CS 548—Spring 2016 Enterprise SoftwareArchitecture and Design

AssignmentSix—Service-Oriented Architecture

Name: Junye Lu

In this assignment, I need to define a SOA for thedomain-driven design that I developed in theprevious assignment. I create data transfer objectsfor my SOA and service facades for access to mydomain model in terms of DTOs transferred to andfrom client. Also I create a Web service (SOAP andWSDL) interface for clients that want to access myapplication over the Web.

Data transfer objects(DTOs)

In this assignment, I define the XML schemas for the DTOs and then generate the java class from the schemas. The XML schemas will be shown in the appendix part. There are three kinds of DTOs for patient, provider and treatment objects respectively. The DTO of patient includes database id, patient id, patient name, and a list of treatment ids. Like patient, provider DTO has database id for provider, provider id, provider name, and a list of treatment ids. The treatment DTO is more complex than patient and provider. It contains database id for treatment, diagnosis, provider and patient related to that treatment and treatments. The treatments element is treatment type. The treatment type has a choice element for three kinds of treatment: drugtreatment, radiology and surgery. Then I define the specific information for each kind of treatment. In addition, I use the visitor pattern to encapsulate treatments in the patient and provider objects, to enforce the aggregate patter. In the service facades, I need to define visitor objects that construct treatment DTOs from the underlying PDOs.

```
public class TreatmentExporter implements ITreatmentExporter<TreatmentDto> {

    private ObjectFactory factory = new ObjectFactory();

    @Override
    public TreatmentDto exportDrugTreatment(long tid, Provider p, String diagnosis, String drug, float dosage) {
        TreatmentDto dto = factory.createTreatmentDto();
        dto.setDiagnosis(diagnosis);
        DrugTreatmentType drugInfo = factory.createDrugTreatmentType();
        drugInfo.setDosage(dosage);
        drugInfo.setName(drug);
        dto.setDrugTreatmentType(drugInfo);
        return dto;
    }

    @Override
    public TreatmentDto exportRadiology(long tid, Provider p, String diagnosis, List<Date> dates) {
        // TODO Auto-generated method stub
        TreatmentDto dto = factory.createTreatmentDto();
        dto.setDiagnosis(diagnosis);
        RadiologyType radiology = factory.createRadiologyType();
        radiology.getDate().addAll(dates);
        dto.setRadiology(radiology);
        return dto;
    }

    @Override
    public TreatmentDto exportSurgery(long tid, Provider p, String diagnosis, Date date) {
        // TODO Auto-generated method stub
        TreatmentDto dto = factory.createTreatmentDto();
        dto.setDiagnosis(diagnosis);
        SurgeryType surgery = factory.createSurgeryType();
        surgery.setDate(date);
        dto.setSurgery(surgery);
        return dto;
    }

}
```

Figure 1. visitor objects

Service Facades

In EJB project, the patient service façade provides operations for:

1. Adding a patient to a clinic. The operation takes a patient DTO and returns a new unique patient key. If the patient with that patient id already exists in the database, there will be a patient service exception thrown out.
2. Obtaining a single patient DTO, given a patient key (database primary key). Also if the patient is not in the database, then a patient service exception will be thrown.
3. Obtaining a single patient DTO, given a patient id. An exception will be thrown, if the patient is not found.
4. Return the DTO for a treatment, given a patient key and the key for the treatment.

```
public PatientService() {
    // TODO initialize factories
    patientFactory = new PatientFactory();
    patientDtoFactory = new PatientDtoFactory();
}

// TODO use dependency injection and EJB lifecycle methods to initialize
// DAOs
@Inject
@ClinicDomain
private EntityManager em;

@PostConstruct
private void initialization() {
    patientDAO = new PatientDAO(em);
}
```

