752     }
753     if (objectives == null) {
754         content = "TBA";
755     }
756     subjectidLabel.setText(id);
757     subjectnameLabel.setText(title);
758     ContentText.setText(content);
759     objectiveText.setText(objectives);
760     loginPane.setEnabledAt(6, false);
761     loginPane.setEnabledAt(7, false);
762     setAssessment();
763
764     try {
765         setEnrolment();
766     } catch (RemoteException ex) {
767
Logger.getLogger(NonsuchGUI.class.getName()).log(Level.SEVERE, null, ex);
768     }
769
770
771
772     }

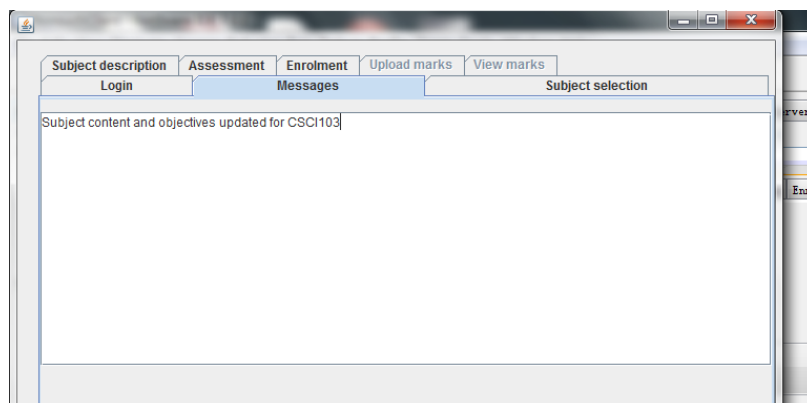
```

Subject description

Code



After updated.



Code

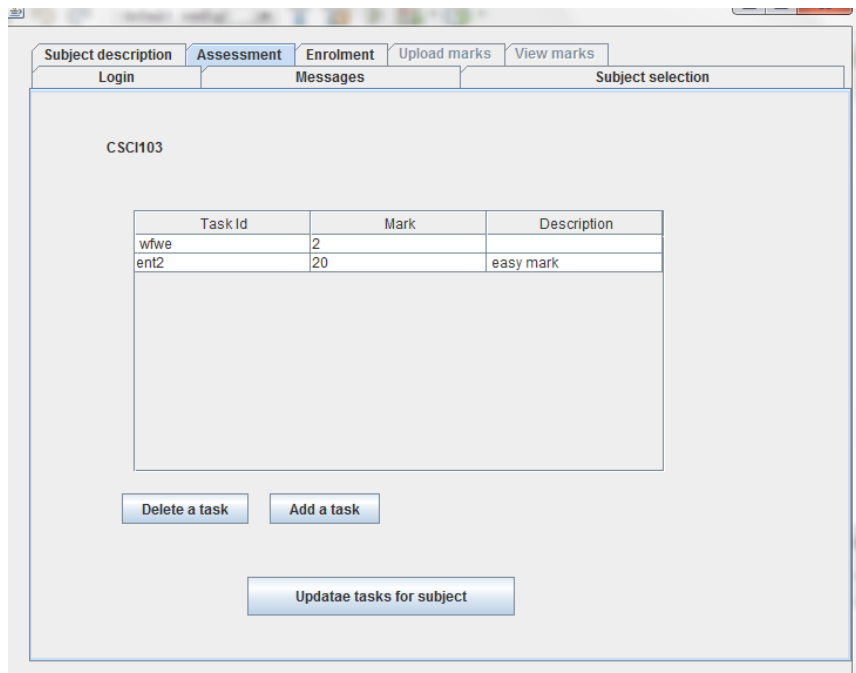
```

839     private void updateSubjectActionPerformed(java.awt.event.ActionEvent
840     evt) {
841         try {
842             // TODO add your handling code here:
843             //set value
844             myServer.setContent(ContentText.getText(),
845             objectiveText.getText(), subjectId);
846             String s = "Subject content and objectives updated for " +
847             subjectId;
848             message.setText(s);
849             loginPane.setSelectedComponent(Messages);
850             //objectiveText.setText(sArray[i].getObjectives());
851             } catch (RemoteException ex) {
852                 Logger.getLogger(NonsuchGUI.class.getName()).log(Level.SEVERE, null, ex);
853             }
854         }
855     }

```

Assessment

SET UP



Code

```
791 //italize the enrolment panel
807 // initialize the assessment panel
808
809 private void setAssessment() {
810     // set label to correct value
811     String title = subject.getTitle();
812     subjectidLabel2.setText(subjectId);
813     subjectnameLabel2.setText(title);
814
815     // get the tasks in subject
816     try {
817         Tasks[] tasks = myServer.getTasks(subjectId);
818
819         //display the tasks;
820         Vector<String> row;
821         dtm = new DefaultTableModel();
822         dtm.addColumn("Task Id");
823         dtm.addColumn("Mark");
824         dtm.addColumn("Description");
825         for (int j = 0; j < tasks.length; j++) {
826             row = new Vector();
```

