DISTRIBUTED SYSTEM DESIGN

COMP6231

Assignment 3

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

Submitted By – Amandeep Singh (40052070)

Contents

Overview

This assignment is the modification of the work, we did in the previous assignment. Now instead of CORBA we are using Web Services to make client-server interaction. The system functionality remains the same as previous.

The system is build using the JAX-WS (Java API for XML Web Service) and the data exchange occurs using XML over the SOAP (Simple Object Access Protocol) protocol. I have used the Java annotations such as "*@WebService, @SOAPBinding,…*" for building the web service instead of using the traditional "wsgen & wsimport" commands. The final system is thoroughly test for multiple users' access at the same time.

Since JAX-WS only supports primitive data-types by default, so in order to send custom data structure such as HashMap and SimpleEntry, I had to write mappers for the same and do a slight modification to the Enrollment interface contract. This was one of the key challenging areas which I worked upon in this project.

System Requirements

- Development Environment: Eclipse Photon
- Programming Language: Java 8, JAX-WS 2.2
- Class Diagram Creation: Object Aid Eclipse plugin

Running the Project

- Through the eclipse run the following files as Java Applications:
    - COMP_Server.java
    - SOEN_Server.java
    - INSE_Server.java
- Finally run the “Login.java” file and follow the command line menu.

Working

The functionality for this project is exactly similar to the previous one, apart that the client server interaction occurs over SOAP web service instead of CORBA. Each department server publishes an endpoint using which the client can invoke its methods.

Similar to the previous assignment, I have used the *Reentrant Lock* [1] for code synchronization. The reentrant lock offers more features than the normal synchronized blocks, such as fairness of the lock etc.