Figure 2. factories and entitymanager initialization

```
/**
 * @see IPatientService#addPatient(String, Date, long)
 */
@Override
public long addPatient(PatientDto dto) throws PatientServiceExn {
    // Use factory to create patient entity, and persist with DAO
    try {
        Patient patient = patientFactory.createPatient(dto.getPatientId(), dto.getName(), dto.getDob(),
                dto.getAge());
        patientDAO.addPatient(patient);
        return patient.getId();
    } catch (PatientExn e) {
        throw new PatientServiceExn(e.toString());
    }
}

/**
 * @see IPatientService#getPatient(long)
 */
@Override
public PatientDto getPatient(long id) throws PatientServiceExn {
    // TODO use DAO to get patient by database key
    try {
        Patient patient = patientDAO.getPatient(id);
        PatientDto patientDto = patientDtoFactory.createPatientDto(patient);
        return patientDto;
    } catch (PatientExn e) {
        throw new PatientServiceExn(e.toString());
    }
}
```

Figure 3. Add pratient and get patient, given a database id.

```
/**
 * @see IPatientService#getPatientByPatId(long)
 */
@Override
public PatientDto getPatientByPatId(long pid) throws PatientServiceExn {
    // TODO use DAO to get patient by patient id
    try {
        Patient patient = patientDAO.getPatientByPatientId(pid);
        PatientDto patientDto = patientDtoFactory.createPatientDto(patient);
        return patientDto;
    } catch (PatientExn e) {
        throw new PatientServiceExn(e.toString());
    }
}
```

Figure 4. get patient, given a patient id

```
@Override
public TreatmentDto getTreatment(long id, long tid)
        throws PatientNotFoundExn, TreatmentNotFoundExn, PatientServiceExn {
    // Export treatment DTO from patient aggregate
    try {
        Patient patient = patientDAO.getPatient(id);
        TreatmentExporter visitor = new TreatmentExporter();
        return patient.exportTreatment(tid, visitor);
    } catch (PatientExn e) {
        throw new PatientNotFoundExn(e.toString());
    } catch (TreatmentExn e) {
        throw new PatientServiceExn(e.toString());
    }
}
```

Figure 5. get treatment, given a patient key and treatment key

The provider service façade operations for: adding a provider to a clinic, obtaining a single provider by a provider id, obtaining a single provider by a provider key, adding a treatment for a patient, given a treatment DTO. The first three operations are almost the same with the patient operations, just change patient to provider. The last operation is different form the patient.

```
public long addDrugTreatment(TreatmentDto t) throws ProviderServiceExn{
        try {
            Provider provider = providerDAO.getProviderByProviderId(t.getProvider());
            String diagnosis = t.getDiagnosis();
            String drugname = t.getDrugTreatmentType().getName();
            float dosage = t.getDrugTreatmentType().getDosage();
            Treatment drugTreatment = treatmentFactory.createDrugTreatment(diagnosis, drugname, dosage);
            drugTreatment.setPatient(patientDAO.getPatientByPatientId(t.getPatient()));
            provider.addTreatment(drugTreatment);
            return drugTreatment.getId();
        } catch (ProviderExn e) {
            throw new ProviderServiceExn(e.toString());
        } catch (PatientExn f) {
            throw new ProviderServiceExn(f.toString());
        }
}
/**
 * @see IProviderService#addSurgery(TreatmentDto)
 */
public long addSurgery(TreatmentDto t) throws ProviderServiceExn{
        try{
            Provider provider = providerDAO.getProviderByProviderId(t.getProvider());
            String diagnosis = t.getDiagnosis();
            Date date = t.getSurgery().getDate();
            Treatment surgery = treatmentFactory.createSurgery(diagnosis, date);
            surgery.setPatient(patientDAO.getPatientByPatientId(t.getPatient()));
            return provider.addTreatment(surgery);
        } catch(ProviderExn e){
            throw new ProviderServiceExn(e.toString());
        } catch(PatientExn f){
            throw new ProviderServiceExn(f.toString());
        }
}
/**
 * @see IProviderService#addRadiology(TreatmentDto)
 */
public long addRadiology(TreatmentDto t) throws ProviderServiceExn{
        try{
            Provider provider = providerDAO.getProviderByProviderId(t.getProvider());
            String diagnosis = t.getDiagnosis();
            List<Date> list = t.getRadiology().getDate();
            Treatment radiology = treatmentFactory.createRadiology(diagnosis, list);
            radiology.setPatient(patientDAO.getPatientByPatientId(t.getPatient()));
            return provider.addTreatment(radiology);
        } catch(ProviderExn e){
            throw new ProviderServiceExn(e.toString());
        } catch (PatientExn f){
            throw new ProviderServiceExn(f.toString());
        }
}
```

