DEPARTMENT OF
COMPUTER SCIENCE AND SOFTWARE
ENGINEERING

Web Service Implementation of the Distributed Course Registration System (DCRS)

By

Name : Deep Patel

1. Methods:

1) addCourse (courseID, semester):

This method is invoke by the professor to add new course to the system.

2) removeCourse (courseID, semester):

This method is invoke by the professor to remove course from the system.

3) listCourseAvailability (semester):

This method is invoked by the professor to get seats remain in all courses in respective semester.

4) enrolCourse (studentID, courseID, semester):

This method is invoked by the professor as well as student to enrol the student into respective course.

5) dropCourse (studentID, courseID):

This method is invoked by the professor as well as student to drop the student from respective course.

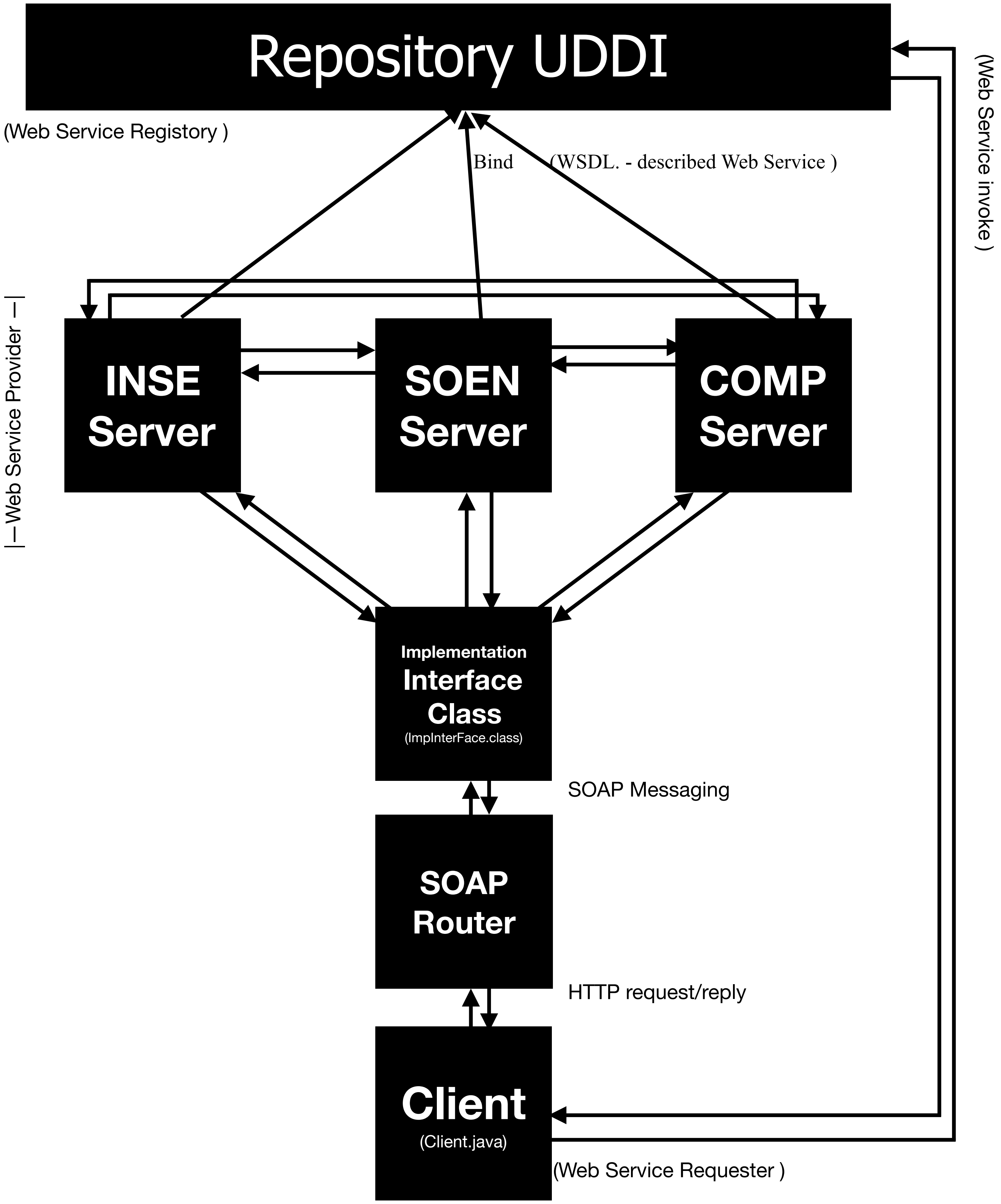
6) getClassSchedule (studentID):

This method is invoked by the professor as well as student to get the student's courses in all semester.

