

CSC570 I WEB SERVICES

FINAL PROJECT

DOCTOR-PATIENT MANAGEMENT

By,

Prateek Mulye

Under Guidance of: Dr. Kamyar Dezhgosha

1. Implementation Design

- **Development Approach:**

This project has been developed using JAX-RS API using Jersey framework. There are two datafiles which acts as database, doctors.db and patients.db. Doctors datafile contains the information of doctors like the doctor's name and number of patients each doctor consults. Whereas the patients datafile has patients name and patients insurance number. The code is spread over six java files, Doctor.java and Patients.java are the POJO classes. DoctorsList.java and PatientsList.java creates the lists of all doctors and patients extracted from the datafiles respectively and assigning ID's with autoincrement integer to the individual doctors and patients entity. DoctorRS.java has the REST API implementation for all the CRUD operation such as GET, POST, PUT and DELETE.

This Service gives all the list of doctors and associated patients from a hospital in plain text as well as XML. It also provides list of a single doctor with all associated patients in plain text and XML.

- **Problems Faced:**

No crucial problems were faced in terms of development of the Web Service. Used the breakpoint concept for debugging the java code.

- **CURL utility:**

I used curl utility to run the POST, PUT and DELETE operation on the service. The use of curl utility can be seen in the below screenshots for all the POST, PUT and DELETE operation.

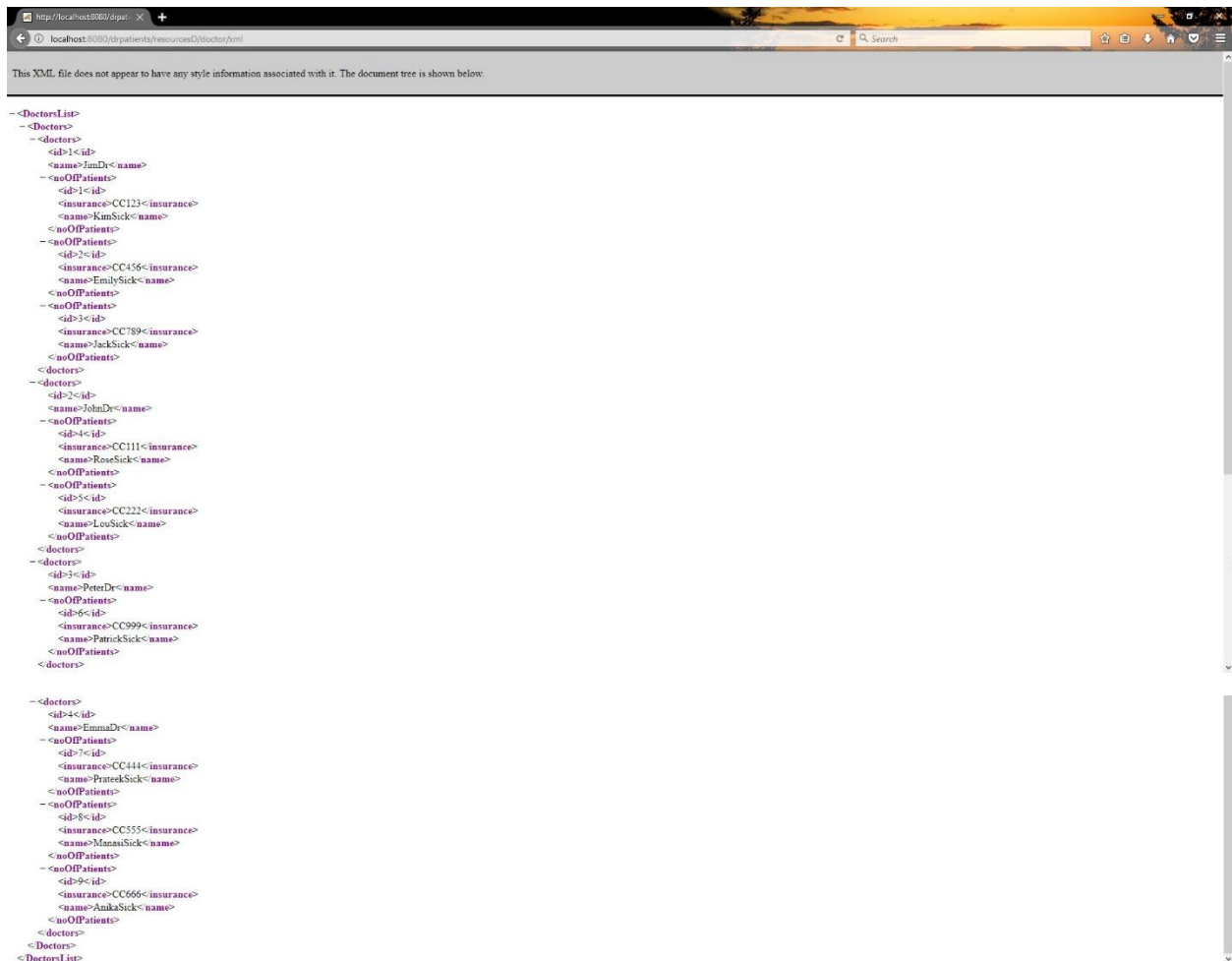
- **Web Service API used:**

I used JAX-RS API using Jersey Framework version 1.10 for implementation of this project.