```

827         row.add(tasks[j].getTasksPK().getTaskid());
828         row.add(tasks[j].getMark().toString());
829         row.add(tasks[j].getDescription());
830         dtm.addRow(row);
831     }
832     taskTable.setModel(dtm);
833 } catch (RemoteException ex) {
834

```

```

Logger.getLogger(NonsuchGUI.class.getName()).log(Level.SEVERE, null, ex);

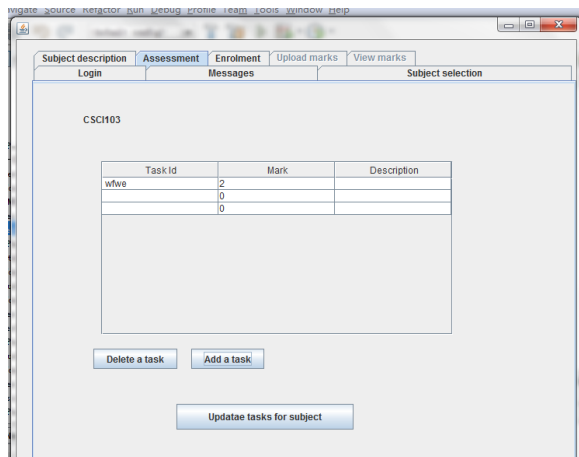
```

```

835     }

```

Add two new row



Code

```

864 private void addTaskActionPerformed(java.awt.event.ActionEvent evt) {
865     Vector<String> row;
866     row = new Vector();
867     row.add(" ");
868     row.add("0");
869     row.add(" ");
870     dtm.addRow(row);
871     // TODO add your handling code here:
872 }
873

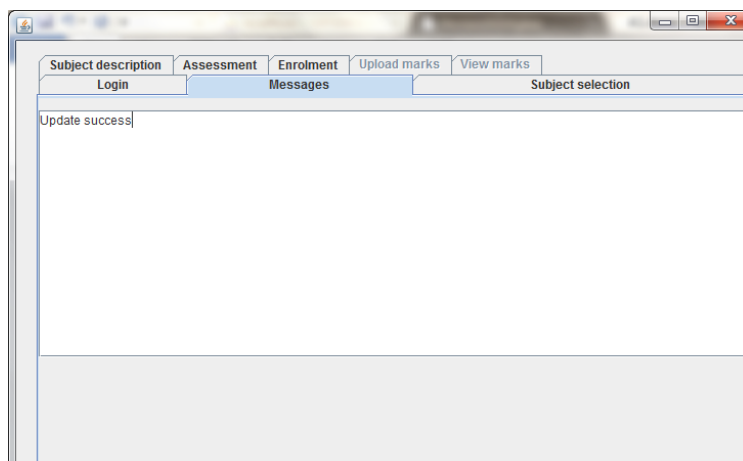
```

Edit

CSCI103

Task Id	Mark	Description
wfwe	2	
a4	20	ok

Update



Code

```

864 private void addAtaskActionPerformed(java.awt.event.ActionEvent evt) {
865     Vector<String> row;
866     row = new Vector();
867     row.add(" ");
868     row.add("0");
869     row.add(" ");
870     dtm.addRow(row);
871     // TODO add your handling code here:
872 }
873
874 private void updateTasksActionPerformed(java.awt.event.ActionEvent
875 evt) {
876     //checking the data
877     //add tasks into database

```

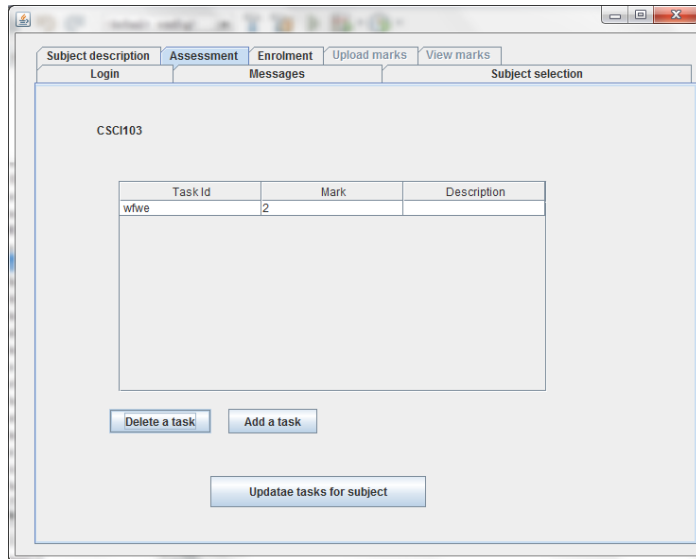
```