7) swapCourse (studentID, newCourseID, oldCourseID)

When an advisor/student invokes this method then student will exchange old course with new course in respective semester.

2. Architecture of System



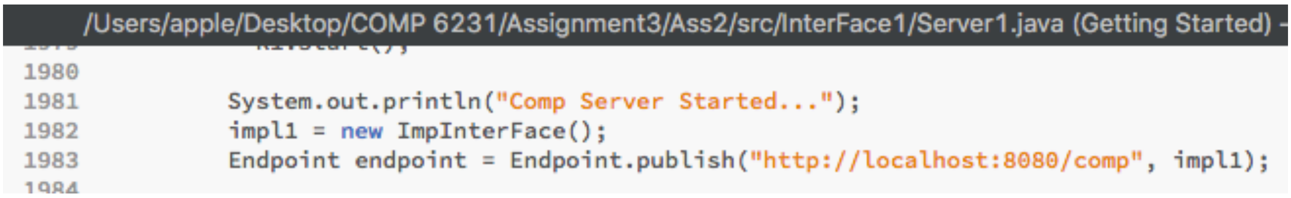
WSDL : It is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. [1]

UDDI : Universal Description, Discovery, and Integration is an XML-based standard for describing, publishing, & finding web services. [2]

SOAP : It is a messaging protocol specification for exchanging structured information in the implementation of web services. [3]

Workflow :

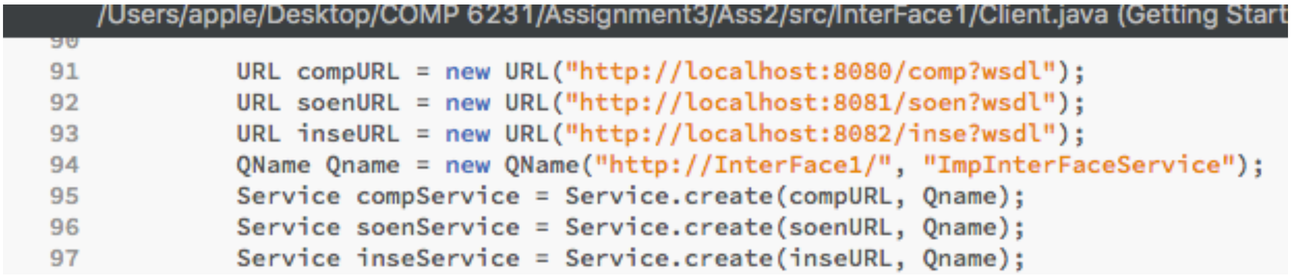
- First of all Server publish their services to service registry or repository so that service requestor can find appropriate services.



/Users/apple/Desktop/COMP 6231/Assignment3/Ass2/src/InterFace1/Server1.java (Getting Started) -

```
        System.out.println("Comp Server Started...");
        impl1 = new ImpInterFace();
        Endpoint endpoint = Endpoint.publish("http://localhost:8080/comp", impl1);
```

- Service registory contains description of services using UDDI language which is defined in XML.
- Then, client discovers the service in service registry using URL of service in wsdl and obtain service and binding information. Then, client binds to the SOAP server.



/Users/apple/Desktop/COMP 6231/Assignment3/Ass2/src/InterFace1/Client.java (Getting Start

```
        URL compURL = new URL("http://localhost:8080/comp?wsdl");
        URL soenURL = new URL("http://localhost:8081/soen?wsdl");
        URL inseURL = new URL("http://localhost:8082/inse?wsdl");
        QName Qname = new QName("http://InterFace1/", "ImpInterFaceService");
        Service compService = Service.create(compURL, Qname);
        Service soenService = Service.create(soenURL, Qname);
        Service inseService = Service.create(inseURL, Qname);
```

- In SOAP request Marshaling is needed in which needed services and args are send to the server.
- SOAP router routes the request to the appropriate server , Server compute result and send it back to the client.

3. Data Structure :

HashMap:

1.

```
HashMap<String, Integer> compfallsubject = new HashMap<String, Integer>();
```

This Hashmap is used to store remain course capacity.

ex. compsummersubject , soenfallsubject , compwintersubject etc.

2.

```
HashMap<String, String> compfallstudent = new HashMap<String, String>();
```

This Hashmap is used to store students information.