Figure 6. adding a treatment for a patient.

As there are three kinds of treatments, so I define three methods corresponding to each treatment. The principle is the same. I use the treatment DTO to get information about diagnosis, provider, patient, and the specific information for each treatment. Then a treatment is created by treatment factory. And add the treatment by the method of provider addtreatment.

Session beans

I define my service facades as stateless session beans. In order to retrieve PDOs from the database, my EJBs will have to use DAOs for the entity classes. I use dependency injection to inject an entity manager into my service façade, and then instantiate a patient or provider DAO with that entity manager.

```
@Inject
@ClinicDomain
private EntityManager em;

@PostConstruct
private void initialization() {
    providerDAO = new ProviderDAO(em);
    patientDAO = new PatientDAO(em);
}
```

Figure 7. dependency injection of entity manager

Using the InitBean to test the functionality of my service. The test result will shown in the server log.

```
@Inject
private IPatientServiceLocal service;
@Inject
private IProviderServiceLocal servicepro;
```

Figure 8. dependency injection of service

Web Services

I define web services that provide access to my service facades outside my own middleware. I deploy the service as a plain old java object(POJO) in the web tier. I define web service interfaces and implementation classes, annotated with the @WebService annotation, in a dynamic web project.

```
@WebService(
        endpointInterface = "edu.stevens.cs548.clinic.service.web.soap.IPatientWebService",
        targetNamespace = "http://cs548.stevens.edu/clinic/service/web/soap/patient",
        serviceName = "PatientWebService", portName = "PatientWebPort")

@SOAPBinding(
        style=SOAPBinding.Style.DOCUMENT,
        use=SOAPBinding.Use.LITERAL,
        parameterStyle=SOAPBinding.ParameterStyle.WRAPPED)

public class PatientWebService implements IPatientWebService {

    // TODO use CDI to inject the service
    @EJB(beanName="PatientServiceBean")
    IPatientServiceLocal service;
```

Figure 9. patient web service

```
@WebService(
        endpointInterface = "edu.stevens.cs548.clinic.service.web.soap.IProviderWebService",
        targetNamespace = "http://cs548.stevens.edu/clinic/service/web/soap/provider",
        serviceName = "ProviderWebService", portName = "ProviderWebPort")

@SOAPBinding(
        style=SOAPBinding.Style.DOCUMENT,
        use=SOAPBinding.Use.LITERAL,
        parameterStyle=SOAPBinding.ParameterStyle.WRAPPED)

public class ProviderWebService implements IProviderWebService {

    @EJB(beanName="ProviderServiceBean")
    IProviderServiceLocal service;
```

Figure 10. provider web service

A siteInfo() method in the web service interface, which returns a string value defined in the deployment descriptor for the session bean.

```java
// TODO inject resource value
@Resource(name = "SiteInfo")
private String siteInformation;

@Override
public String siteInfo() {
    return siteInformation;
}
```

Figure 11. siteInfo method

Appendix

1. XML schemas for DTOs

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<schema targetNamespace="http://cs548.stevens.edu/clinic/dto"
    elementFormDefault="unqualified" xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:tns="http://cs548.stevens.edu/clinic/dto"
    attributeFormDefault="unqualified">

    <element name="patient-dto">
        <complexType>
            <sequence>
                <element name="id" type="long" />
                <element name="patient-id" type="long" />
                <element name="name" type="string" />
                <element name="dob" type="date" />
                <element name="age" type="int" />
                <element name="treatments" type="long" nillable="true"
                    minOccurs="0" maxOccurs="unbounded" />
            </sequence>
        </complexType>
    </element>