877 //show message in the message panel.
878 int rowCount = dtm.getRowCount();
879 boolean flag = true;
880 String tasks[][] = new String[rowCount][3];
881
882 //check if the task id or mark is legal;
883 for (int i = 0; i < rowCount; i++) {
884     tasks[i][0] = dtm.getValueAt(i, 0).toString();
885     for (int m = 0; m < i; m++) {
886         if (tasks[i][0].equals(tasks[m][0])) {
887             flag = false;
888             break;
889         }
890     }
891
892     // check if marks is integer use regular
893     tasks[i][1] = dtm.getValueAt(i, 1).toString();
894     Pattern pattern = Pattern.compile("[0-9]+");
895     if (!pattern.matcher(tasks[i][1]).matches()) {
896         flag = false;
897         break;
898     }
899     tasks[i][2] = dtm.getValueAt(i, 2).toString();
900 }
901 if (flag != true) {
902     message.setText("Update failed");
903     loginPane.setSelectedComponent(Messages);
904 } else {
905     try {
906         myServer.deleteAlltasks(subjectId);
907     } catch (RemoteException ex) {
908
909         Logger.getLogger(NonsuchGUI.class.getName()).log(Level.SEVERE, null, ex);
910     }
911     try {
912         myServer.updateTasks(tasks, subjectId);
913     } catch (RemoteException ex) {
914
915         Logger.getLogger(NonsuchGUI.class.getName()).log(Level.SEVERE, null, ex);
916     }
917     message.setText("Update success");
918     loginPane.setSelectedComponent(Messages);

```

919
920
921 }

Delete

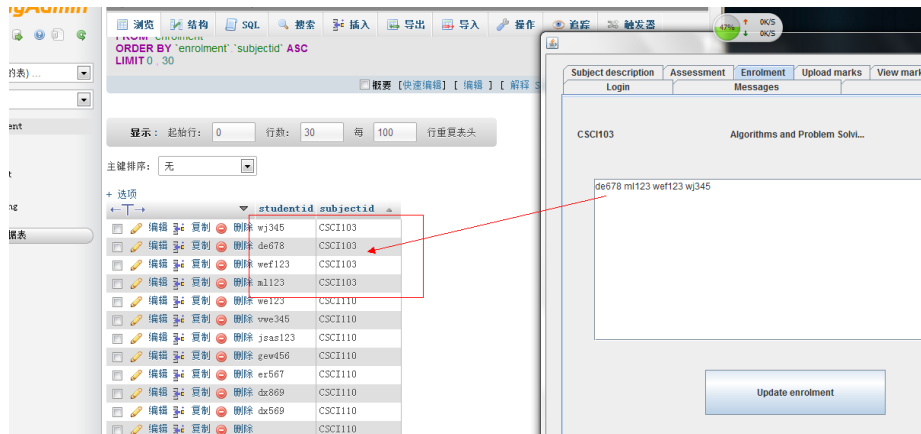


Code

```
854 private void deletAtaskActionPerformed(java.awt.event.ActionEvent evt)
855 {
856     int selIndex = taskTable.getSelectedRow();
857     dtm.removeRow(selIndex);
858
859
860
861     // TODO add your handling code here:
862 }
```

Enrolment

Set up



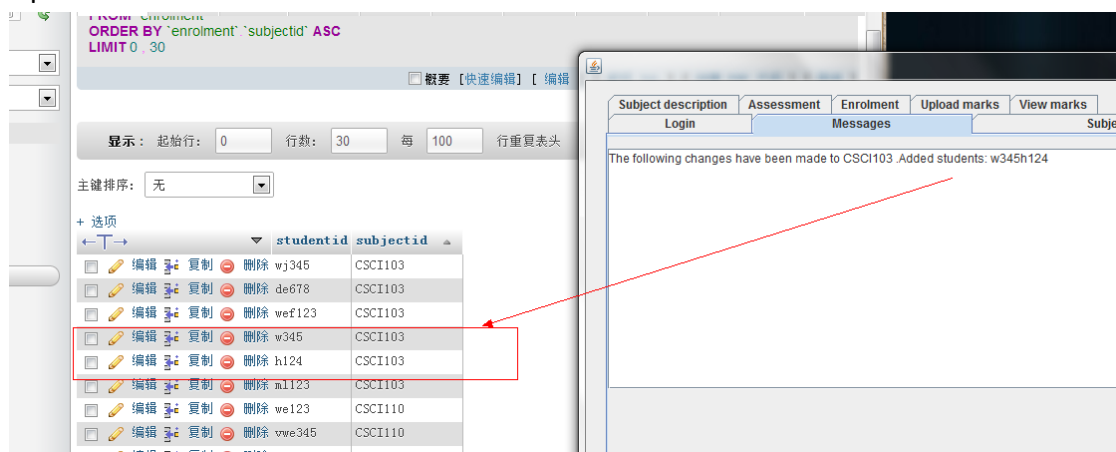
Code

```

792     private void setEnrolment() throws RemoteException {
793         String title = subject.getTitle();
794         subjectidLabel4.setText(subjectId);
795         subjectnameLabel4.setText(title);
796         String[] studentId = myServer.enrolmentInfo(subjectId);
797         if (studentId == null) {
798             enrolmentText.setText("");
799         } else {
800             String enrolmentString = "";
801             for (int i = 0; i < studentId.length; i++) {
802                 enrolmentString = enrolmentString + studentId[i] + " ";
803             }
804             enrolmentText.setText(enrolmentString);
805         }
806     }

```

Update



Code

