CS548 HW 5: Domain Driven Architecture – Clinic Domain

DOMAIN LOGIC

1. Add a new patient to the clinic

   edu.cs548.PatientDAO

   - `public Long addPatient(String name, Date dob, int age)`

2. Retrieve a patient aggregate from the system, given the **primary key** for the patient.

   edu.cs548.PatientDAO

   -> `public Patient getPatientByDbId(long id) throws PatientExn`

3. Delete a patient aggregate from the system, given the **primary key** for the patient.

   edu.cs548.PatientDAO

   -> `public void deletePatientByPrimaryKey(long id) throws PatientExn`

   Since the JPA annotation specify that the relationship between treatment and patient is

   ```
   @OneToMany(cascade = CascadeType.REMOVE,mappedBy = "patient"),
   ```

   the treatments aggregate to this patient will be removed.

4. Retrieve a patient aggregate from the system, given the **patient identifier** for the patient.

   edu.cs548.PatientDAO

   -> `public Patient getPatientByPatientId(long pid) throws PatientExn`

5. Given a patient's name and birthday, retrieve a list of patient aggregates with that identifying information from the system.

   edu.cs548.PatientDAO

   -> `public List<Patient> getPatientByNameDob(String name, Date dob)`

6. Add a provider to a clinic

   edu.cs548.ProviderDAO

   Overload methods to add a provider:

   ```
   public void addProvider(Provider provider) throws ProviderExn
   ```

   or

   ```
   public void addProvider(long npi, String name) throws ProviderExn;
   ```

7. Return all providers with a given name.
   1.
   ```
   Edu.cs548.ProviderDAO
   public List<Provider> getProviderByName(String name) throws ProviderExn
   ```
   2.
   ```
   Edu.cs548.ProviderDAO
   public Provider getProviderByNpi(long npi) throws ProviderExn
   ```
8. Delete the provider entity
   ```
   Edu.cs548.ProviderDAO
   public void deleteProvider(Provider provider) throws ProviderExn
   ```
   however, the treatment entities associated with the provider entity is not deleted since:
   ```
   @OneToMany(mappedBy = "provider") //this provider is the field in Treatment
   @OrderBy
   private List<Treatment> treatments;
   ```
9. Add a treatment for a patient.
   Edu.cs548.Provider
   ```
   public Long addTreatment (Treatment t)
   ```

10. A. Return a List of treatment identifiers with these operations:
    ```
    Edu.cs548.Patient
    public List<Long> getTreatmentIds()
    ```
    B. Visit a particular treatment.
    ```
    Edu.cs548.Patient
    public void visitTreatment(long tid, ITreatmentVisitor visitor) throws
    TreatmentExn
    ```
11. edu.cs548.Provider
    a. Return a list of treatments (id) associated with this provider
       ```
       public List<Long> getTreatmentIds()
       ```
    b. Return a list of treatments (id) administered to a patient supervised by that provider.
       ```
       public List<Long> getTreatmentIds(Patient patient)
       ```

    c. Visit a treatment by the treatment id
       ```
       public void visitTreatment(long tid, ITreatmentVisitor visitor) throws
       TreatmentExn
       ```

    d. Delete a treatment
       ```
       public void deleteTreatment(long tid) throws TreatmentExn
       ```

Initializing DAOs

The CliniGateway object can be used to initialize a ProviderDAO or a PatientDAO. However, EntityManager is thread unsafe, so, by now a thread cannot hold both a ProviderDAO and a PatientDAO at the same time for safe reason. So, this protocol might need to be updated may be in the following assignment according to the requirements. But the EntityManagerFactory is thread safe though.

```
public class ClinicGateway implements IClinicGateway{

    @Override
    public IPatientFactory getPatientFactory() {
        return new PatientFactory();
    }Ω
    private EntityManagerFactory emf;
    @Override
    public IPatientDAO getPatientDAO() {
        EntityManager em = emf.createEntityManager();
        return new PatientDAO(em);
    }

    public ClinicGateway(){
        //bootstrap
        emf = Persistence.createEntityManagerFactory("ClinicDomain");
        //the name can be found in persistence.xml
    }

    @Override
    public IProviderDAO getProviderDAO() {
        EntityManager em = emf.createEntityManager();
        return new ProviderDAO(em);
    }
}
```

TreatmentDAO aggregated in PatientDAO will be initialized in the constructor of PatientDAO:

```
    public PatientDAO(EntityManager em){
        this.em = em;
        this.treatmentDAO = new TreatmentDAO(em);
    }
```

TreatmentDAO aggregated in ProviderDAO will be initialized in the constructor of ProviderDAO:

```
    public ProviderDAO(EntityManager em){
        this.em = em;
        this.treatmentDAO = new TreatmentDAO(em);
    }
```