    <element name="treatment-dto">
        <complexType>
            <sequence>
                <element name="id" type="long"/>
                <element name="diagnosis" type="string"/>
                <element name="patient" type="long"/>
                <element name="provider" type="long"/>
                <!-- Use choice element to add treatment-specific information -->
                <!-- Define complex types for treatment types below. -->
                <element name="treatments" type="tns:TreatmentType" />
                <element name="drugTreatmentType" type="tns:DrugTreatmentType" />
                <element name="radiology" type="tns:RadiologyType" />
                <element name="surgery" type="tns:SurgeryType" />
            </sequence>
        </complexType>
```

```
    <complexType name="TreatmentType">
        <sequence>
            <choice>
                <element name="drug-treatment" type="tns:DrugTreatmentType"/>
                <element name="radiology" type="tns:RadiologyType"/>
                <element name="surgery" type="tns:SurgeryType"/>
            </choice>
        </sequence>
    </complexType>

    <complexType name="DrugTreatmentType">
        <sequence>
            <element name="name" type="string"/>
            <element name="dosage" type="float"/>
        </sequence>
    </complexType>

    <complexType name="RadiologyType">
        <sequence>
            <element name="date" type="date" minOccurs="1" maxOccurs="unbounded"/>
        </sequence>
    </complexType>

    <complexType name="SurgeryType">
        <sequence>
            <element name="date" type="date"/>
        </sequence>
    </complexType>

    <element name="provider-dto">
        <complexType>
            <sequence>
                <element name="id" type="long" />
                <element name="provider-id" type="long" />
                <element name="name" type="string" />
                <element name="specialization" type="string" />
                <element name="treatments" type="long" nillable="true"
                    minOccurs="0" maxOccurs="unbounded" />
            </sequence>
        </complexType>
    </element>

</schema>
```

2. WSDL for my web service

It is difficult to show the xml file in this file. You can also see the xml in my root directory.

PatientWebService.xml

```
<?xml version='1.0' encoding='UTF-8'?><!-- Published by JAX-WS RI (http://jax-ws.java.net). RI's version is
Metro/2.3.1-b419 (branches/2.3.1.x-7937; 2014-08-04T08:11:03+0000) JAXWS-RI/2.2.10-b140803.1500 JAXWS-API/2.2.11
JAXB-RI/2.2.10-b140802.1033 JAXB-API/2.2.12-b140109.1041 svn-revision#unknown. --><!-- Generated by JAX-WS RI
(http://jax-ws.java.net). RI's version is Metro/2.3.1-b419 (branches/2.3.1.x-7937; 2014-08-04T08:11:03+0000)
JAXWS-RI/2.2.10-b140803.1500 JAXWS-API/2.2.11 JAXB-RI/2.2.10-b140802.1033 JAXB-API/2.2.12-b140109.1041
svn-revision#unknown. --><definitions
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:wsp="http://www.w3.org/ns/ws-policy" xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://cs548.stevens.edu/clinic/service/web/soap" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://cs548.stevens.edu/clinic/service/web/soap"
name="PatientWebService">
<types>
<xsd:schema>
<xsd:import namespace="http://cs548.stevens.edu/clinic/service/web/soap"
schemaLocation="http://54.208.3.81:8080/ClinicSoapWebService-1.0.0/PatientWebService?xsd=1"/>
</xsd:schema>
<xsd:schema>
<xsd:import namespace="http://cs548.stevens.edu/clinic/dto"
schemaLocation="http://54.208.3.81:8080/ClinicSoapWebService-1.0.0/PatientWebService?xsd=2"/>
</xsd:schema>
</types>
<message name="getPatient">
<part name="parameters" element="tns:getPatient"/>
</message>
<message name="getPatientResponse">
<part name="parameters" element="tns:getPatientResponse"/>
</message>
<message name="PatientServiceExn">
<part name="fault" element="tns:PatientServiceExn"/>
</message>
<message name="siteInfo">
<part name="parameters" element="tns:siteInfo"/>
</message>
```