2. Operations available on Doctor-Patient resource.

I. Get a List of all the doctors along with their patients in XML and plain text.

a. XML



This XML file does not appear to have any style information associated with it. The document tree is shown below:

```
--<DoctorsList>
--<Doctors>
--<doctors>
  <id>1</id>
  <name>JimDr</name>
  --<noOPatient>
    <id>1</id>
    <insurance>CC123</insurance>
    <name>KimSick</name>
  </noOPatient>
  --<noOPatient>
    <id>2</id>
    <insurance>CC456</insurance>
    <name>EmilySick</name>
  </noOPatient>
  --<noOPatient>
    <id>3</id>
    <insurance>CC789</insurance>
    <name>JackSick</name>
  </noOPatient>
</doctors>
--<doctors>
  <id>2</id>
  <name>JohnDr</name>
  --<noOPatient>
    <id>4</id>
    <insurance>CC111</insurance>
    <name>RoseSick</name>
  </noOPatient>
  --<noOPatient>
    <id>5</id>
    <insurance>CC222</insurance>
    <name>LeoSick</name>
  </noOPatient>
</doctors>
--<doctors>
  <id>3</id>
  <name>PeterDr</name>
  --<noOPatient>
    <id>6</id>
    <insurance>CC999</insurance>
    <name>PatrickSick</name>
  </noOPatient>
</doctors>
--<doctors>
  <id>4</id>
  <name>EmmaDr</name>
  --<noOPatient>
    <id>7</id>
    <insurance>CC444</insurance>
    <name>PratekSick</name>
  </noOPatient>
  --<noOPatient>
    <id>8</id>
    <insurance>CC555</insurance>
    <name>MananSick</name>
  </noOPatient>
  --<noOPatient>
    <id>9</id>
    <insurance>CC666</insurance>
    <name>AnikaSick</name>
  </noOPatient>
</doctors>
</Doctors>
</DoctorsList>
```

b. Plain Text

```
http://localhost:8080/doctor/plan
1) JimDr--- 1: KimSick ----- CC123
2: EmilySick ----- CC456
3: JackSick ----- CC789
2) JohnDr--- 4: RoseSick ----- CC111
5: LouSick ----- CC222
3) PeterDr--- 6: PatrickSick ----- CC999
4) EmmaDr--- 7: PrateekSick ----- CC444
8: ManasiSick ----- CC555
9: AnikaSick ----- CC666
```

II. Get a List of an individual doctor and his/her patients in XML and plain text.

a. XML of Doctor with ID 4

```
http://localhost:8080/doctor/xml/4
This XML file does not appear to have any style information associated with it. The document tree is shown below:
<doctor>
  <id>4</id>
  <name>EmmaDr</name>
  <noOfPatients>
    <id>7</id>
    <insurance>CC444</insurance>
    <name>PrateekSick</name>
  </noOfPatients>
  <noOfPatients>
    <id>8</id>
    <insurance>CC555</insurance>
    <name>ManasiSick</name>
  </noOfPatients>
  <noOfPatients>
    <id>9</id>
    <insurance>CC666</insurance>
    <name>AnikaSick</name>
  </noOfPatients>
</doctor>
```

b. Plain Text of Doctor with ID 4

```
http://localhost:8080/doctor/plan/2
2) JohnDr--- 4: RoseSick ----- CC111
5: LouSick ----- CC222
```

III. Add or Create a new doctor along with his/her patients to the existing list.

In this operation, I added a new doctor with name HarryDr having 2 new patients. TimothySick and TonySick via the curl utility using POST command.

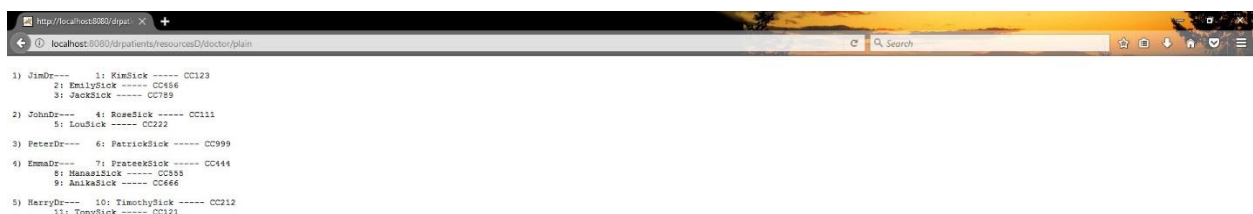
a. Before Adding new doctor.



b. POST operation using curl utility.



