COMP6231 Assignment 3

Web Service Implementation of the Distributed Appointment Management System (DAMS)

Darshak Kachchhi (40206619)

Instructor: R. Jayakumar

TA: Brijesh Lakkad

Computer Science & Engineering

Concordia University

Montreal, Quebec

Table of Contents

1. Overview

The Distributed Appointment Management System (DAMS) is a distributed system for health care; It is used by an admin of the hospitals who manages the information about the medical appointments and patients to book or cancel a medical appointment across three different hospitals Montreal (MTL), Sherbrooke (SHE), and Quebec (QUE) within a system.

Hospital admins and patients are uniquely identified by the admin id (e.g., MTLA0000) and patient id (QUEP2981) respectively. There are 3 types of appointment types for which slots can be created by the admin: Physician, Surgeon, Dental. There are three-time slots are available for each appointment type in a day. Each appointment type is a combination of city, appointment slot, and date.

Each server maintains its database using the HashMap. Client and Server are communicating using the Web Service Implementation. While inter-server communication is done by the UDP communication. Each server maintains a log file for all the operations performed by the server. Also, for each patient and admin client log file is maintained.

To make the system more robust, inter-server communication is done using the thread. Since there are multiple users are accessing the server concurrently, the proper synchronization of data is implemented in the code. All the user inputs are case insensitive.

G. Tools

- Java IDE Eclipse
- Java JDK version 1.8
- JAX-WS

2. Create & Run System in Eclipse

To create a Distributed system using Web service there are 9 steps.

a) Code the implementation class.
b) Compile the implementation class.
c) Use wsgen to generate the artifacts required to deploy the service.
d) Package the files into a WAR file.
e) Deploy the WAR file. The tie classes (which are used to communicate with clients) are generated by the Application Server during deployment.
f) Code the client class.
g) Use wsimport to generate and compile the stub files.
h) Compile the client class.
i) Run the client.

Run command of Server files

java MontrealPublisherServer
java SherbrookePublisherServer
java QuebecPublisherServer

Run command of AdminClient.java, PatientClient.java file

java AdminClient <AdminId>
java PatientClient <PatientId>

Order to run the system

- Start Montreal, Sherbrooke, and Quebec server in any order
- Start either AdminClient or PatientClient based on requirements.

Endpoint publisher for each server

- http://localhost:8789/MTLServer?wsdl
- http://localhost:8788/QUEServer?wsdl
- http://localhost:8787/SHEServer?wsdl

3. Architecture

There are three different servers MTL, QUE, and SHE. When all these servers are started, all the servers start their own UDP servers for communicating with the patient-client and server client. These servers are running all the time to listen to requests from clients.

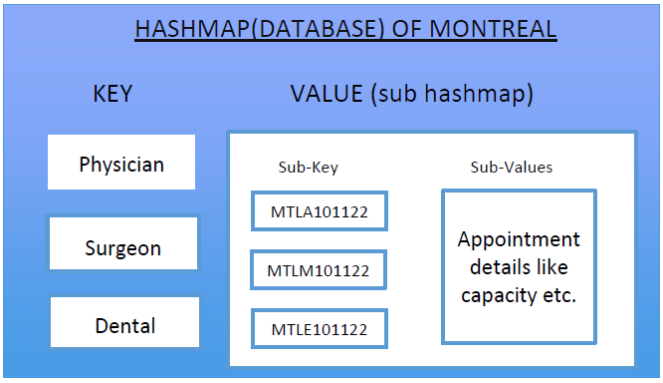
In the publish operation, a service description must be published so that a service requester can find the service. Depending on the client ID of the patient or admin, the client system will connect you to the respective server by doing look up from the service requestor retrieves the service description directly. It can be involved in two different lifecycle phases for the service requestor.

While binding, the service requestor invokes or initiates an interaction with the service at runtime using the binding details in the service description to locate, contact, and invoke the service. A service is an interface described by a service description. The service description is the implementation of the service. A service is a software module deployed on network-accessible platforms provided by the service provider. It interacts with a service requestor. Sometimes it also functions as a requestor, using other Web Services in its implementation.