```
<message name="siteInfoResponse">
<part name="parameters" element="tns:siteInfoResponse"/>
</message>
<message name="addPatient">
<part name="parameters" element="tns:addPatient"/>
</message>
<message name="addPatientResponse">
<part name="parameters" element="tns:addPatientResponse"/>
</message>
<message name="getPatientByPatId">
<part name="parameters" element="tns:getPatientByPatId"/>
</message>
<message name="getPatientByPatIdResponse">
<part name="parameters" element="tns:getPatientByPatIdResponse"/>
</message>
<message name="patientGetTreatment">
<part name="parameters" element="tns:patientGetTreatment"/>
</message>
<message name="patientGetTreatmentResponse">
<part name="parameters" element="tns:patientGetTreatmentResponse"/>
</message>
<message name="PatientNotFoundExn">
<part name="fault" element="tns:PatientNotFoundExn"/>
</message>
<message name="TreatmentNotFoundExn">
<part name="fault" element="tns:TreatmentNotFoundExn"/>
</message>
<portType name="IPatientWebPort">
<operation name="getPatient">
<input wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IPatientWebPort/getPatientRequest"
message="tns:getPatient"/>
<output wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IPatientWebPort/getPatientResponse"
message="tns:getPatientResponse"/>
<fault message="tns:PatientServiceExn" name="PatientServiceExn"
wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IPatientWebPort/getPatient/Fault/PatientServiceExn"/>
```

```
<fault message="tns:PatientServiceExn" name="PatientServiceExn"
wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IPatientWebPort/getPatient/Fault/PatientServiceExn"/>
</operation>
<operation name="siteInfo">
<input wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IPatientWebPort/siteInfoRequest"
message="tns:siteInfo"/>
<output wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IPatientWebPort/siteInfoResponse"
message="tns:siteInfoResponse"/>
</operation>
<operation name="addPatient">
<input wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IPatientWebPort/addPatientRequest"
message="tns:addPatient"/>
<output wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IPatientWebPort/addPatientResponse"
message="tns:addPatientResponse"/>
<fault message="tns:PatientServiceExn" name="PatientServiceExn"
wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IPatientWebPort/addPatient/Fault/PatientServiceExn"/>
</operation>
<operation name="getPatientByPatId">
<input wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IPatientWebPort/getPatientByPatIdRequest"
message="tns:getPatientByPatId"/>
<output wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IPatientWebPort/getPatientByPatIdResponse"
message="tns:getPatientByPatIdResponse"/>
<fault message="tns:PatientServiceExn" name="PatientServiceExn"
wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IPatientWebPort/getPatientByPatId/Fault/PatientServiceExn"/>
</operation>
<operation name="patientGetTreatment">
<input wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IPatientWebPort/patientGetTreatmentRequest"
message="tns:patientGetTreatment"/>
<output wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IPatientWebPort/patientGetTreatmentResponse"
message="tns:patientGetTreatmentResponse"/>
<fault message="tns:PatientNotFoundExn" name="PatientNotFoundExn"
wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IPatientWebPort/patientGetTreatment/Fault/PatientNotFoundExn"/
>
<fault message="tns:TreatmentNotFoundExn" name="TreatmentNotFoundExn"
wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IPatientWebPort/patientGetTreatment/Fault/TreatmentNotFoundExn
```

```
message="tns:patientGetTreatmentResponse"/>
<fault message="tns:PatientNotFoundExn" name="PatientNotFoundExn"
wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IPatientWebPort/patientGetTreatment/Fault/PatientNotFoundExn"/
>
<fault message="tns:TreatmentNotFoundExn" name="TreatmentNotFoundExn"
wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IPatientWebPort/patientGetTreatment/Fault/TreatmentNotFoundExn
"/>
<fault message="tns:PatientServiceExn" name="PatientServiceExn"
wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IPatientWebPort/patientGetTreatment/Fault/PatientServiceExn"/>
</operation>
</portType>
<binding name="PatientWebPortBinding" type="tns:IPatientWebPort">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
<operation name="getPatient">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
<fault name="PatientServiceExn">
<soap:fault name="PatientServiceExn" use="literal"/>
</fault>
</operation>
<operation name="siteInfo">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="addPatient">
```