```

923     private void
updateEnrolmentActionPerformed(java.awt.event.ActionEvent evt) {
924         try {
925             String s = enrolmentText.getText();

```

```

926 String[] old = myServer.enrolmentInfo(subjectId);
927 String[] updated = s.split(" ");
928 String add = "";
929 String delete = "";
930
931
932 // find the which one has been deleted from old studentid array;
933 //call function to delete this studentid from database
934 //make string for message panel to show
935 if (old == null) {
936     add = s;
937     for (int i = 0; i < updated.length; i++) {
938         myServer.addEnrolment(updated[i], subjectId);
939     }
940 } else {
941     for (int i = 0; i < old.length; i++) {
942         boolean flag = false;
943         for (int j = 0; j < updated.length; j++) {
944             if (old[i].equals(updated[j])) {
945                 flag = true;
946             }
947         }
948         if (flag == false) {
949             delete = delete + old[i];
950             myServer.deleteEnrolment(old[i], subjectId);
951         }
952     }
953     // find the which one has been added from updated
studentid array;
954     //call function to add this studentid from database
955     //make string for message panel to show
956     for (int i = 0; i < updated.length; i++) {
957         boolean flag = false;
958         for (int j = 0; j < old.length; j++) {
959             if (updated[i].equals(old[j])) {
960                 flag = true;
961             }
962         }
963         if (flag == false) {
964             add = add + updated[i];
965             myServer.addEnrolment(updated[i], subjectId);
966         }
967     }
968 }