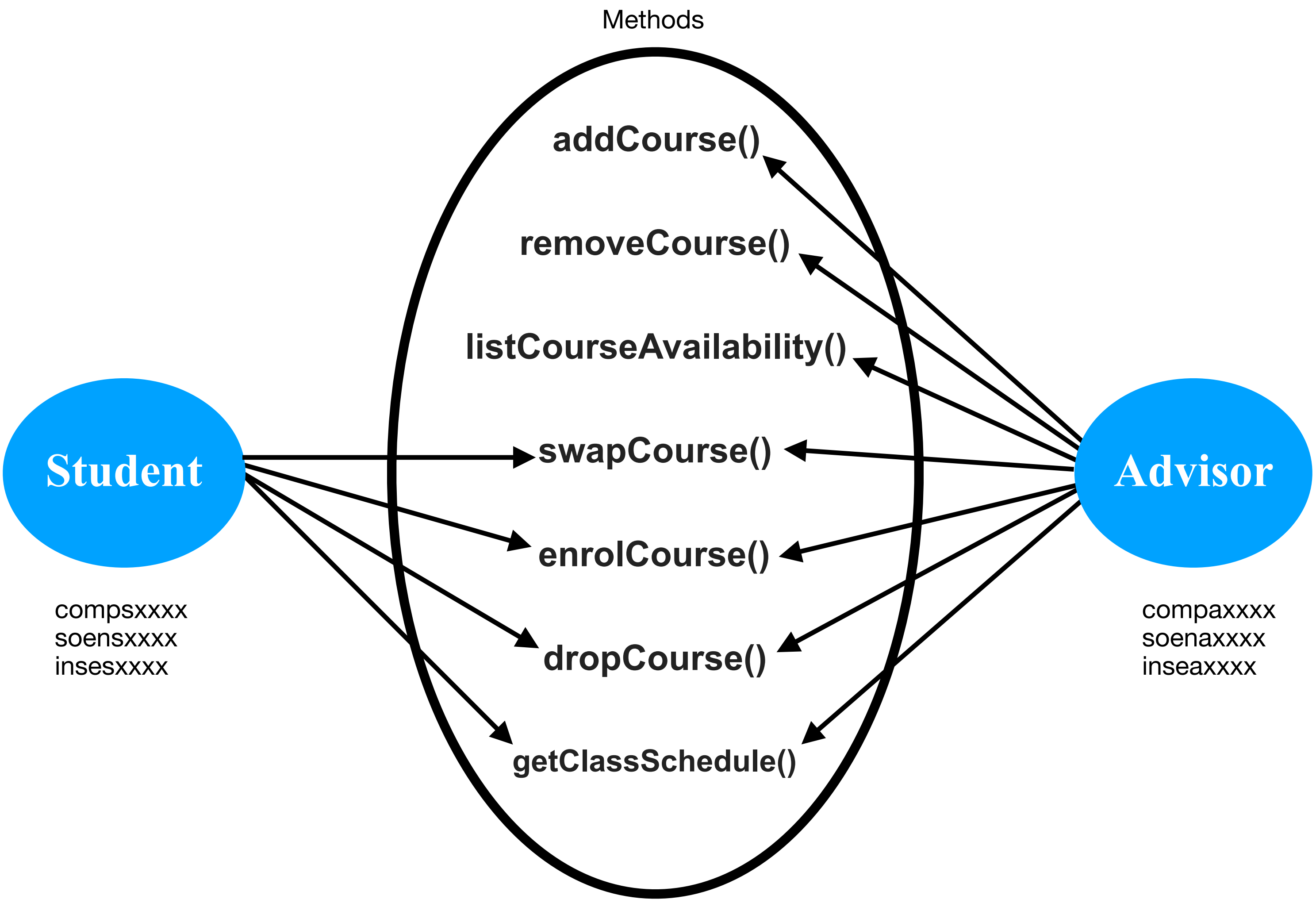
Ex. soenfallstudent , insefallstudent etc.