Architecture

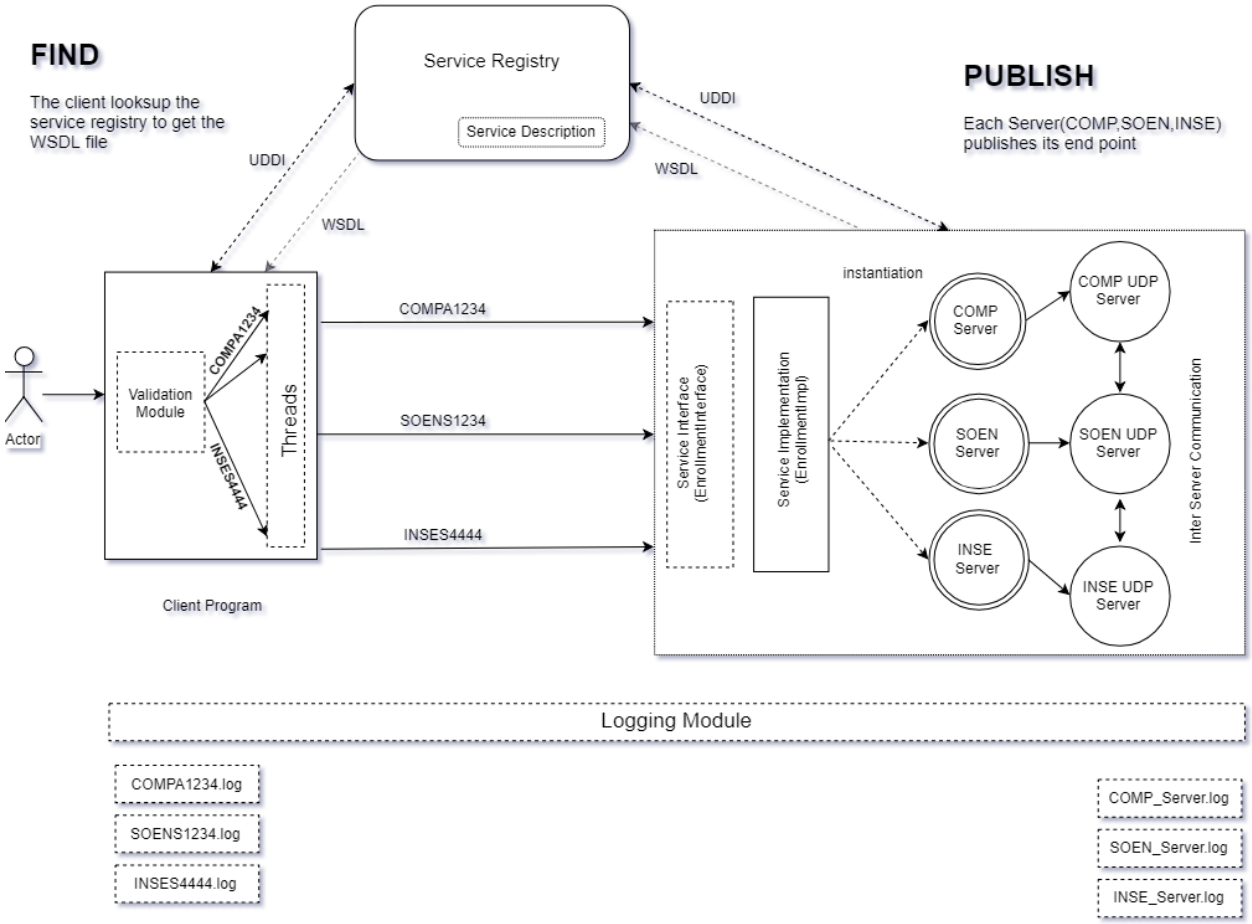


Figure 1 Project Architecture

Class Diagram

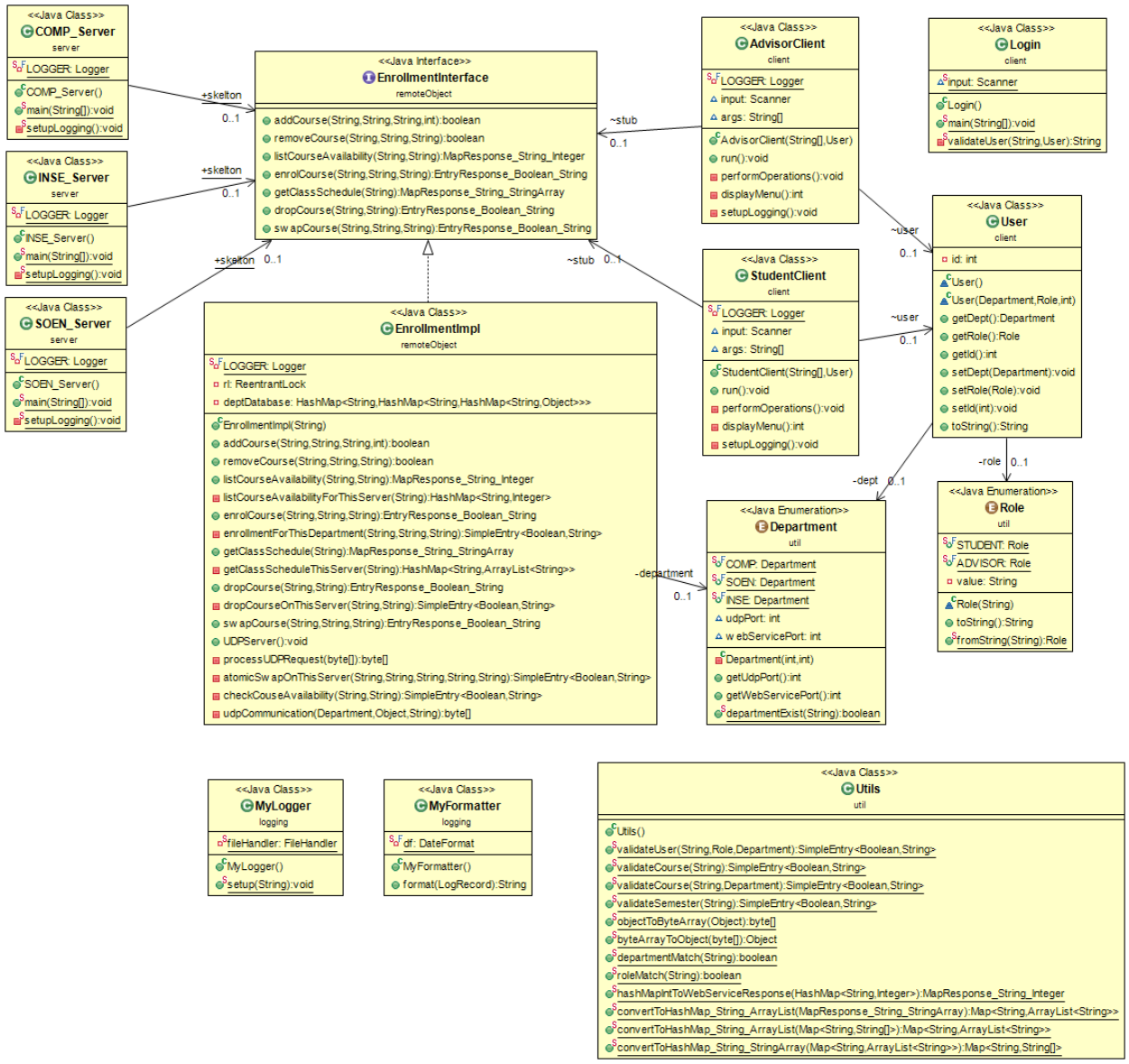


Figure 2 Class Diagrams

Key Features

Apart from the features of the previous project, following new features are added.

1. Java 8 Lambdas
   As in the previous project, I'm continuing using the Java 8 lambdas.

```
studentSchedule.forEach((sem, courses) -> {
    courses.forEach((course) -> {
        Department dept = Department.valueOf(course.substring(0, 4).toUpperCase());
        if (dept == this.department)
            departmentCourses.add(course);
        else
            outOfDepartmentCourses.add(course);
    });
});
```

Figure 3 Java Lambdas

2. Reentrant Lock [1]
   Proper synchronization of operations is obtained using the Reentrant lock [1].

```
private ReentrantLock rl;

this.rl = new ReentrantLock(true); // fair reentrant lock

//Acquire Lock
rl.lock();

 // release the lock
 rl.unlock();
```

Figure 4 Reentrant lock usage

3. Sending custom Data-Type over SOAP Web Service [2]

The Soap webservice only supports primitive and String data types, so in order to send *HashMap* and *SimpleEntry* over the web service, I have used custom adapter classes (in the "*mapper*" package of project).

```
@WebService
@SOAPBinding(style = SOAPBinding.Style.RPC)
public interface EnrollmentInterface extends Remote {

    /* Advisor Operations */

    boolean addCourse(String advisorId, String courseId, String semester, int capacity) throws RemoteExce

    boolean removeCourse(String advisorId, String courseId, String semester) throws RemoteException;

    MapResponse_String_Integer listCourseAvailability(String advisorId, String semester) throws RemoteExc

    /* Student Operations */

    EntryResponse_Boolean_String enrolCourse(String studentId, String courseId, String semester) throws R

    MapResponse_String_StringArray getClassSchedule(String studentId) throws RemoteException;

    EntryResponse_Boolean_String dropCourse(String studentId, String courseId) throws RemoteException;

    EntryResponse_Boolean_String swapCourse(String studentId,String newCourseId,String oldCourseId) throw
```

