Comet Patient Monitor

Design Model

Team Members:
Patrick Haring
Christian Bürgi

Client:
Prof. Dr. Olivier Biberstein

Revision: 123

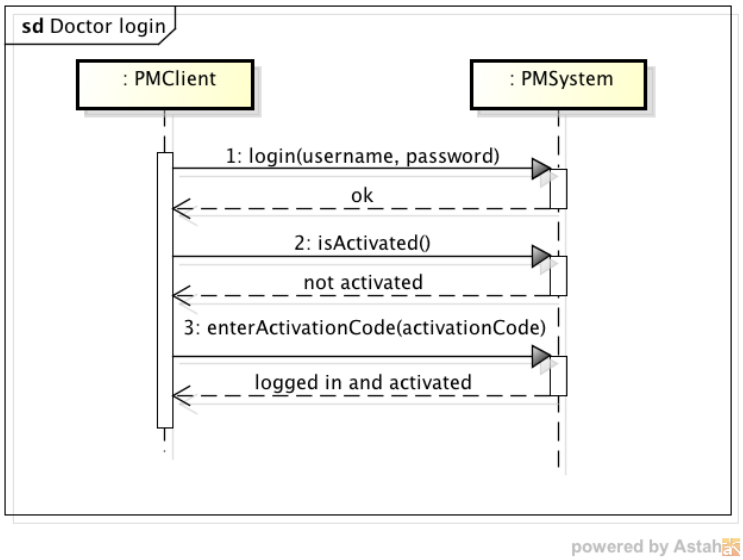
May 13, 2012

https://svn.bfh.ch/repos/projects/patmon1/trunk/doc/src/design_model.tex?p=123

Contents

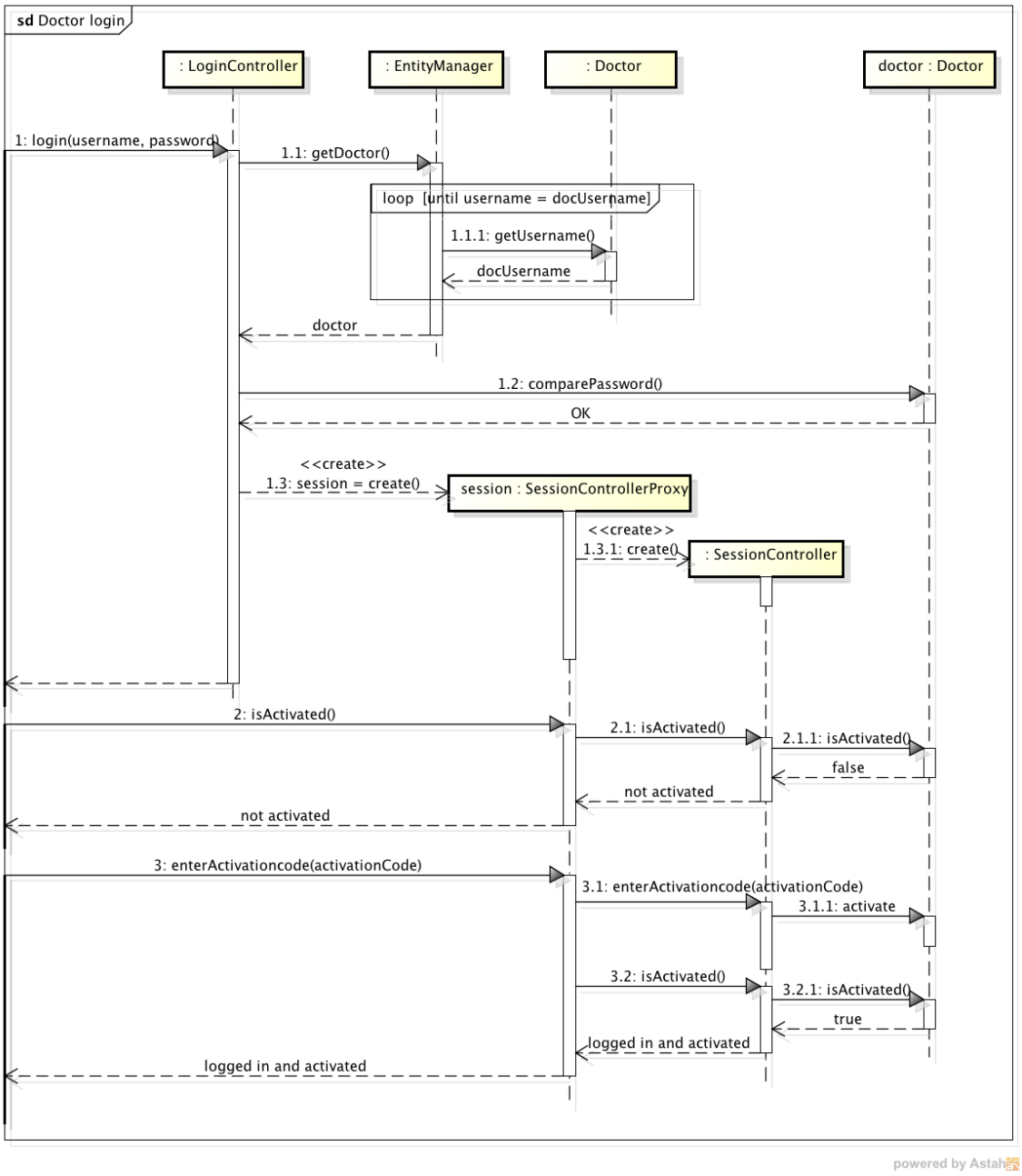
1 Doctor login

1.1 System sequence diagram



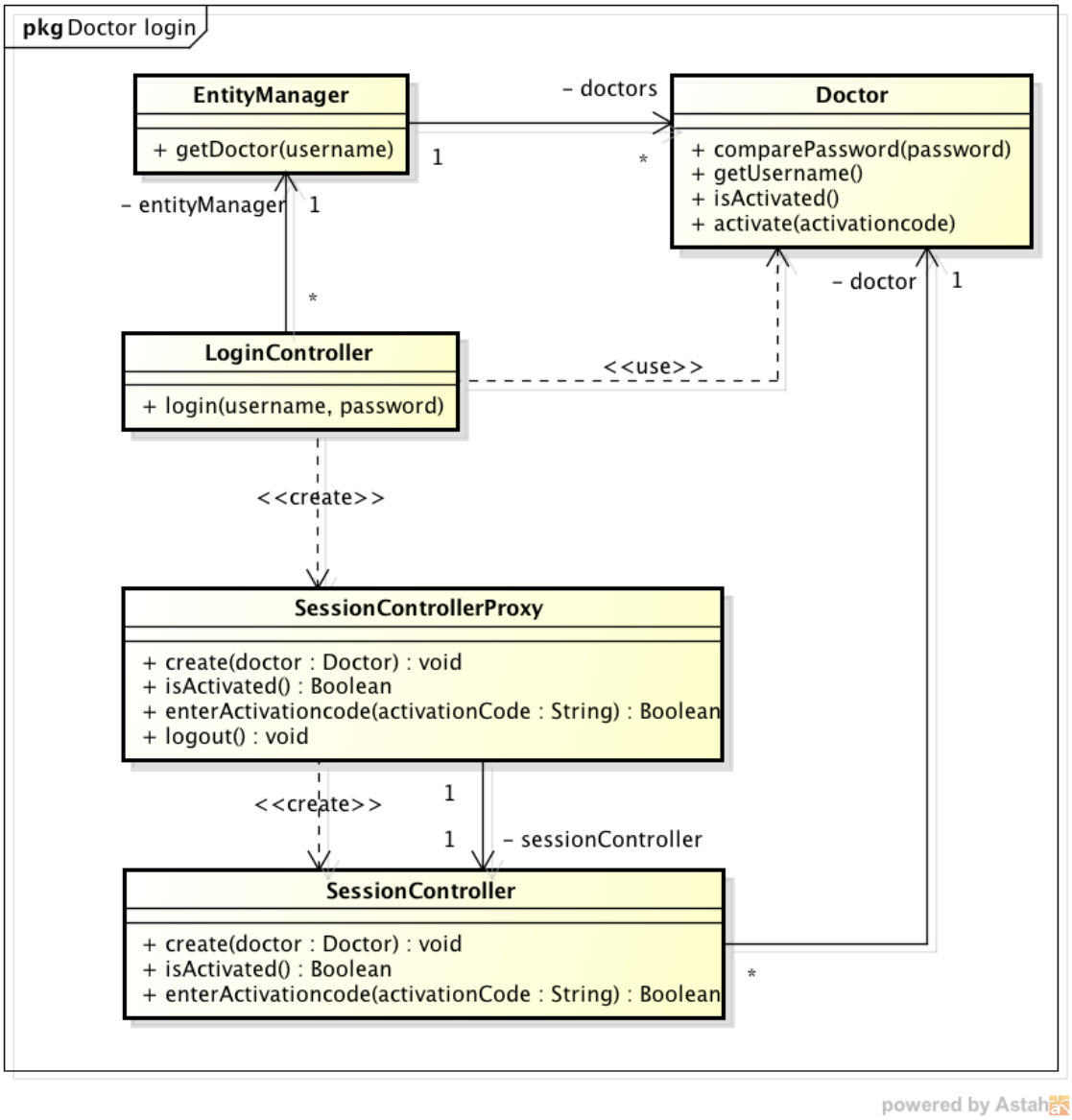
This diagram describes the communication between the client and the server during a login of a doctor. The client sends a login message containing the login information of the doctor. The server confirms and then the client follows protocol checking if the account doctor is activated and proceeds with activating the doctor.