3.

```
HashMap<String, ArrayList<String>> compadvisor = new HashMap<String,
        ArrayList<String>>();

ArrayList<String> compfallsub = new ArrayList<String>();

ArrayList<String> compwintersub = new ArrayList<String>();

ArrayList<String> compsummersub = new ArrayList<String>();
```

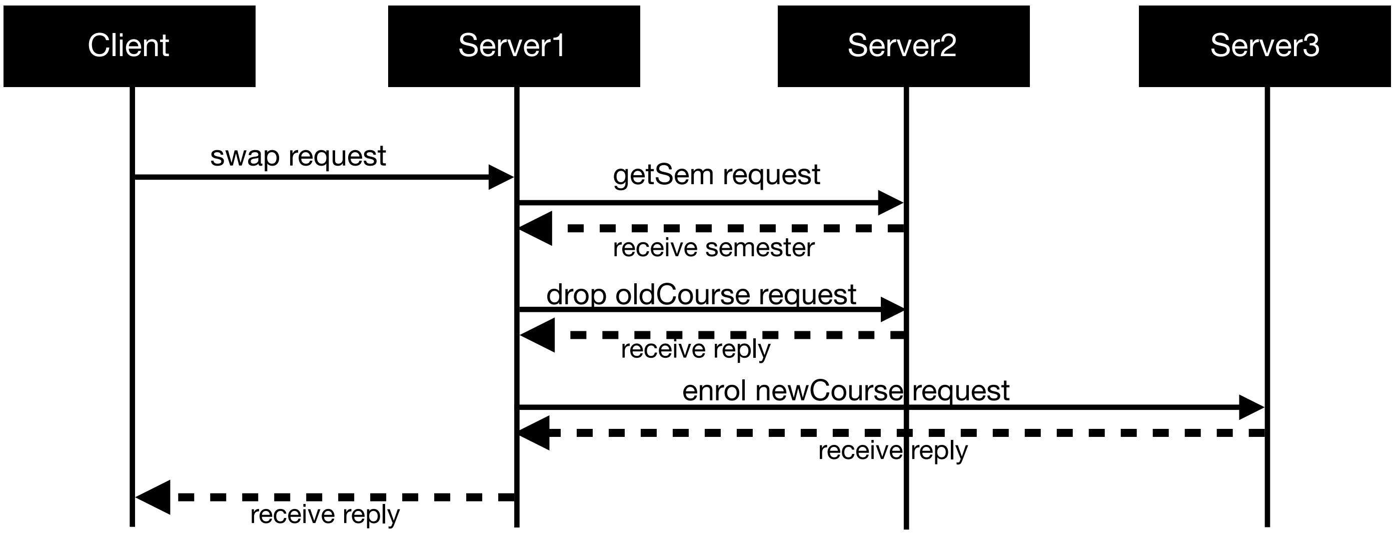
This hashmap is used to store advisor’s information as well as courses added by that advisor in all semesters. Here specific ArrayList stores the courses added by advisors in specific semester.

4. Use Case Diagram:



5. Sequence Diagram:

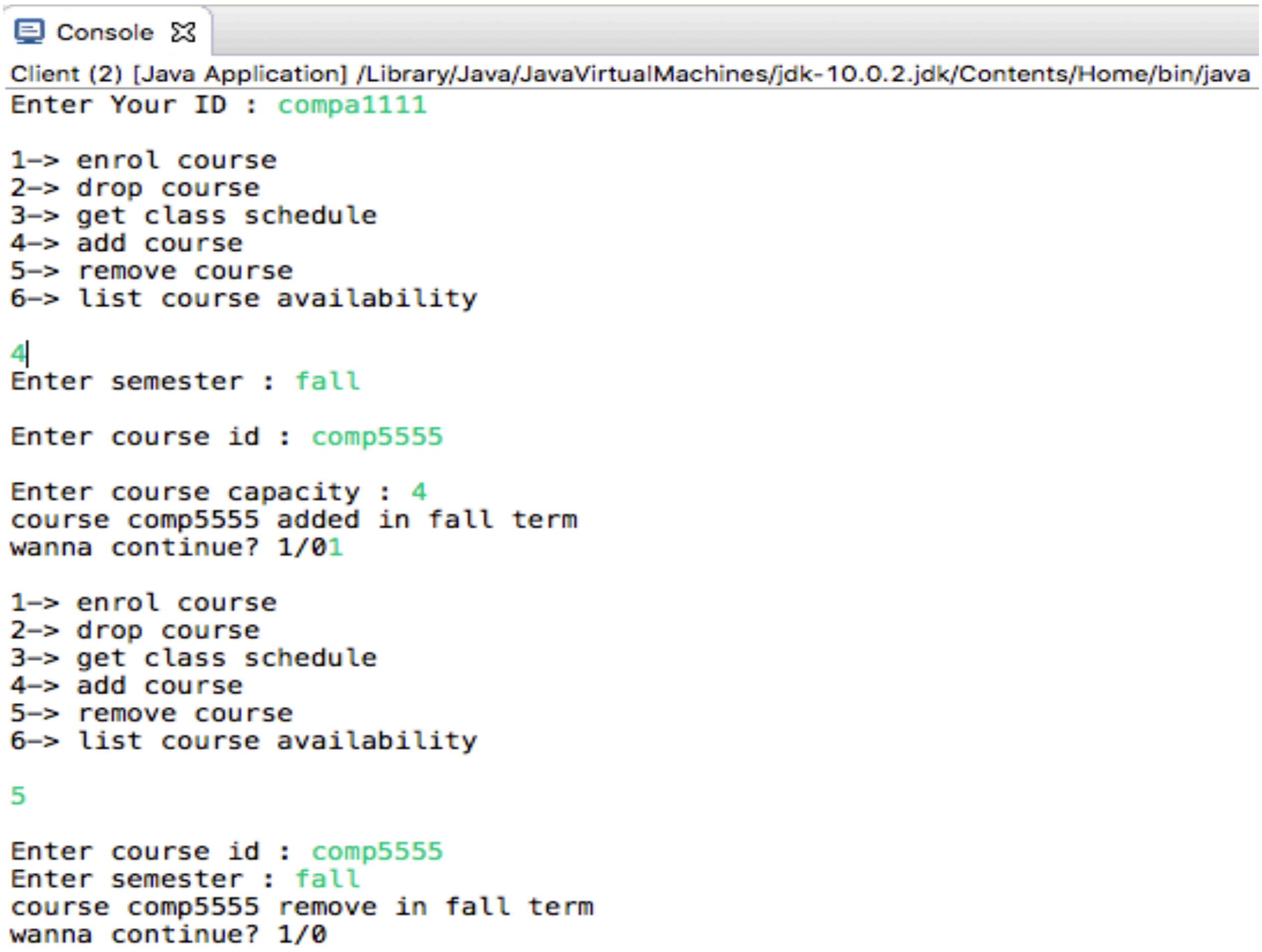
COMP student wants to swap SOEN course (oldCourse) with INSE course (newCourse)