Figure 5: Inter*face modification for custom data type*

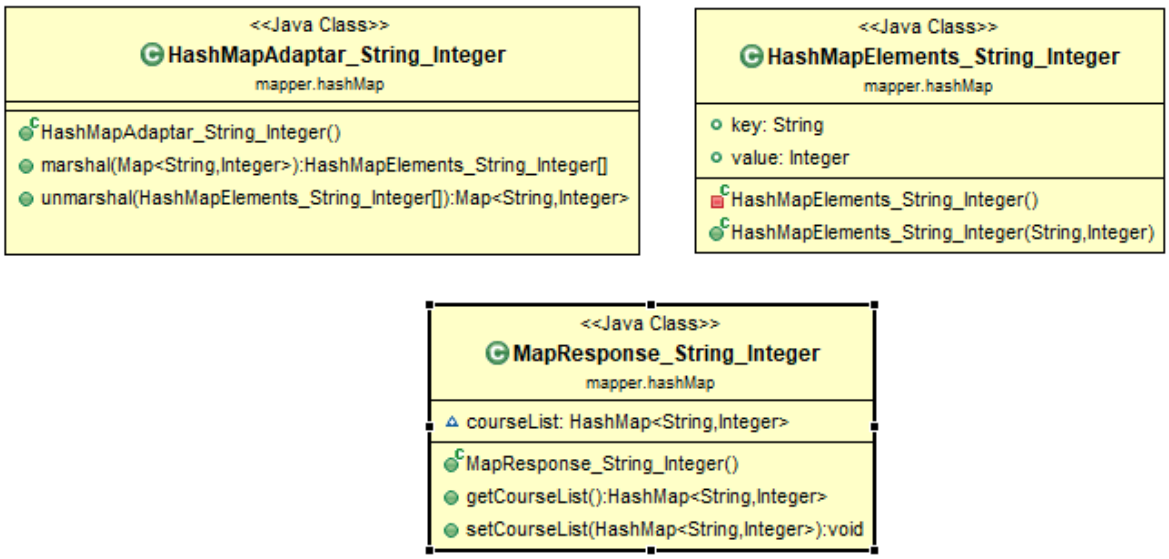


Figure 6: Mapper classes for HashMap<String,Integer>

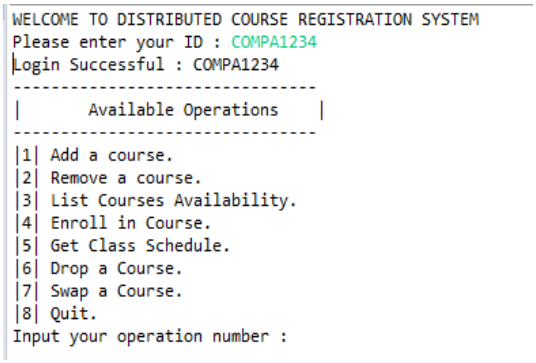
Test Cases

- Login

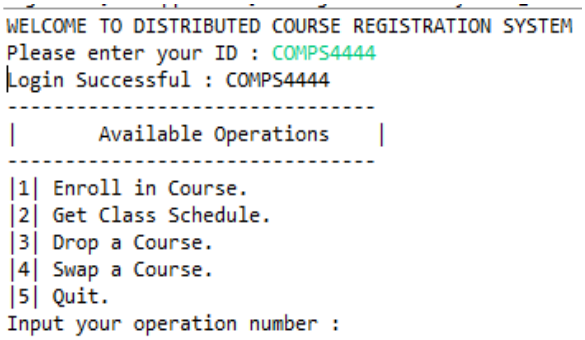
#	Operation	Input	Output	Result
1	Validate Department	ABSCA1236	Your department('ABSC') isn't recognized	PASS
2	Validate User Type	COMPY1234	Your role('Y') isn't recognized.	PASS
3	Validate Number	COMPA_qaw	Your id('_qaw') isn't recognized.	PASS
4	Case Insensitivity	COMPA1234 Compa1234	User can login	PASS