1.2 Sequence diagram



There is a login controller for all doctor login messages. It looks up the doctor using the entity manager and lets the doctor (if found) check the credentials. If this succeeds, a session controller is created, which is responsible for the current session of the client. For security reasons, there is also a proxy for the controller which is sent to the client instead of the controller itself. The client then checks activation and activates the login via the session controller.

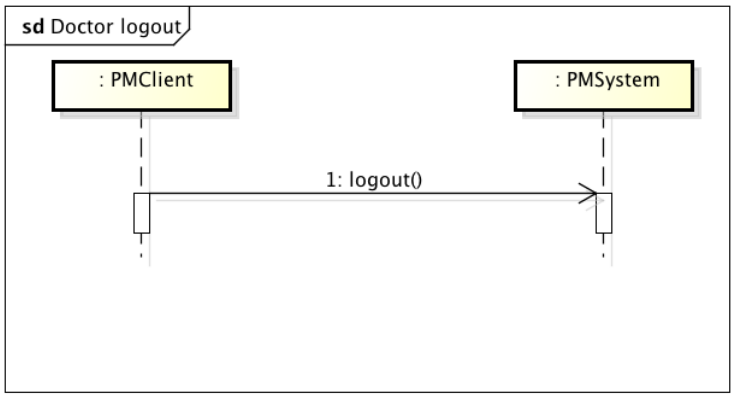
1.3 Design class diagram



Thesession controller gets access to the doctor using the entity manager. After creating the session controller, the doctor is referenced by the session controller.