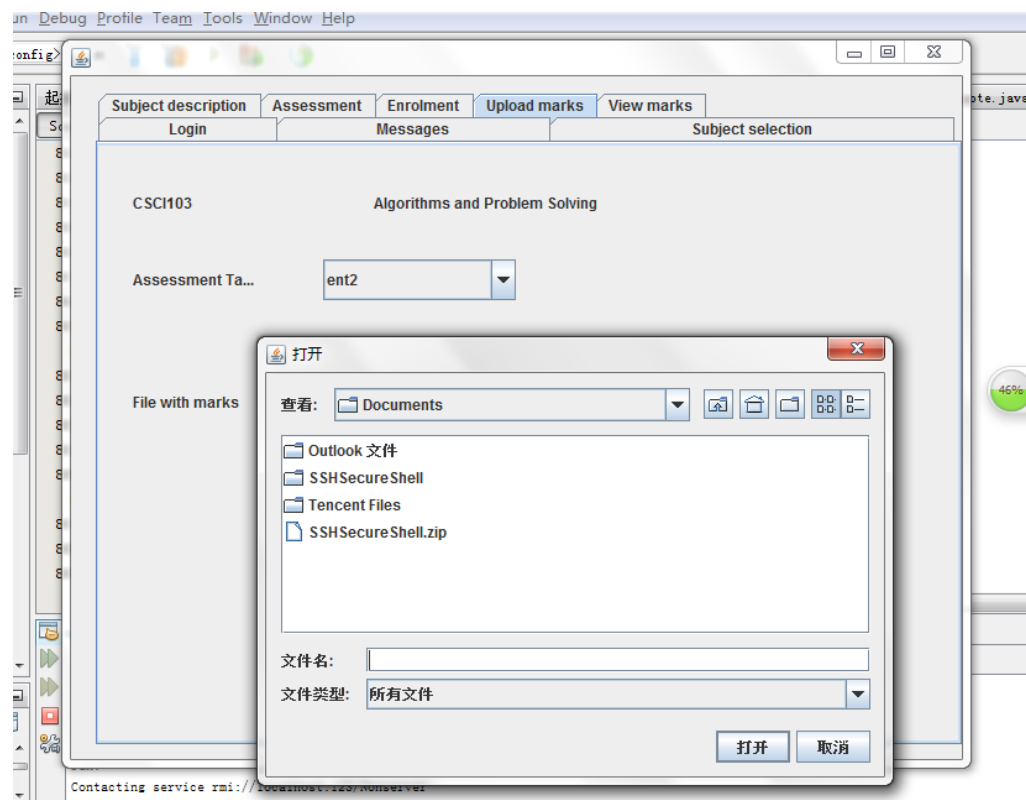
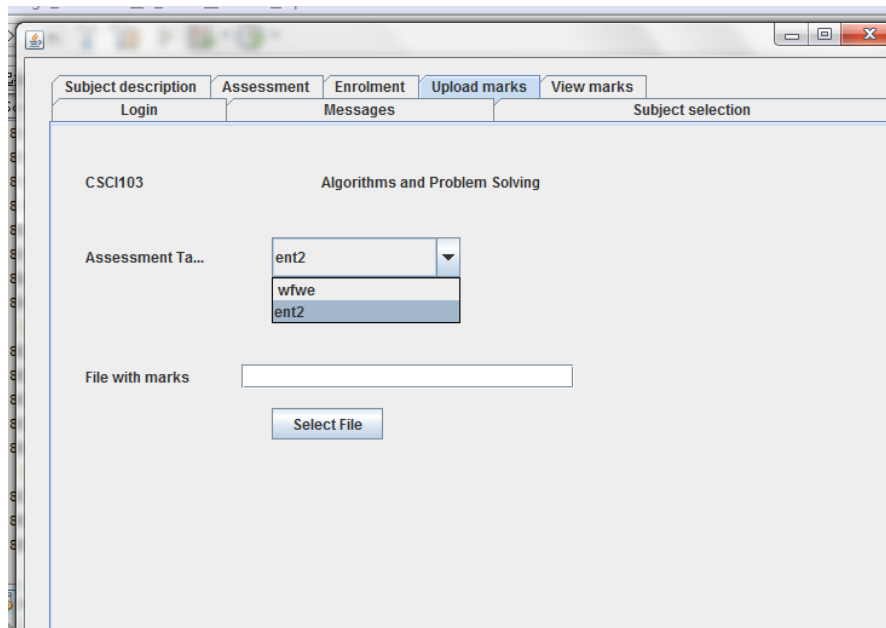
```

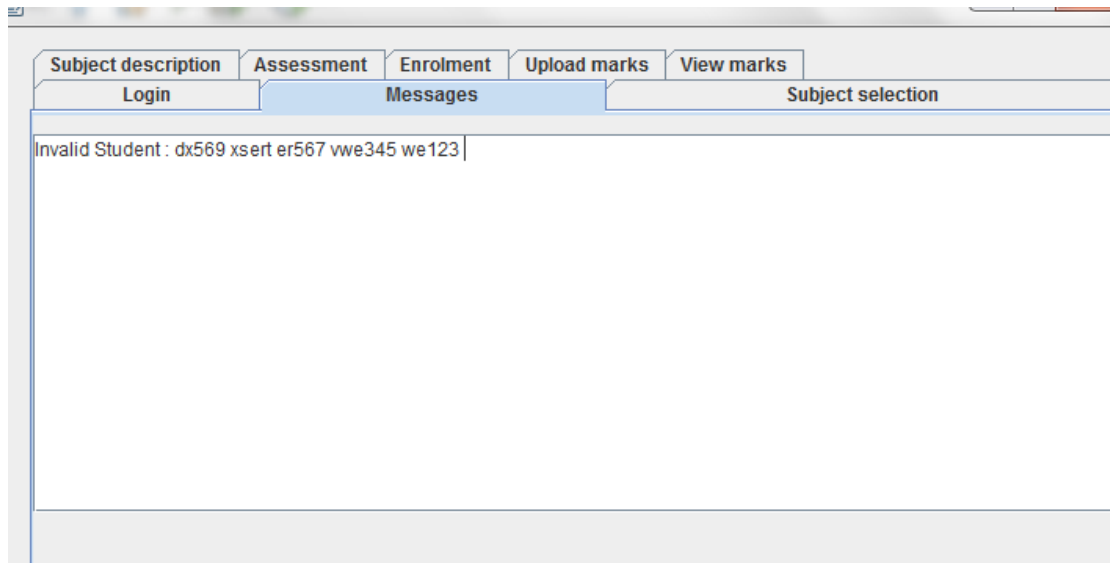
```

969         }
970
971         //show the message
972         String aMessage = "";
973         if (add == null && delete == null) {
974             aMessage = "There is no change for your update";
975         } else {
976             aMessage = "The following changes have been made to " +
subjectId + " .";
977             if (!add.equals("")) {
978                 aMessage = aMessage + "Added students: " + add;
979             }
980             if (!delete.equals("")) {
981                 aMessage = aMessage + "Removed students: " +
delete;
982             }
983         }
984         message.setText(aMessage);
985         loginPane.setSelectedComponent(Messages);
986         setViewMarks();
987
988         // TODO add your handling code here:
989     } catch (RemoteException ex) {
990
991         Logger.getLogger(NonsuchGUI.class.getName()).log(Level.SEVERE, null, ex);
992     }
993
994     // the rest two panel is available now
995     loginPane.setEnabledAt(6, true);
996     loginPane.setEnabledAt(7, true);
997     try {
998         setUploadMarks();
999     } catch (RemoteException ex) {
1000
1001         Logger.getLogger(NonsuchGUI.class.getName()).log(Level.SEVERE, null, ex);
1002     }
1003 }

```

Update Marks





Code

Set up

```

775     private void setUploadMarks() throws RemoteException {
776         String title = subject.getTitle();
777         subjectidLabel3.setText(subjectId);
778         subjectnameLabel3.setText(title);
779         Tasks[] tasks = myServer.getTasks(subjectId);
780         if (tasks == null) {
781             } else {
782                 int a = tasks.length;
783                 String[] array = new String[a];
784                 for (int i = 0; i < a; i++) {
785                     array[i] = tasks[i].getTasksPK().getTaskid();
786                 }
787                 selectMarkBox.setModel(new
java.swing.DefaultComboBoxModel(array));
788             }
789         }

```

Read file