6. Test cases:

1) addCourse 2) removeCourse

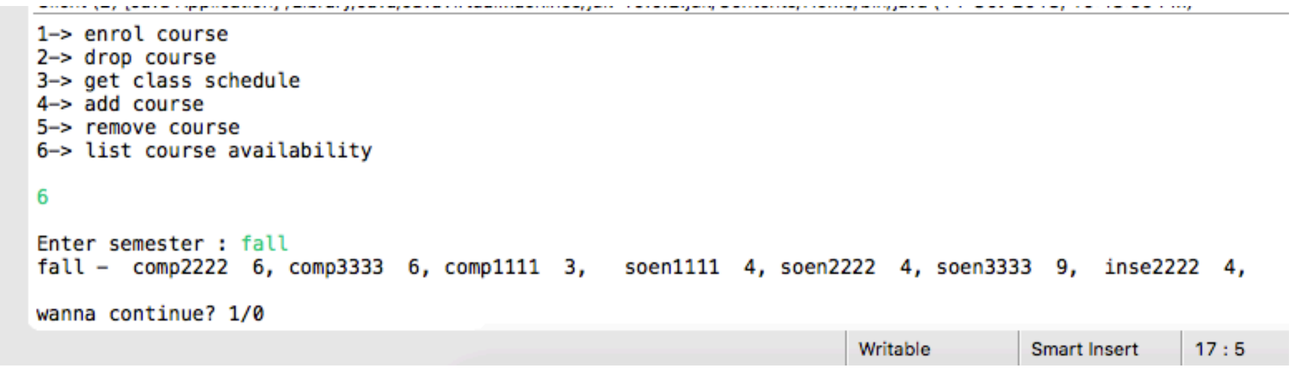
Advisor of COMP department **'compa1111'** wants to add and remove course **'comp5555'** in **'fall'** term.



```
Client (2) [Java Application] /Library/Java/JavaVirtualMachines/jdk-10.0.2.jdk/Contents/Home/bin/java
Enter Your ID : compa1111

1-> enrol course
2-> drop course
3-> get class schedule
4-> add course
5-> remove course
6-> list course availability

4
Enter semester : fall

Enter course id : comp5555

Enter course capacity : 4
course comp5555 added in fall term
wanna continue? 1/01

1-> enrol course
2-> drop course
3-> get class schedule
4-> add course
5-> remove course
6-> list course availability

5

Enter course id : comp5555
Enter semester : fall
course comp5555 remove in fall term
wanna continue? 1/0
```