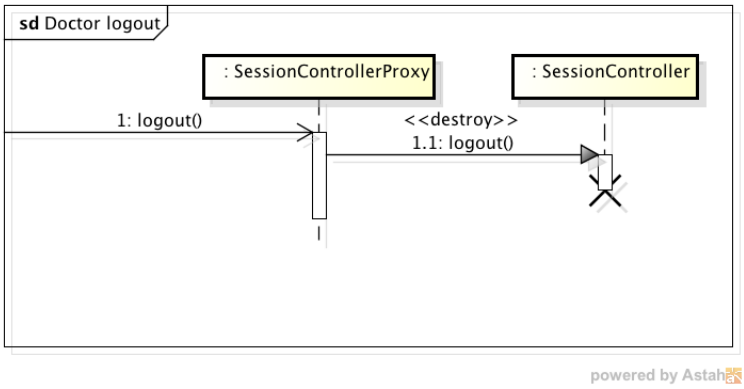
2 Logout

2.1 System sequence diagram



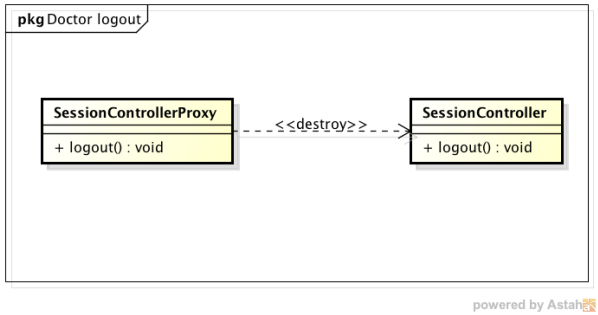
A logout is done simply by sending a logout message from the client to the server.

2.2 Sequence diagram



A logout on the session controller proxy destroys the session controller. Because the session controller only lives on the server, there is no way for the client to sent messages to the server, because he only has a reference to the proxy.

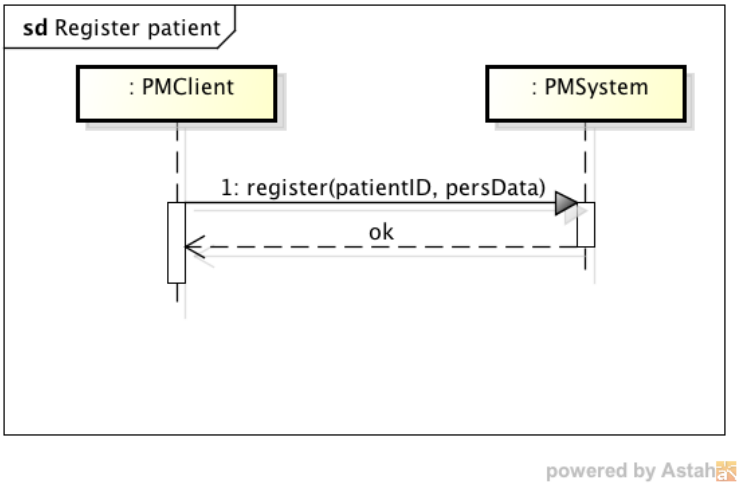
2.3 Design class diagram



During logout the proxy destroys the session controller.

