

FILS

Course: Software Design Techniques

JMS Homework

We are developing an integrated system for emergency healthcare. It should perform various scenarios like the following:

Scenario: Suppose somebody has a serious health problem and is taken by an emergency unit (Salvare). While the emergency unit is on its way to the hospital with the patient, the hospital must be notified of the situation using the EmergencyUnit application that is installed in the emergency unit. Also FamilyDoctor, the application of the patient's GP, should be notified. When the patient enters the hospital the doctors may need additional data so they send a request for the medical history of the patient to PatientRegistry, a service-oriented (not web service-oriented!) application that accesses a repository (database) containing medical information about all the citizens.

Let consider the following interaction of components aimed to carry out the above scenario :

1. EmergencyUnit sends a message (Message1) to a topic (jms/NewPatientTopic). The message should contain the name of the patient and his/her condition.
2. HospitalGateway is a MessageDrivenBean that listens to jms/NewPatientTopic. When a message (Message1) is received on jms/NewPatientTopic, it sends another message (Message2) to a queue jms/InquireQueue containing the name of the patient. Since it's a request-reply scenario the return address must also be set in the message. It listens for the reply to the inquiry on the return address set in the message, jms/ResponseQueue.
3. FamilyDoctor is a MessageDrivenBean that listens to jms/NewPatientTopic. Since it is not always online it must be a durable subscriber.
4. PatientRegistry is a MessageDrivenBean that listens to jms/InquireQueue. It receives a message (Message2), looks up in the database and responds with a message (Message3) to the return address specified in the received message. The message should contain the medical history of the patient and the correlation id.

Hint. You first have to create the topics and queues in GlassFish (Services->GlassFish->Start->View Admin Console->Resources->JMS)