```

1051 //get file and read file deal with file...
1052     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
1053     {
1054         InputStream is = null;
1055         String invalidStudent = "Invalid Student : ";
1056         Object o = selectMarkBox.getSelectedItem();
1057         String taskid = o.toString();
1058         boolean exit;
1059         int mark;

```



```

1060         //Create a file chooser
1061         final JFileChooser fc = new JFileChooser();
1062         int returnVal = fc.showOpenDialog(loginPane);
1063         File file = fc.getSelectedFile();
1064
1065         //change text field 's value
1066         fileName.setText(file.getPath());
1067         try {
1068             //open and read the file
1069             is = new FileInputStream(file);
1070         } catch (FileNotFoundException ex) {
1071
1072             Logger.getLogger(NonsuchGUI.class.getName()).log(Level.SEVERE, null, ex);
1073
1074             InputStreamReader isr = new InputStreamReader(is);
1075             BufferedReader reader = new BufferedReader(isr);
1076             try {
1077                 String text = reader.readLine();
1078                 while (text != null) {
1079                     String[] pairs = text.split(" ");
1080
1081                     //check if the student is exit and enrolled in this subject
1082                     exit = myServer.enrolled(pairs[0], subjectId);
1083                     if (exit) {
1084                         mark = Integer.parseInt(pairs[1]);
1085
1086                         //check if the mark is less than the total mark in the
1087                         task.
1088                         if (mark <= myServer.getTask(taskid, subjectId)) {
1089                             myServer.insertMark(subjectId, taskid, pairs[0],
1090                             mark);
1091                         } else {
1092                             invalidStudent = invalidStudent + pairs[0] + " ";
1093                         }
1094                     } else {
1095                         invalidStudent = invalidStudent + pairs[0] + " ";
1096                     }
1097                     text = reader.readLine();
1098                 }
1099                 message.setText(invalidStudent);
1100                 loginPane.setSelectedComponent(Messages);
1101             } catch (IOException ex) {

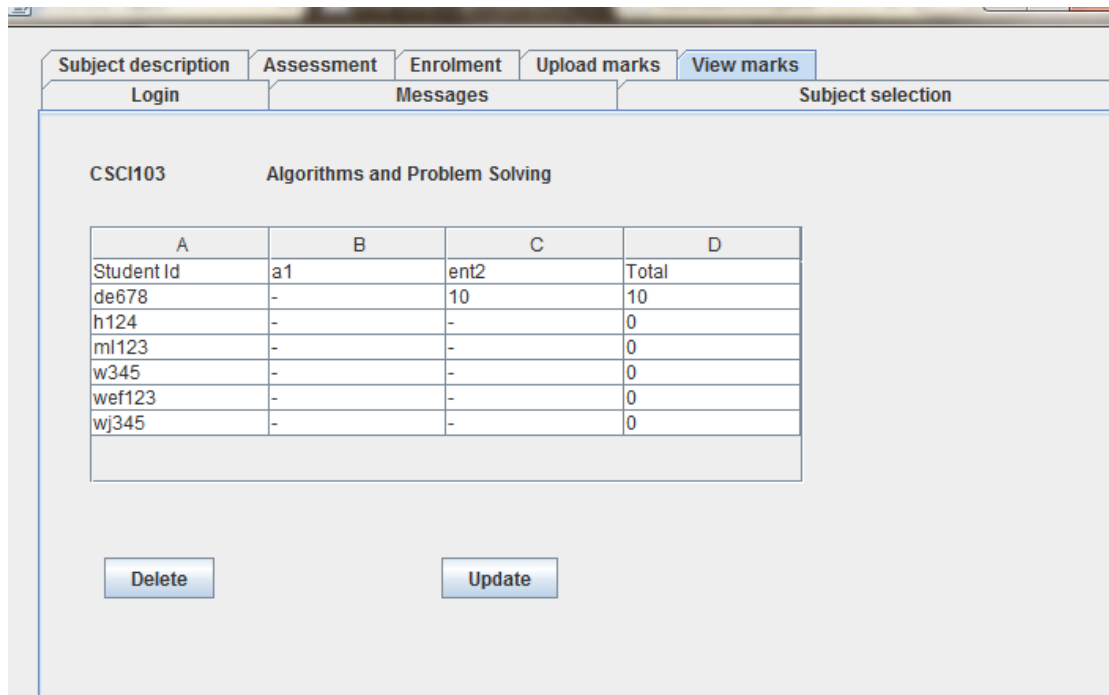
```

```

1101
1102     Logger.getLogger(NonsuchGUI.class.getName()).log(Level.SEVERE, null, ex);
1103     }
1104
1105     //
1106
1107
1108
1109     }

```

View Marks



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

Buttons: Delete, Update

Set up