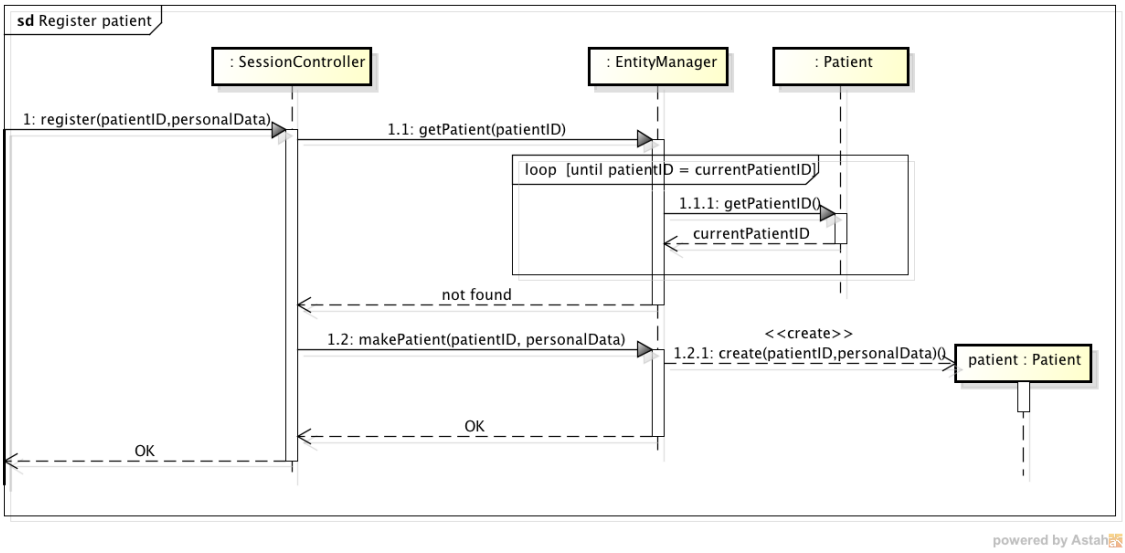
3 Register patient

3.1 System sequence diagram



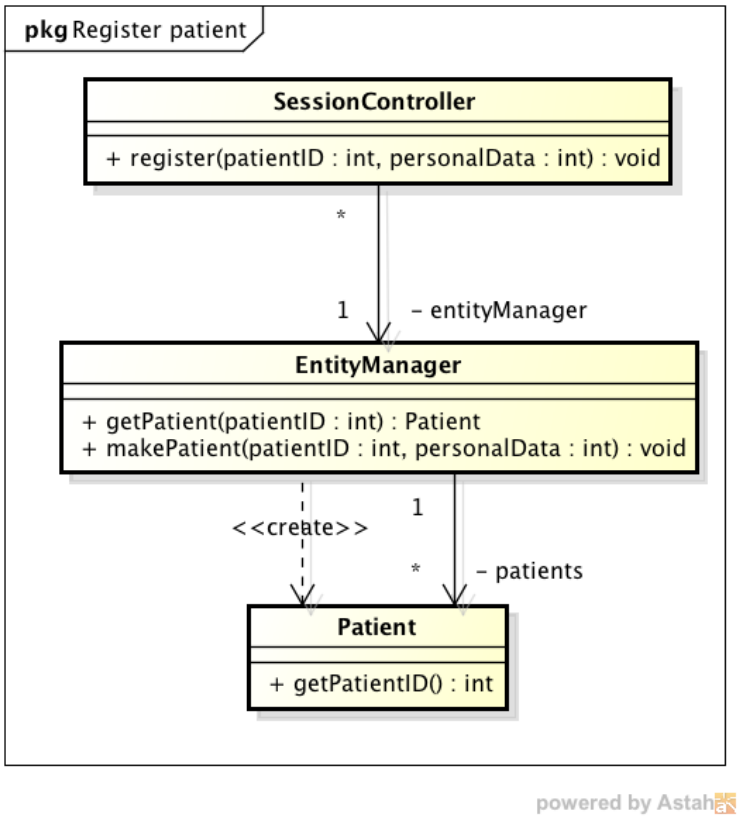
The diagram shows the communication between the client and the patient monitoring system. The client sends the patients SSN as the patientID and adds the personal data. If the registration went well the System will send a confirmation signal.

3.2 Sequence diagram



The SessionController looks up if the patients id (his SSN) was already registered in the system and if not e will create a new patient by calling the makePatient method of the EntitiyManager.

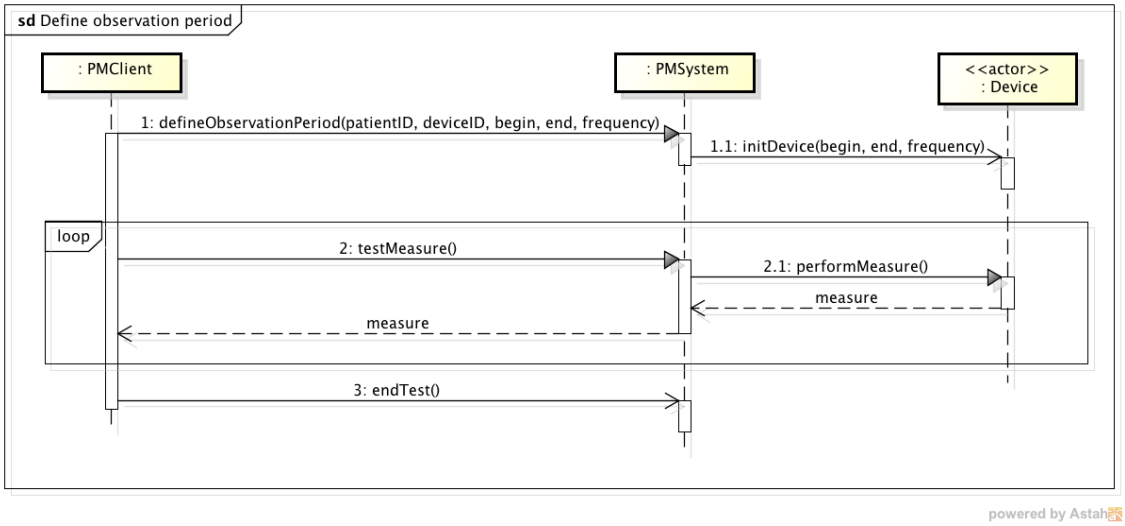
3.3 Design class diagram



The SessionController creates a new patient via the the makePatient method of the EntityManager.