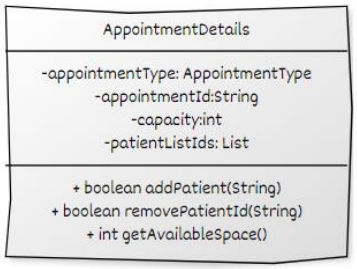
A client only communicates with their corresponding server. But there are multiple options (list appointment availability, book appointment, get schedule appointment of a patient, cancel appointment and Swap appointment) which have required to communicate with the other servers. This communication is done by the UDP socket communication. The client-server makes a UDP request to other servers concurrently and it will get the response from the other servers, and it is returned to the client.

4. Features

A. Server Database: each server has its own database, and it is implemented using HashMap.



B. Appointment Details are maintained by AppointmentDetails class object which has details of type of appointment, appointment id, capacity and list of patient id who has booked the slot.



C. Log has been implemented to track all the activities on each server as well as for each admin and patient who use the system.

Format of the log:
Request Date Time | Request Type | Request Parameters | Server Response | Status of Completion

```
11-02-2022 18:22:46 | Add appointment | [QUEA110222, SURGEON, 15] | Success: Appointment Added | false
11-02-2022 18:24:01 | Add appointment | [MTLA110222, DENTAL, 10] | Failed: Cannot book appointment Id - MTLA110222 by Quebec server | false
11-02-2022 18:35:33 | Add appointment | [QUEA110222, PHYSICIAN, 15] | Success: Appointment Added | false
11-02-2022 18:36:10 | Add appointment | [QUEE110222, PHYSICIAN, 15] | Success: Appointment Added | false
11-02-2022 18:37:03 | Add appointment | [QUEA120222, SURGEON, 10] | Success: Appointment Added | false

```

5. Test Case

Already Added Test Data

Server Name	Appointment Type	Appointment ID	Patient List
Montreal	Physician	MTLA030222	**MTLP2345, QUEP5465**
		MTLE030222	**MTLP1245, MTLP2463, MTLP9875**
		MTLM010222	**MTLP2345, MTLP9875, MTLP3246**
	Dental	**MTLM030222**	**MTLP2345, MTLP3246**
		MTLE030222	**MTLP2345**
		MTLA010222	**MTLP3246, MTLP1245, MTLP2463, MTLP5465**
		MTLA020222	**MTLP2345, MTLP5465**
	Surgeon	**MTLA030222**	**MTLP1245, MTLP2463**
		MTLM030222	**MTLP3246, MTLP9875**
Quebec	**Physician**	**QUEA040222**	**MTLP2345, QUEP5465**
	Dental	**QUEA010222**	**QUEP5465**
		QUEA020222	**QUEP5465**
Sherbrooke	**Physician**	**SHEE080222**	**MTLP2345, SHEP5565, SHEP2475**
	Dental	**SHEA050222**	**SHEP5565, SHEP2475**
	Surgeon	**SHEE070222**	**MTLP2345, SHEP5565, SHEP2475**

Test Data

Test Method	Expected Output	Actual Output
Add appointment	**Success: Appointment Added**	Enter your choice > 2 Enter appointmentID [(MTL|SHE|QUE)(A|E|M)(MMDDYY)] > QUE110222 Invalid ID Try again. Enter appointmentID [(MTL|SHE|QUE)(A|E|M)(MMDDYY)] > QUEA110222 Enter Capacity > 15 Success: Appointment Added
	Failed: Cannot book appointment Id of another server	Enter your choice > 3 Enter appointmentID [(MTL|SHE|QUE)(A|E|M)(MMDDYY)] > MTLA110222 Enter Capacity > 10 Failed: Cannot book appointment Id - MTLA110222 by Quebec server
Book Appointment	**Failed: Patient has already book appointment in the same day with**	Enter your choice > 1 Enter appointmentID [(MTL|SHE|QUE)(A|E|M)(MMDDYY)] > MTLA030222 Failed: Patient has already booked appointment in same day withPHYSICIAN