3) listCourseAvailability

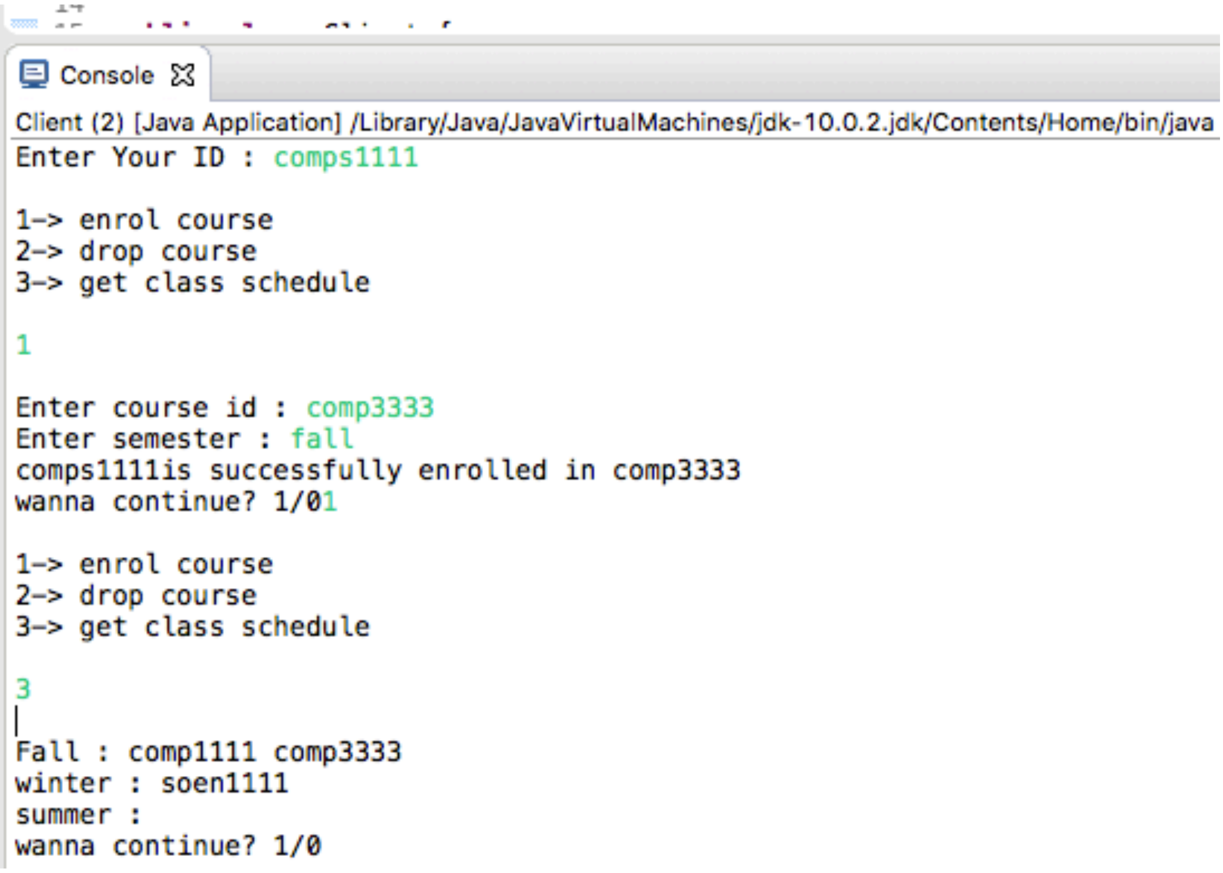
Advisor of COMP department **'compa1111'** wants to get course availability of all courses of all department in **'fall'** term.



```
1-> enrol course
2-> drop course
3-> get class schedule
4-> add course
5-> remove course
6-> list course availability

6

Enter semester : fall
fall -   comp2222  6, comp3333  6, comp1111  3,   soen1111  4, soen2222  4, soen3333  9,  inse2222  4,

wanna continue? 1/0
```