```

public void setViewMarks() throws RemoteException {
1003     String title = subject.getTitle();
1004     subjectidLabel5.setText(subjectId);
1005     subjectnameLabel5.setText(title);
1006     String[] studentId = myServer.enrolmentInfo(subjectId);
1007     String nothing = "-";
1008     //display the tasks;
1009
1010
1011     try {
1012

```

```

1013
1014
1015         Tasks[] tasks = myServer.getTasks(subjectId);
1016         mt = new DefaultTableModel();
1017         mt.setColumnCount((tasks.length + 1));
1018         mt.setRowCount((studentId.length + 1));
1019         markTable.setModel(mt);
1020         mt.setValueAt("Student Id", 0, 0);
1021         // first lay of for-loop is find the taskid and put to title.
1022         // second lay of for-loop is when tasks id id known, get the
student mars set as a row
1023
1024         //set the sutdent id in first column.
1025         int j = 1;
1026         for (int i = 0; i < studentId.length; i++) {
1027             mt.setValueAt(studentId[i], (i + 1), 0);
1028         }
1029         for (Tasks t : tasks) {
1030             mt.setValueAt(t.getTasksPK().getTaskid(), 0, j);
1031
1032             for (int i = 0; i < studentId.length; i++) {
1033                 int mark = myServer.searchMark(studentId[i],
subjectId, t.getTasksPK().getTaskid());
1034                 if (mark == 101) {
1035                     mt.setValueAt(nothing, (i + 1), j);
1036                 } else {
1037                     mt.setValueAt(mark, (i + 1), j);
1038                 }
1039             }
1040             j++;
1041         }
1042
        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);
        }

```

```

1043
1044         markTable.setModel(mt);
1045     } catch (RemoteException ex) {
1046
1047         Logger.getLogger(NonsuchGUI.class.getName()).log(Level.SEVERE, null, ex);
1048     }
1049
1050 }

```

Delete

```

1111     private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
1112     {
1113         int columncount = markTable.getSelectedColumn();
1114         String taskid = markTable.getValueAt(0, columncount).toString();
1115         try {
1116             myServer.deleteTask(taskid);
1117         } catch (RemoteException ex) {
1118
1119             Logger.getLogger(NonsuchGUI.class.getName()).log(Level.SEVERE, null, ex);
1120         }
1121         TableColumnModel columnModel = markTable.getColumnModel();
1122         TableColumn tableColumn =
1123             columnModel.getColumn(columncount);
1124         columnModel.removeColumn(tableColumn);
1125     }

```

Update

```

1126     private void jButton3ActionPerformed(java.awt.event.ActionEvent evt)
1127     {
1128         int i=0;
1129
1130         int row[]=new int[markTable.getRowCount()];
1131         int col[]=new int[markTable.getColumnCount()];
1132
1133         //when it is editing then get the rownumber and column number , stop
1134         editing.
1135         if(markTable.isEditing()){
1136             row[i]=markTable.getEditingRow();
1137             col[i]=markTable.getEditingColumn();

```

```

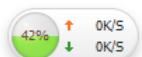
1136         markTable.getCellEditor().stopCellEditing();
1137
1138
1139         //get task id & student id & mark;
1140         String taskid=markTable.getValueAt(0, col[i]).toString();
1141         String studentid=markTable.getValueAt(row[i], 0).toString();
1142         String markS=markTable.getValueAt(row[i], col[i]).toString();
1143         int mark=Integer.parseInt(markS);
1144
1145
1146         //call remote method to update
1147         try {
1148             myServer.updateMark(taskid,studentid,mark);
1149         } catch (RemoteException ex) {
1150
1151             Logger.getLogger(NonsuchGUI.class.getName()).log(Level.SEVERE, null, ex);
1152         }
1153
1154
1155         // TODO add your handling code here:
1156     }

```

Building & Running


```
Output - Nonserver (clean, jar) 88
deps-clean:
Updating property file: F:\web\398\Nonserver\build\build-clean.properties
Deleting directory F:\web\398\Nonserver\build
clean:
init:
deps-jar:
Created dir: F:\web\398\Nonserver\build
Updating property file: F:\web\398\Nonserver\build\build-jar.properties
Created dir: F:\web\398\Nonserver\build\classes
Created dir: F:\web\398\Nonserver\build\classes\META-INF
Copying 1 file to F:\web\398\Nonserver\build\classes\META-INF
Created dir: F:\web\398\Nonserver\build\empty
Created dir: F:\web\398\Nonserver\build\generated-sources\ap-source-output
Compiling 2 source files to F:\web\398\Nonserver\build\classes
warning: Supported source version 'RELEASE_6' from annotation processor 'org.eclipse.persistence.internal.jpa.modelgen.Ca
warning: The following options were not recognized by any processor: '[eclipselink.canonicalmodel.use_static_factory]'
Note: F:\web\398\Nonserver\src\nonserver\RMInonsuchServer.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
1 warning
compile:
Created dir: F:\web\398\Nonserver\dist
Copying 1 file to F:\web\398\Nonserver\build
Copy libraries to F:\web\398\Nonserver\dist\lib.
Building jar: F:\web\398\Nonserver\dist\Nonserver.jar
To run this application from the command line without Ant, try:
java -jar "F:\web\398\Nonserver\dist\Nonserver.jar"
jar:
BUILD SUCCESSFUL (total time: 1 second)
```