```
<operation name="addPatient">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
<fault name="PatientServiceExn">
<soap:fault name="PatientServiceExn" use="literal"/>
</fault>
</operation>
<operation name="getPatientByPatId">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
<fault name="PatientServiceExn">
<soap:fault name="PatientServiceExn" use="literal"/>
</fault>
</operation>
<operation name="patientGetTreatment">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
<fault name="PatientNotFoundExn">
<soap:fault name="PatientNotFoundExn" use="literal"/>
</fault>
<fault name="TreatmentNotFoundExn">
```

```
<fault name="TreatmentNotFoundExn">
<soap:fault name="TreatmentNotFoundExn" use="literal"/>
</fault>
<fault name="PatientServiceExn">
<soap:fault name="PatientServiceExn" use="literal"/>
</fault>
</operation>
</binding>
<service name="PatientWebService">
<port name="PatientWebPort" binding="tns:PatientWebPortBinding">
<soap:address location="http://54.208.3.81:8080/ClinicSoapWebService-1.0.0/PatientWebService"/>
</port>
</service>
</definitions>
```

ProviderWebService.xml

```
<?xml version='1.0' encoding='UTF-8'?><!-- Published by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.1-b419
(branches/2.3.1.x-7937; 2014-08-04T08:11:03+0000) JAXWS-RI/2.2.10-b140803.1500 JAXWS-API/2.2.11 JAXB-RI/2.2.10-b140802.1033
JAXB-API/2.2.12-b140109.1041 svn-revision#unknown. --><!-- Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.1-b419
(branches/2.3.1.x-7937; 2014-08-04T08:11:03+0000) JAXWS-RI/2.2.10-b140803.1500 JAXWS-API/2.2.11 JAXB-RI/2.2.10-b140802.1033
JAXB-API/2.2.12-b140109.1041 svn-revision#unknown. --><definitions
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://cs548.stevens.edu/clinic/service/web/soap"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://cs548.stevens.edu/clinic/service/web/soap" name="ProviderWebService">
<types>
<xsd:schema>
<xsd:import namespace="http://cs548.stevens.edu/clinic/service/web/soap"
schemaLocation="http://54.208.3.81:8080/ClinicSoapWebService-1.0.0/ProviderWebService?xsd=1"/>
</xsd:schema>
<xsd:schema>
<xsd:import namespace="http://cs548.stevens.edu/clinic/dto"
schemaLocation="http://54.208.3.81:8080/ClinicSoapWebService-1.0.0/ProviderWebService?xsd=2"/>
</xsd:schema>
</types>
<message name="addProvider">
<part name="parameters" element="tns:addProvider"/>
</message>
<message name="addProviderResponse">
<part name="parameters" element="tns:addProviderResponse"/>
</message>
<message name="ProviderServiceExn">
<part name="fault" element="tns:ProviderServiceExn"/>
</message>
<message name="getProviderByPatId">
<part name="parameters" element="tns:getProviderByPatId"/>
</message>
<message name="getProviderByPatIdResponse">
<part name="parameters" element="tns:getProviderByPatIdResponse"/>
</message>
<message name="addDrugTreatment">
<part name="parameters" element="tns:addDrugTreatment"/>
</message>
```