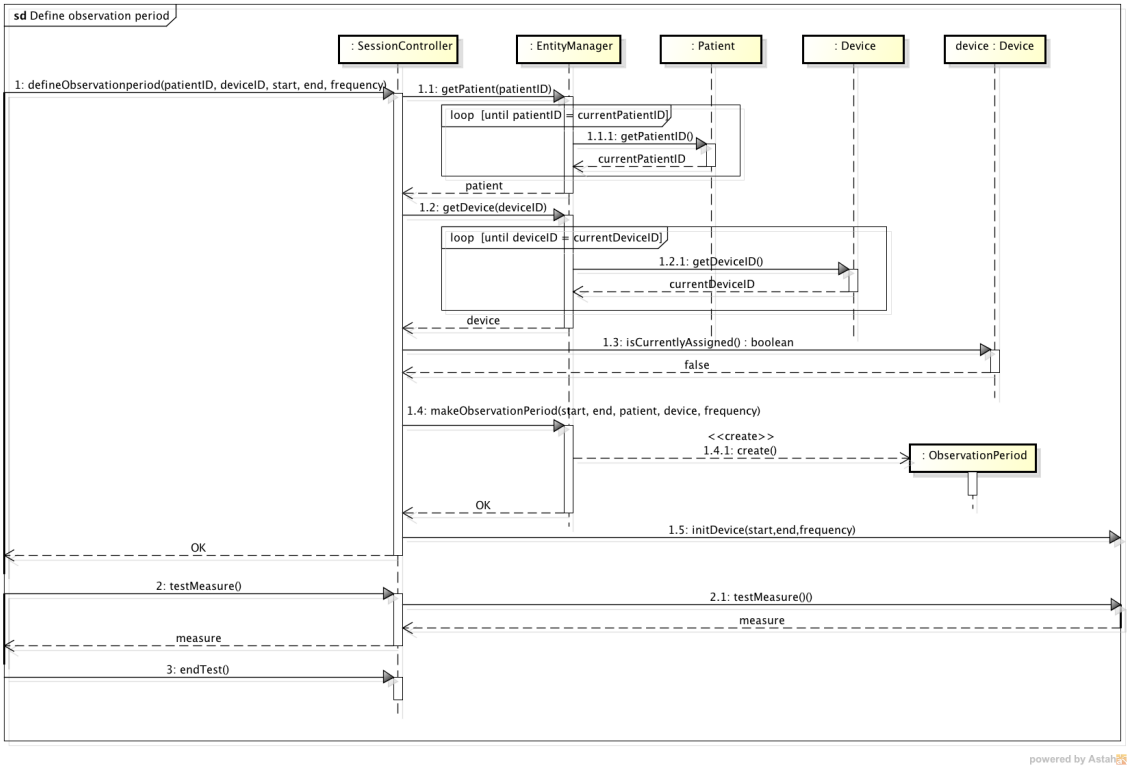
4 Define observation period

4.1 System sequence diagram



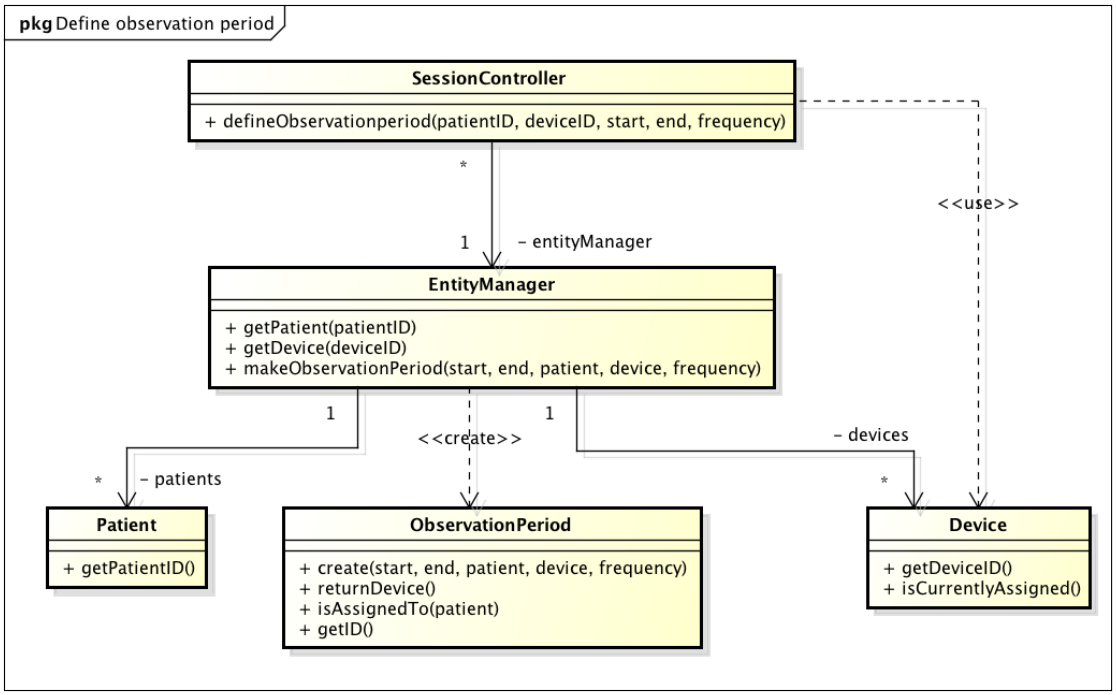
For defining an observation period, the client sends a message to the server. After this, the device is initialized with the data of the period. Then the client performs test-measures by sending a message to the server. The system itself sends a performMeasure-message to the device which performs a measure and sends it to the system. This procedure can be repeated multiple times and is ended with an endTest-message.

4.2 Sequence diagram



For defining an observation period, the session controller gets the patient and device from the entity manager. Then it creates an observation period using the entity manager.

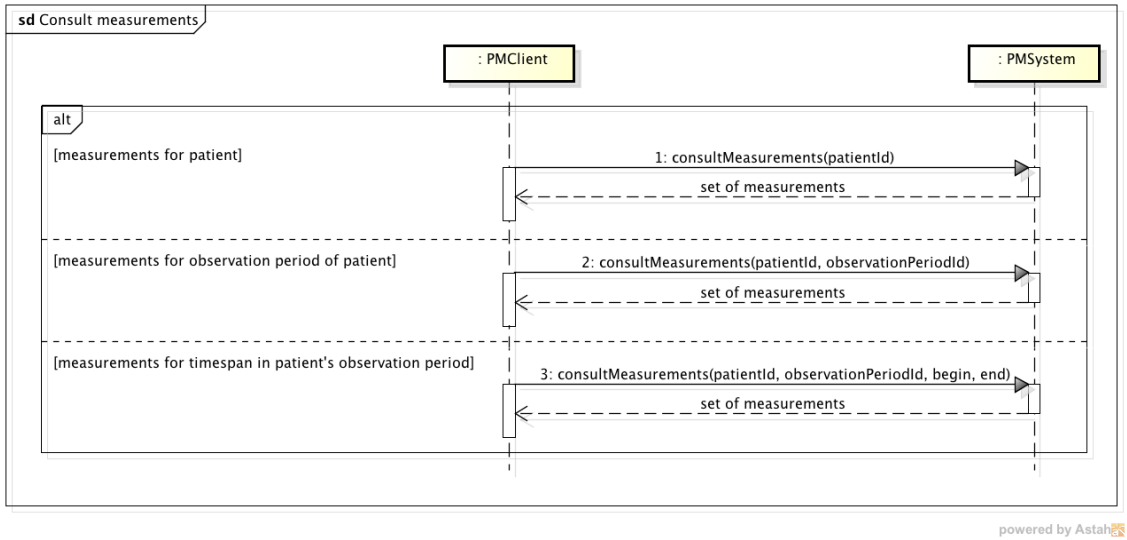
4.3 Design class diagram



The session controller gets patient and device from the entity manager and lets it create an observation period.