4) enrolCourse

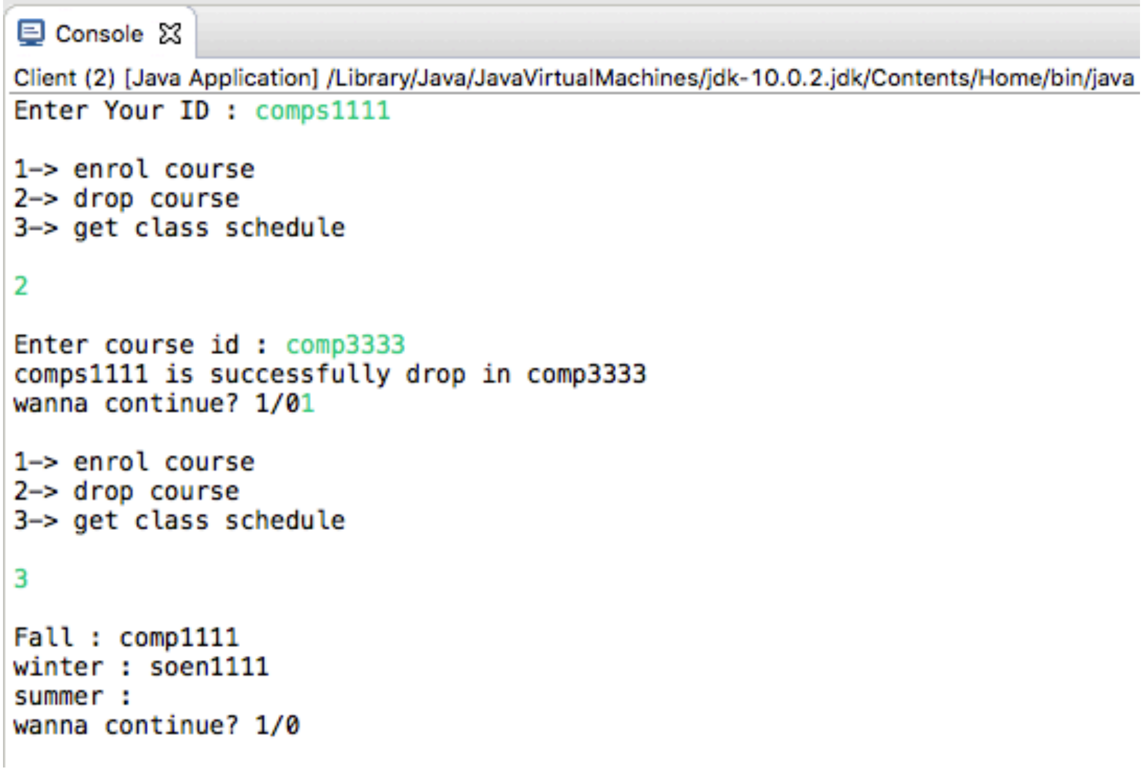
Student of COMP department **'comps1111'** wants to enrol course **'comp3333'** in **'fall'** term.



```
Client (2) [Java Application] /Library/Java/JavaVirtualMachines/jdk-10.0.2.jdk/Contents/Home/bin/java
Enter Your ID : comps1111

1-> enrol course
2-> drop course
3-> get class schedule

1

Enter course id : comp3333
Enter semester : fall
comps1111is successfully enrolled in comp3333
wanna continue? 1/01

1-> enrol course
2-> drop course
3-> get class schedule

3

Fall : comp1111 comp3333
winter : soen1111
summer :
wanna continue? 1/0
```

5) dropCourse 6) getClassSchedule

Student of COMP department **'comps1111'** wants to drop course **'comp3333'** in **'fall'** term and get class schedule of all terms.



```
Client (2) [Java Application] /Library/Java/JavaVirtualMachines/jdk-10.0.2.jdk/Contents/Home/bin/java
Enter Your ID : comps1111

1-> enrol course
2-> drop course
3-> get class schedule

2

Enter course id : comp3333
comps1111 is successfully drop in comp3333
wanna continue? 1/01

1-> enrol course
2-> drop course
3-> get class schedule

3

Fall : comp1111
winter : soen1111
summer :
wanna continue? 1/0
```