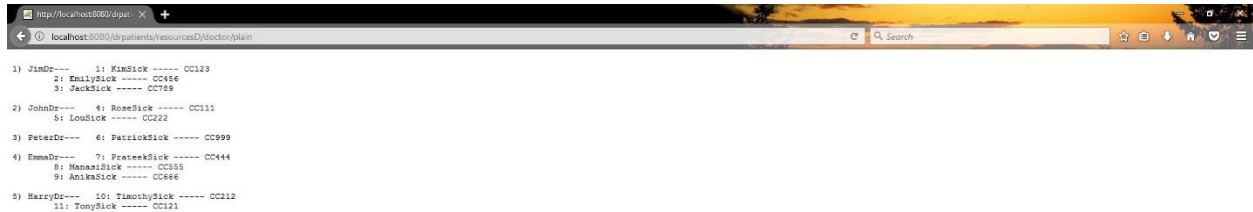
c. After POST Plain Text.



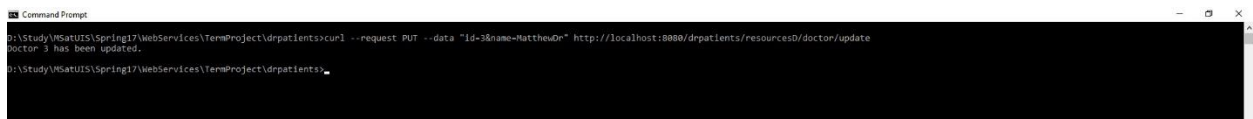
IV. Update (modify) a doctor's name only on an existing list of doctor-patients list. (Give an ID for a doctor along with a new name).

In this operation, I have modified the doctor with ID 3 having his name changed from PeterDr to MatthewDr.

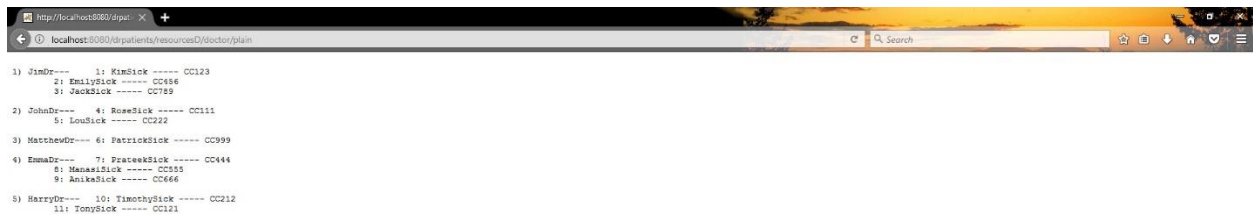
a. Before updating the doctor.



b. PUT operation using curl utility on doctor with ID 3.



c. After PUT Plain Text.



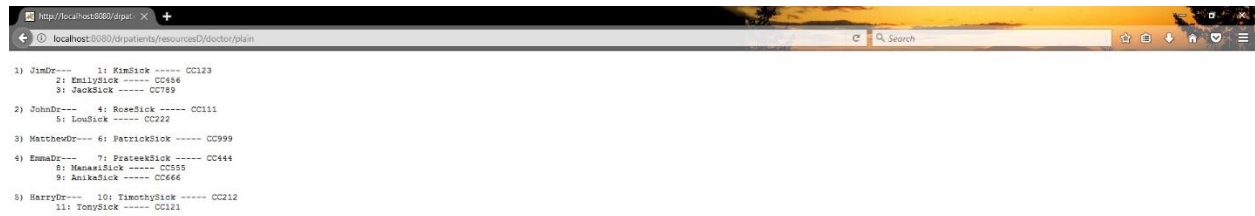
Single output.



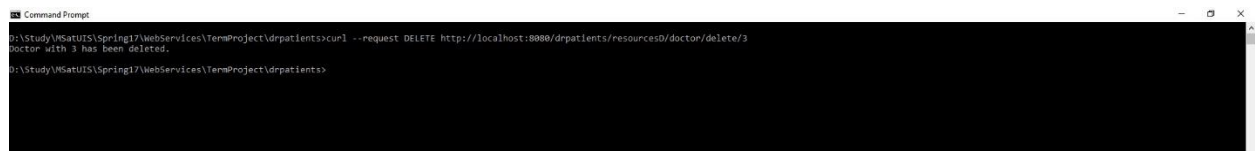
V. Delete or remove an existing doctor along with his/her patients. (Given an ID for a doctor)

In this operation, I have deleted the doctor having ID 3 along with all his patients.

a. Before delete operation.



b. DEETE operation using curl utility on doctor with ID 3.



c. After Delete operation.