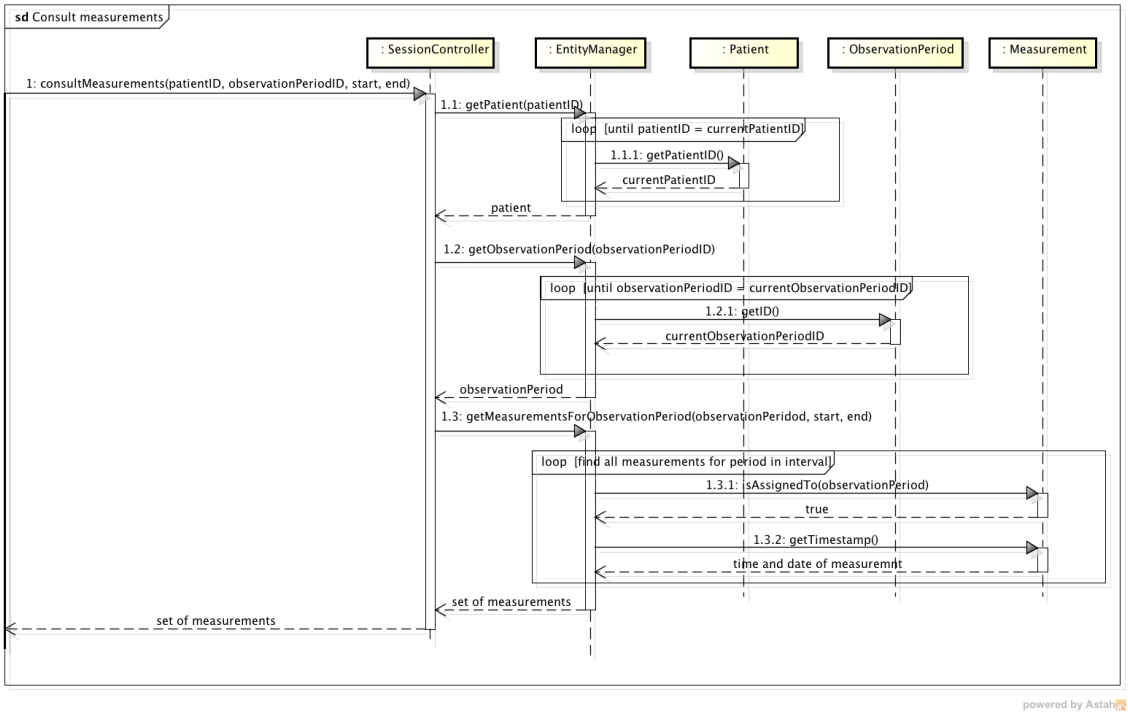
5 Consult measurements

5.1 System sequence diagram



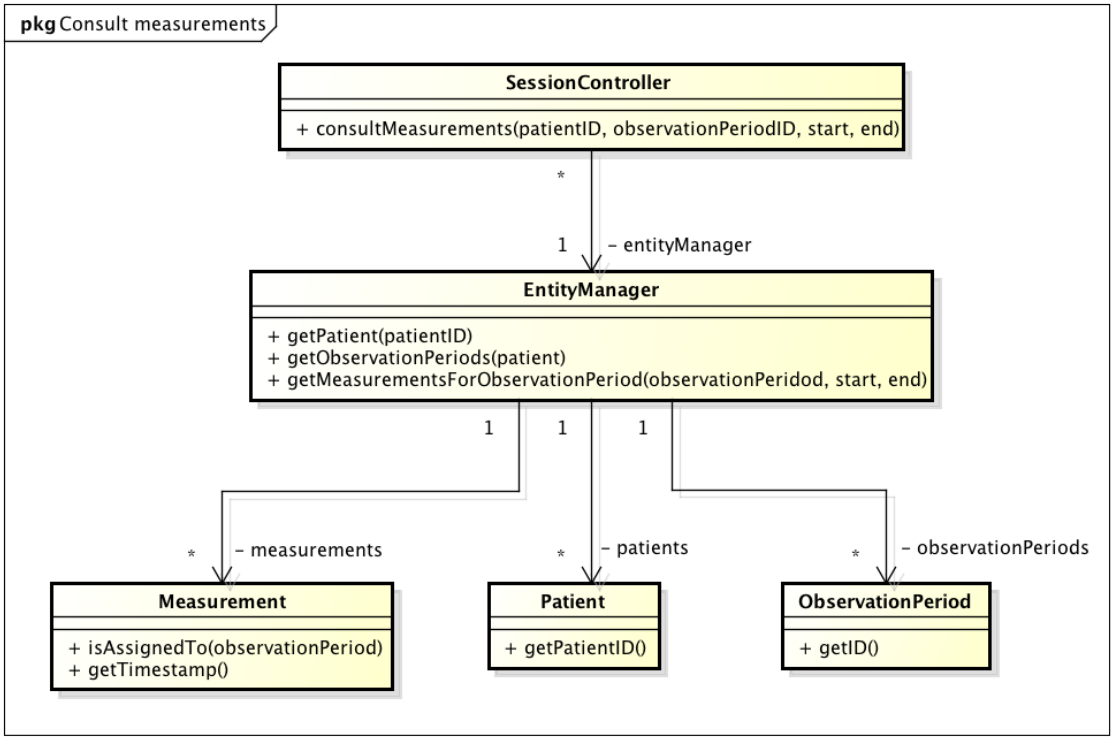
There are three different ways of querying data which apply to this use case. We can implement the most difficult one and implement the others by setting the parameters in the right manner.

5.2 Sequence diagram



The SessionController first looks up the Patient, ObservationPeriod by querying the EntityManager and will then query the EntityManager for all Measurements concerning the given timespan in the ObservationPeriod.