```
Output - NonsuchPersistence (clean, jar) 88
Updating property file: F:\web\398\NonsuchPersistence\build\build-clean.properties
Deleting directory F:\web\398\NonsuchPersistence\build
clean:
init:
deps-jar:
Created dir: F:\web\398\NonsuchPersistence\build
Updating property file: F:\web\398\NonsuchPersistence\build\build-jar.properties
Created dir: F:\web\398\NonsuchPersistence\build\classes
Created dir: F:\web\398\NonsuchPersistence\build\classes\META-INF
Copying 1 file to F:\web\398\NonsuchPersistence\build\classes\META-INF
Created dir: F:\web\398\NonsuchPersistence\build\empty
Created dir: F:\web\398\NonsuchPersistence\build\generated-sources\ap-source-output
Compiling 7 source files to F:\web\398\NonsuchPersistence\build\classes
warning: Supported source version 'RELEASE_6' from annotation processor 'org.eclipse.persistence.internal.jpa.modelgen.CanonicalModelProcessor' less than -s
Note: Creating non-static metadata factory ...
Note: Found Option : eclipselink.canonicalmodel.use_static_factory, with value: false
Note: Optional file was not found: META-INF/orm.xml continuing with generation.
Note: Optional file was not found: META-INF/eclipselink-orm.xml continuing with generation.
Note: Found Option : eclipselink.canonicalmodel.use_static_factory, with value: false
Note: Optional file was not found: META-INF/orm.xml continuing with generation.
Note: Optional file was not found: META-INF/eclipselink-orm.xml continuing with generation.
warning: The following options were not recognized by any processor: '[eclipselink.canonicalmodel.use_static_factory]'
1 warning
compile:
Created dir: F:\web\398\NonsuchPersistence\dist
Building jar: F:\web\398\NonsuchPersistence\dist\NonsuchPersistence.jar
jar:
BUILD SUCCESSFUL (total time: 2 seconds)
```







```
Output - NonsuchInterface (clean, jar) %
ant -f F:\web\398\NonsuchInterface clean jar
init:
deps-clean:
Updating property file: F:\web\398\NonsuchInterface\build\build-clean.properties
Deleting directory F:\web\398\NonsuchInterface\build
clean:
init:
deps-jar:
Created dir: F:\web\398\NonsuchInterface\build
Updating property file: F:\web\398\NonsuchInterface\build\build-jar.properties
Created dir: F:\web\398\NonsuchInterface\build\classes
Created dir: F:\web\398\NonsuchInterface\build\empty
Created dir: F:\web\398\NonsuchInterface\build\generated-sources\ap-source-output
Compiling 1 source file to F:\web\398\NonsuchInterface\build\classes
compile:
Created dir: F:\web\398\NonsuchInterface\dist
Building jar: F:\web\398\NonsuchInterface\dist\NonsuchInterface.jar
jar:
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
Output - NonsuchClient (clean, jar) %
ant -f F:\web\398\NonsuchClient clean jar
init:
deps-clean:
Updating property file: F:\web\398\NonsuchClient\build\build-clean.properties
Deleting directory F:\web\398\NonsuchClient\build
clean:
init:
deps-jar:
Created dir: F:\web\398\NonsuchClient\build
Updating property file: F:\web\398\NonsuchClient\build\build-jar.properties
Created dir: F:\web\398\NonsuchClient\build\classes
Created dir: F:\web\398\NonsuchClient\build\empty
Created dir: F:\web\398\NonsuchClient\build\generated-sources\ap-source-output
Compiling 2 source files to F:\web\398\NonsuchClient\build\classes
Note: F:\web\398\NonsuchClient\src\guistuff\NonsuchGUI.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Note: F:\web\398\NonsuchClient\src\guistuff\NonsuchGUI.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
compile:
Created dir: F:\web\398\NonsuchClient\dist
Copying 1 file to F:\web\398\NonsuchClient\build
Copy libraries to F:\web\398\NonsuchClient\dist\lib.
Building jar: F:\web\398\NonsuchClient\dist\NonsuchClient.jar
To run this application from the command line without Ant, try:
java -jar "F:\web\398\NonsuchClient\dist\NonsuchClient.jar"
jar:
BUILD SUCCESSFUL (total time: 0 seconds)
```



Output - Nonserver (run) 

```
run:
[EL Info]: 2013-08-27 00:24:47.906--ServerSession(1842953823)--EclipseLink, version: Ecclipse Persistence Se
[EL Info]: connection: 2013-08-27 00:24:49.061--ServerSession(1842953823)--file:/F:/web/398/Nonserver/build
About to bindNonserver
yes it is working
```