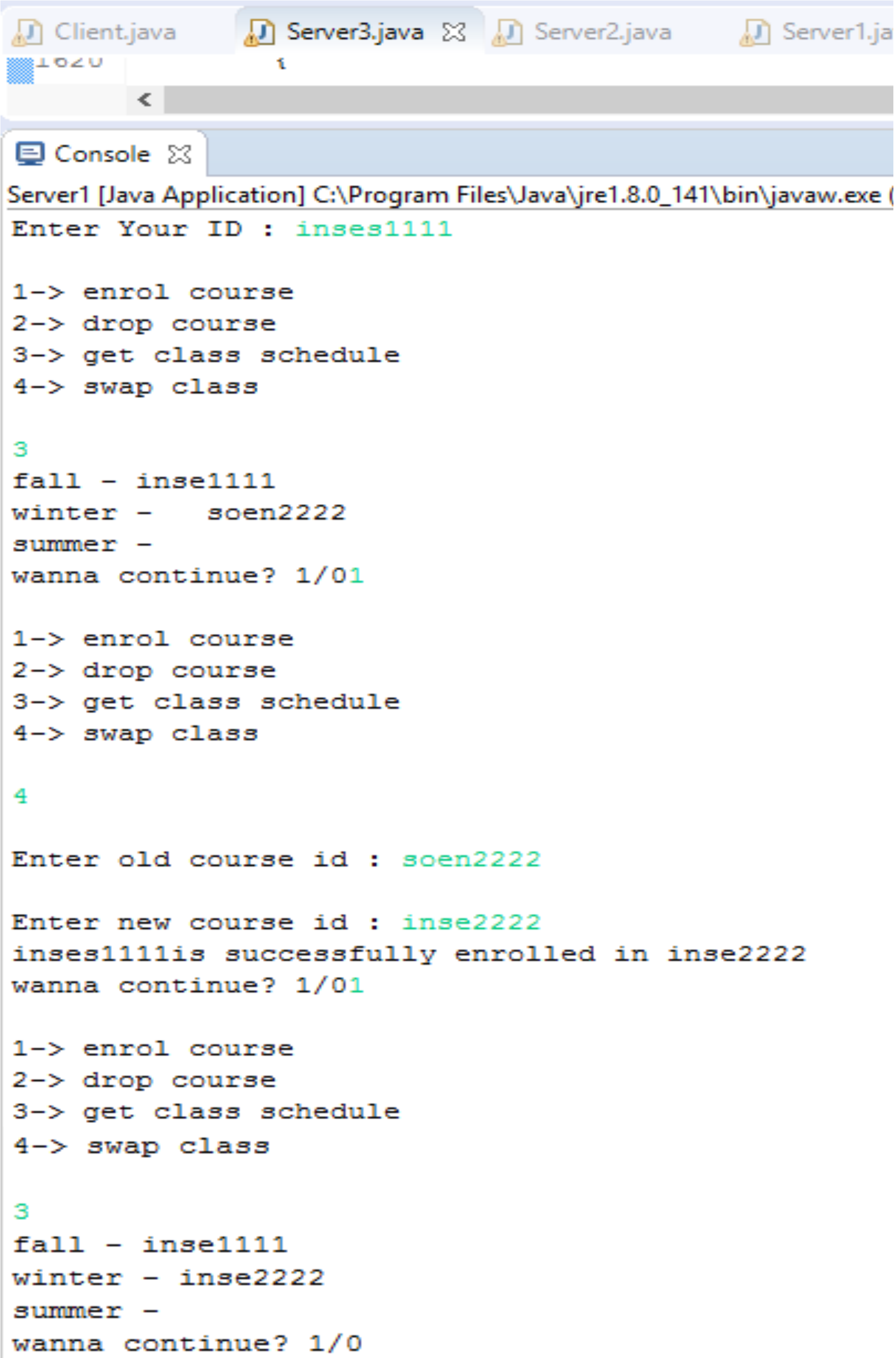
7) swapCourse

Student id: inses1111

Old course : soen2222

New course : inse2222

INSE student **‘inses1111’** wants to swap course of SOEN department with INSE department’s course.



Server1 [Java Application] C:\Program Files\Java\jre1.8.0_141\bin\javaw.exe (

```
Enter Your ID : inses1111

1-> enrol course
2-> drop course
3-> get class schedule
4-> swap class

3
fall - inse1111
winter -   soen2222
summer -
wanna continue? 1/01

1-> enrol course
2-> drop course
3-> get class schedule
4-> swap class

4

Enter old course id : soen2222

Enter new course id : inse2222
inses1111is successfully enrolled in inse2222
wanna continue? 1/01

1-> enrol course
2-> drop course
3-> get class schedule
4-> swap class

3
fall - inse1111
winter - inse2222
summer -
wanna continue? 1/0
```

7. Important/difficult part :

One difficult part of this assignment is inter-server communication in Web Services. I faced trouble many time while sending data from one server to another server. There was a small problem of how to send different information in one single message and how to separate such information like user id, semester etc in the received message on another server. I faced problems in maintaining server log files also.But the important parts of the the assignment are building the end point files before publishing the web services and import wsdl files.

8. References :

[1] . https://www.w3.org/TR/2001/NOTE-wsdl-20010315
[2] . https://www.tutorialspoint.com/uddi/uddi_overview.htm
[3] . https://en.wikipedia.org/wiki/SOAP