- User Specific Actions

ADVISOR



STUDENT



- Advisor Add Course

#	Operation	Input	Output	Result
1	Valid Semester	AUTUM	AUTUM isn't valid semester.	PASS
2	Validate Course Id	comp12345	Seems to be an invalid course (length not equal to 8).	PASS
3	Adding other department course	SOEN6441	You are not authorized for this department('SOEN').	PASS
4	Adding already added course	-	FAILURE = COMP6231 is already offered in FALL semester.	PASS
5	Adding same course to other semester		SUCCESS - Course Added Successfully	PASS

- Advisor Remove Course

#	Operation	Input	Output	Result
1	Valid Semester	rainy	rainy isn't valid semester.	PASS
2	Validate Course Id	comp51486	Seems to be an invalid course (length not equal to 8).	PASS
3	Course Doesn’t exist	Comp6232	FAILURE - comp6232 is not offered in FALL semester.	PASS
4	Remove other department course	SOEN6441	You are not authorized for this department('SOEN').	PASS

- Advisor List Course Availability

#	Operation	Input	Output	Result
1	Valid Semester	HOT	HOT isn't valid semester.	PASS

- Student Enroll Course

#	Operation	Input	Output	Result
1	Valid Semester	COLD	COLD isn't valid semester.	PASS
2	Validate Course Id	comp51486	Seems to be an invalid course (length not equal to 8).	PASS
3	Course Doesn’t exist	COMP6541	COMP6541 is not offered in FALL semester.	PASS
4	Already Enrolled	-	FAILURE - COMPS4444 is already enrolled in COMP6231.	PASS
5	Enrolled in 3 courses for this semester	-	COMPS4444 is already enrolled in 3 courses [COMP6231, COMP6478, COMP6985] for this FALL semester.	PASS
6	Enrolled in 2 off department courses in all the semesters	-	COMPS4444 is already enrolled in 2	PASS

			out-of-department courses.	

- Advisor get Class Schedule

#	Operation	Input	Output	Result
1	Valid StudentId	COMPS258745	Seems to be an invalid id(length not equal to 9).	PASS
2	Student is of his/her department	SOENS5142	You are not authorized for this department('SOEN').	PASS

- Student Drop Course

#	Operation	Input	Output	Result
1	Valid StudentId	COMPS258745	Seems to be an invalid id(length not equal to 9).	PASS
2	Valid course id	comp51486	You are not authorized for this department('SOEN').	PASS
3	Course not offered	COMP9854	COMP9854 isn't offered by the department yet.	PASS

- Swap Course

#	Operation	Input	Output	Result
1	Valid StudentId	COMPS258745	Seems to be an invalid id(length not equal to 9).	PASS
2	Valid New Course id	comp658578	Seems to be an invalid course (length not equal to 8).	PASS
3	Valid old course id	comp658578	Seems to be an invalid course (length not equal to 8).	PASS
4	Student not enrolled in the course to drop	comp6231	COMPS4444 is not enrolled in COMP6231	PASS
5	Student already enrolled in the new course	comp6441	COMPS4444 is already enrolled in COMP6441	PASS

6	New Course is not offered in that semester	(inse6231, inse6441)	INSE6441 is not offered in FALL semester.	PASS
7	New Couse is elective & the student already have enrolled in 2 elective subjects.	-	COMPS4444 is already enrolled in 2 out-of-department courses	PASS
8	New Course is full, old course should not be dropped.	-	COMP6231 is full.	PASS
9	Swap course offered in different semester (SHOULD NOT HAPPEN)	-	-	PASS
10	Swap a course which the student is already enrolled in another semester (SHOULD NOT HAPPEN)	-	-	PASS

- Multiple Users Access Test
  After starting all the three servers (COMP, INSE, SOEN), run the "*Assignment3/ src/ test / java/ multiThreadTest/ TestMultipleThreads.java*" test file to test the system for maximum concurrency.

References

1. Reentrant Lock: https://www.geeksforgeeks.org/reentrant-lock-java/
2. https://stackoverflow.com/questions/3941479/jaxb-how-to-marshall-map-into-keyvalue-key