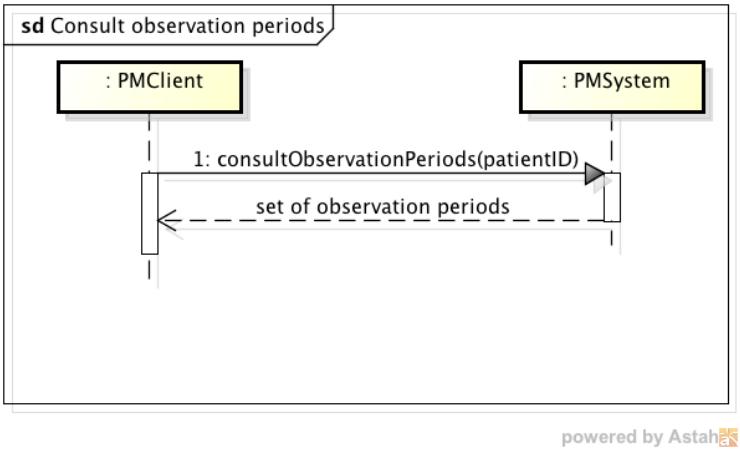
5.3 Design class diagram



The SessionController queries the EntityManager for the desired measurements.

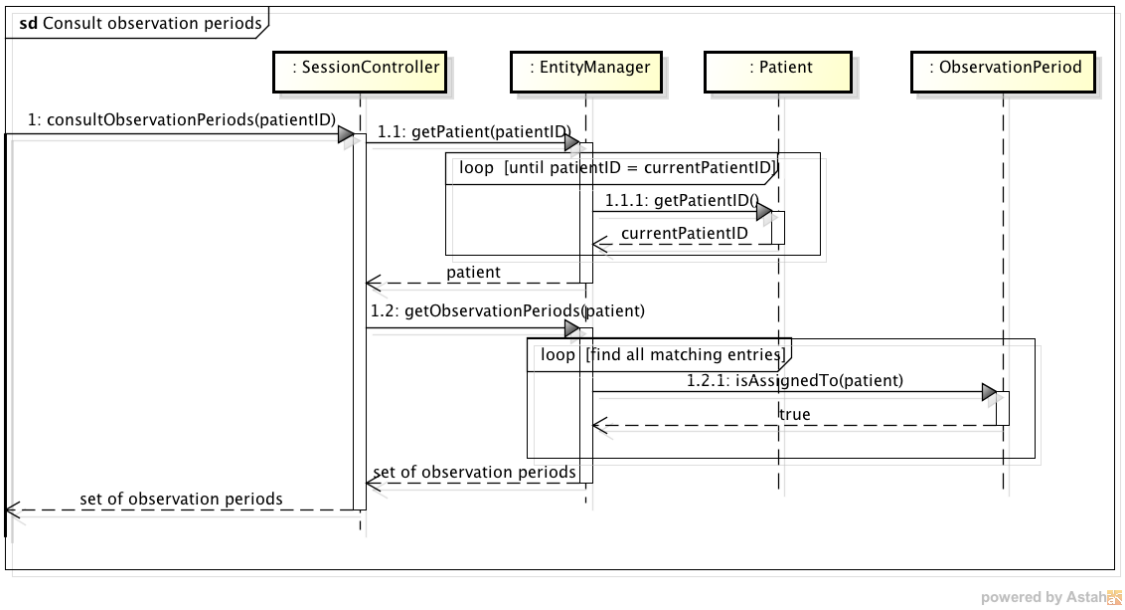
6 Consult observation periods

6.1 System sequence diagram



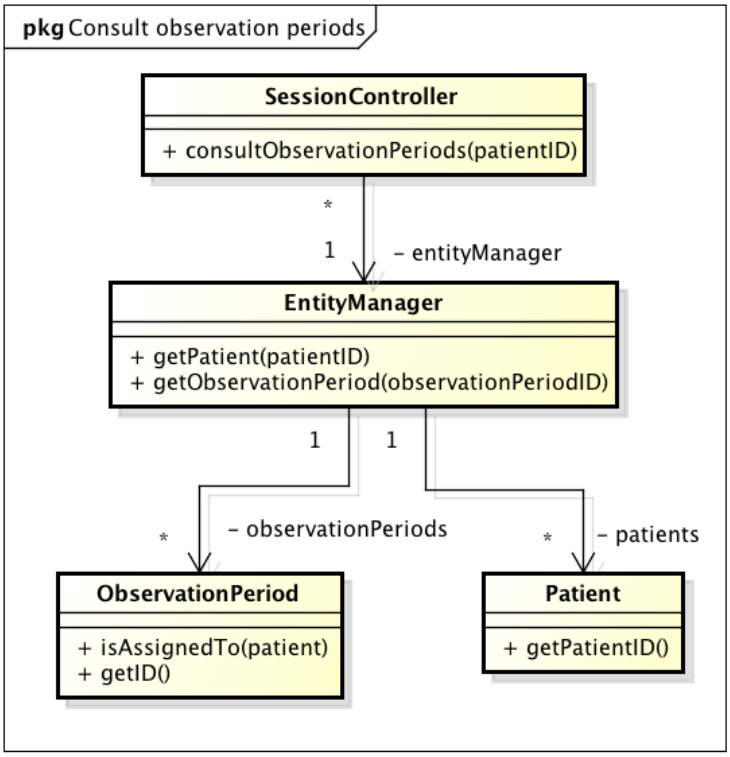
This diagram shows the communication between the client and the patient monitoring system. The patient invokes the method consultObservationPeriods with a patient's id and he gets a set of all observation periods which are assigned to him.

6.2 Sequence diagram



The SessionController looks up the Patient and then he will query the EntityManager for Observation Periods which are assigned to this Patient.