	same Appointment Type	
	Failed: No appointment available for selected slot	Enter your choice > 3 Enter appointmentID [(MTL|SHE|QUE)(A|E|M)(MMDDYY)] > MTLE110222 Failed: No appointment available for selected slot
	Failed: Patient has already booked 3 appointments other than its server	Enter your choice > 1 Enter appointmentID [(MTL|SHE|QUE)(A|E|M)(MMDDYY)] > QUEA120222 Failed: Patient has already booked 3 appointment other than Montreal Server
	Success: appointment successfully booked	Enter your choice > 1 Enter appointmentID [(MTL|SHE|QUE)(A|E|M)(MMDDYY)] > QUEA110222 Success: Appointment successfully booked
Remove appointment	**Success: Appointment is removed**	Enter your choice > 1 Enter appointmentID [(MTL|SHE|QUE)(A|E|M)(MMDDYY)] > QUEA120222 Success: Appointment is removed with appointmentId: QUEA120222
	Success: Appointment is removed with not available next appointment slot	Enter your choice > 1 Enter appointmentID [(MTL|SHE|QUE)(A|E|M)(MMDDYY)] > QUEA110222 Success: Not able to find available appointment and removed the appointmentId: QUEA110222
	Success: Appointment is removed with patient is transferred	Enter your choice > 3 Enter appointmentID [(MTL|SHE|QUE)(A|E|M)(MMDDYY)] > MTLA010222 Success: Patient is transferred to available appointment and removed the appointmentId: MTLA010222
	Failed: No slot available	Enter your choice > 1 Enter appointmentID [(MTL|SHE|QUE)(A|E|M)(MMDDYY)] > MTLA220222 Failed: No appointment is available with appointmentId: MTLA220222
List Appointment	**Success: All Appointment list**	Enter your choice > 3 ========== Appointment Type ========== 1. Physician 2. Surgeon 3. Dental 4. Exit Enter your choice > 2 List appointment SURGEON: [MTLA030222 0, MTLM030222 0, SHEE070222 0,]

Get Appointment Schedule	**Success: Empty appointment schedule**	Enter your choice > 2 Appointment Schedule of MTLP3574 []
	Success: Appointment Schedule	``` MENU 1. Book Appointment 2. Get Appointment SChedule 3. Cancel Appointment 4. Exit Enter your choice > 2 Appointment Schedule of MTLP2345 [PHYSICIAN:MTLA030222,PHYSICIAN:MTLM010222,DENTAL:MTLE030222 ```
Cancel Appointment	**Success: cancelled appointment**	Enter your choice > 3 Enter appointmentID [(MTL|SHE|QUE)(A|E|M)(MMDDYY)] > MTLA030222 Success: cancelled appointment with MTLA030222
	Failed: No record of appointment found	Enter your choice > 3 Enter appointmentID [(MTL|SHE|QUE)(A|E|M)(MMDDYY)] > QUEM110222 Failed: No record found of [MTLP2345, QUEM110222]
Swap Appointment	**Success: Appointment swapped**	Enter your choice > 4 OLD APPOINTMENT DETAILS Enter appointmentID [(MTL|SHE|QUE)(A|E|M)(MMDDYY)] > MTLA030222 ========== Appointment Type ========== 1. Physician 2. Surgeon 3. Dental 4. Exit Enter your choice > 1 NEW APPOINTMENT ID Enter appointmentID [(MTL|SHE|QUE)(A|E|M)(MMDDYY)] > MTLA230222 ========== Appointment Type ========== 1. Physician 2. Surgeon 3. Dental 4. Exit Enter your choice > 1 Success: appointment swapped

Swap Appointment	**Failed: No appointment booked of old Appointment ID**	Enter your choice > 4 OLD APPOINTMENT DETAILS Enter appointmentID [(MTL|SHE|QUE)(A|E|M)(MMDDYY)] > MTLA230222 ========== Appointment Type ========== 1. Physician 2. Surgeon 3. Dental 4. Exit Enter your choice > 2 NEW APPOINTMENT ID Enter appointmentID [(MTL|SHE|QUE)(A|E|M)(MMDDYY)] > MTLA030222 ========== Appointment Type ========== 1. Physician 2. Surgeon 3. Dental 4. Exit Enter your choice > 1 Failed: No record found of [MTLP2345, MTLA230222]
	Failed: No appointment available for new appointment Id	Enter your choice > 4 OLD APPOINTMENT DETAILS Enter appointmentID [(MTL|SHE|QUE)(A|E|M)(MMDDYY)] > MTLA230222 ========== Appointment Type ========== 1. Physician 2. Surgeon 3. Dental 4. Exit Enter your choice > 1 NEW APPOINTMENT ID Enter appointmentID [(MTL|SHE|QUE)(A|E|M)(MMDDYY)] > MTLE240222 ========== Appointment Type ========== 1. Physician 2. Surgeon 3. Dental 4. Exit Enter your choice > 1 Failed: No appointment available for selected slot