```
<message name="addDrugTreatment">
<part name="parameters" element="tns:addDrugTreatment"/>
</message>
<message name="addDrugTreatmentResponse">
<part name="parameters" element="tns:addDrugTreatmentResponse"/>
</message>
<message name="addRadiology">
<part name="parameters" element="tns:addRadiology"/>
</message>
<message name="addRadiologyResponse">
<part name="parameters" element="tns:addRadiologyResponse"/>
</message>
<message name="addSurgery">
<part name="parameters" element="tns:addSurgery"/>
</message>
<message name="addSurgeryResponse">
<part name="parameters" element="tns:addSurgeryResponse"/>
</message>
<message name="siteInfo">
<part name="parameters" element="tns:siteInfo"/>
</message>
<message name="siteInfoResponse">
<part name="parameters" element="tns:siteInfoResponse"/>
</message>
<message name="getProvider">
<part name="parameters" element="tns:getProvider"/>
</message>
<message name="getProviderResponse">
<part name="parameters" element="tns:getProviderResponse"/>
</message>
```

```
<portType name="IProviderWebPort">
<operation name="addProvider">
<input wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IProviderWebPort/addProviderRequest" message="tns:addProvider"/>
<output wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IProviderWebPort/addProviderResponse" message="tns:addProviderResponse"/>
<fault message="tns:ProviderServiceExn" name="ProviderServiceExn"
wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IProviderWebPort/addProvider/Fault/ProviderServiceExn"/>
</operation>
<operation name="getProviderByPatId">
<input wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IProviderWebPort/getProviderByPatIdRequest"
message="tns:getProviderByPatId"/>
<output wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IProviderWebPort/getProviderByPatIdResponse"
message="tns:getProviderByPatIdResponse"/>
<fault message="tns:ProviderServiceExn" name="ProviderServiceExn"
wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IProviderWebPort/getProviderByPatId/Fault/ProviderServiceExn"/>
</operation>
<operation name="addDrugTreatment">
<input wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IProviderWebPort/addDrugTreatmentRequest" message="tns:addDrugTreatment"/>
<output wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IProviderWebPort/addDrugTreatmentResponse"
message="tns:addDrugTreatmentResponse"/>
<fault message="tns:ProviderServiceExn" name="ProviderServiceExn"
wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IProviderWebPort/addDrugTreatment/Fault/ProviderServiceExn"/>
</operation>
<operation name="addRadiology">
<input wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IProviderWebPort/addRadiologyRequest" message="tns:addRadiology"/>
<output wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IProviderWebPort/addRadiologyResponse"
message="tns:addRadiologyResponse"/>
<fault message="tns:ProviderServiceExn" name="ProviderServiceExn"
wsam:Action="http://cs548.stevens.edu/clinic/service/web/soap/IProviderWebPort/addRadiology/Fault/ProviderServiceExn"/>
</operation>
```

```
</portType>
<binding name="ProviderWebPortBinding" type="tns:IProviderWebPort">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
<operation name="addProvider">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
<fault name="ProviderServiceExn">
<soap:fault name="ProviderServiceExn" use="literal"/>
</fault>
</operation>
<operation name="getProviderByPatId">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
<fault name="ProviderServiceExn">
<soap:fault name="ProviderServiceExn" use="literal"/>
</fault>
</operation>
<operation name="addDrugTreatment">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
```

```
<fault name="ProviderServiceExn">
<soap:fault name="ProviderServiceExn" use="literal"/>
</fault>
</operation>
<operation name="addRadiology">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
<fault name="ProviderServiceExn">
<soap:fault name="ProviderServiceExn" use="literal"/>
</fault>
</operation>
<operation name="addSurgery">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
<fault name="ProviderServiceExn">
<soap:fault name="ProviderServiceExn" use="literal"/>
</fault>
</operation>
<operation name="siteInfo">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
```

```
<operation name="getProvider">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
<fault name="ProviderServiceExn">
<soap:fault name="ProviderServiceExn" use="literal"/>
</fault>
</operation>
</binding>
<service name="ProviderWebService">
<port name="ProviderWebPort" binding="tns:ProviderWebPortBinding">
<soap:address location="http://54.208.3.81:8080/ClinicSoapWebService-1.0.0/ProviderWebService"/>
</port>
</service>
</definitions>
```