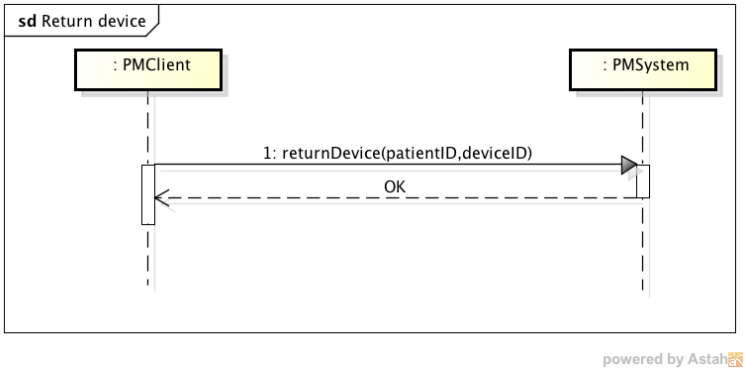
6.3 Design class diagram



The SessionController accesses the Patient and the ObservationPeriods via the EntityManager.

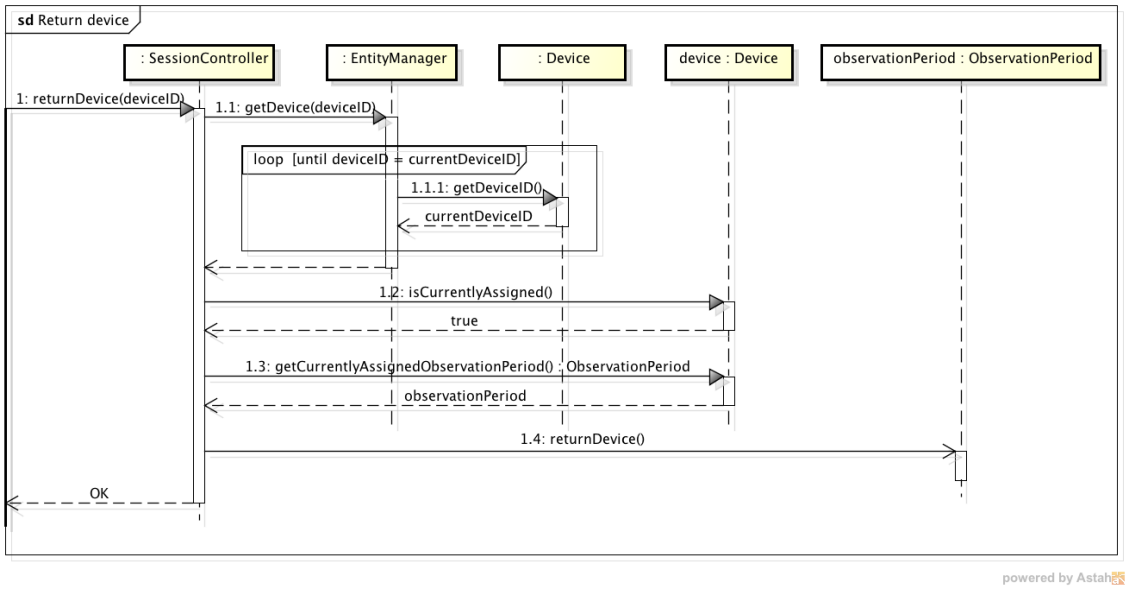
7 Return device

7.1 System sequence diagram



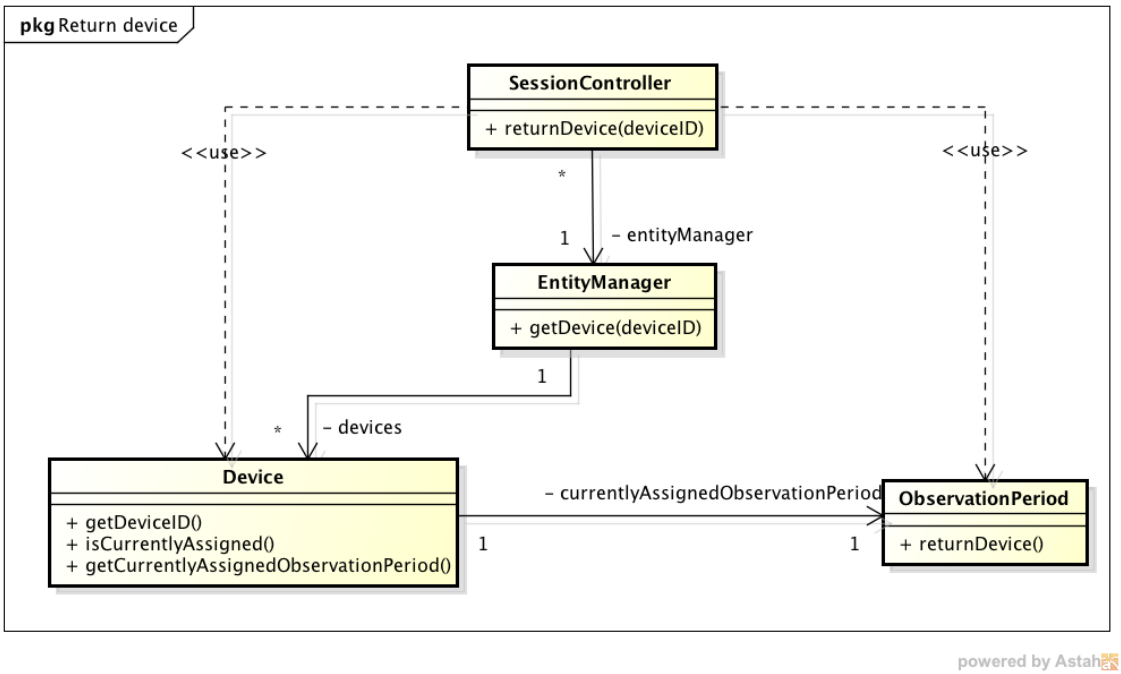
For returning a device the client sends a message to the system which is acknowledged.

7.2 Sequence diagram



For returning a device the session controller get the device from the entity manager. He checks if the device is actually assigned and gets the observation period from it. The returning of the device is then done on the observation period object.

7.3 Design class diagram



The session controller uses the entity manager to get the device, from where it gets the observation period.