SYMPTOM MANAGEMENT

System overview

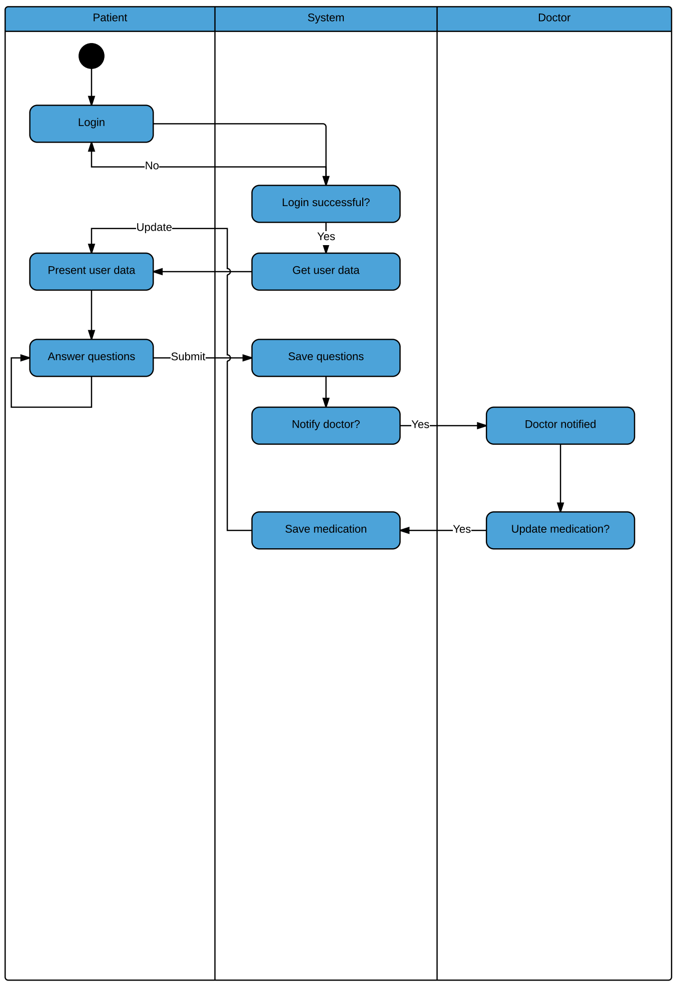
All communication between the app and backend will go through https. Below is a table describing the different services, http-method, who (patient or doctor) will use it and what it does. The services will be secured, so that a patient only can see his/her information and the doctors only his/her patients. All services will be implemented in Java Spring.

Service name	HTTP-method	User	Description
patient/	GET	Doctor	Get a list of a doctors patients
patient/{id}	GET	Patient/doctor	Information about the patient. Name, date of birth, medical record number
patient/search?name={name}	GET	Doctor	Search for a patient by name
patient/{id}/medication	GET	Patient/doctor	Patient pain medication
patient/{id}/medication	PUT	Doctor	Update a patient pain medication
patient/{id}/answers	GET	Doctor	Get patient answers
patient/{id}/answers/{date}	GET	Doctor	Get a patient answers for a specific date
patient/{id}/answers	POST	Patient	Saves patient answers
questions/	GET	Patient	Get all questions the patient is suposed to answer

Login

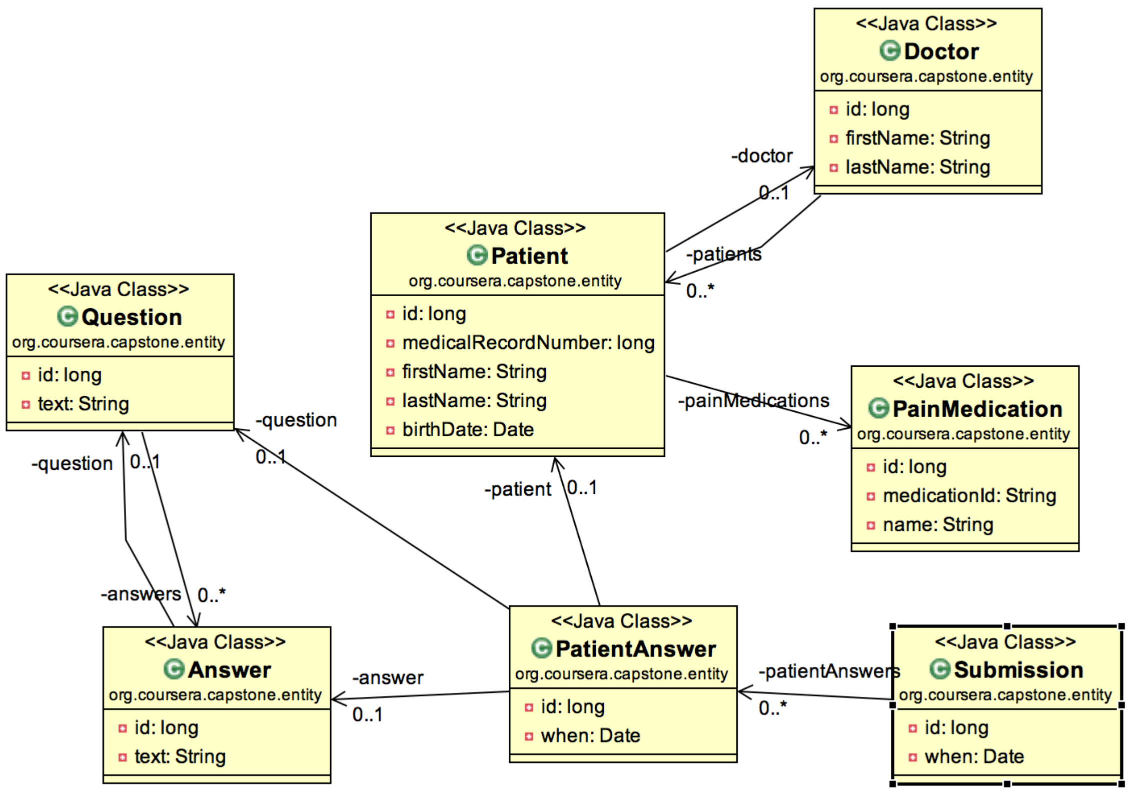
When launching the application a login screen will be shown. The patient or doctor will login with his/her credentials and the app will start. The backend will use OAuth2 so the user will not have to login every time he or she starts the app. After a successful login the patient can configure the reminders and then begin to answer the questions. The doctor will see a list of his/her patients, where it’s possible to search and to go in and see details for a specific patient (see attachment AndroidPatientMockups.pdf and AndroidDoctorMockups.pdf). The patient search will be done server-side. On login the app will either start the doctor or patient UI depending on what the user is.

Basic flow



To the left is the overall patient workflow. When the patient has submitted the question answers, it sends a http(s)-request to the backend that will store them in a SQL-database. If the patient has had either of 12 hours of “severe pain,” 16 hours of "moderate" to "severe pain," or 12 hours of “I can’t eat, the doctor will be notified by a notification to his/her phone. The implementation on how the doctor receives it will be either an Android Service that polls the server every other 10 minutes or so, or by google GCM. When the doctor receives the notification he or she can update the patient pain medication and the patient will receive an updated list of medications (again either by Service or GCM). The doctor always has this choice, but to simplify the flow, it’s added here. The box “Present user data” fetches both patient information and his/her medications.

Data model



Android/UI

See attachments AndroidPatientMockups.pdf and AndroidDoctorMockups.pdf for app UI. The app will be realised with two main activities, one for doctor and one for patient, that each have different fragments that will animate when navigating between them.

Communication with backend will be done in background services, that doesn't run on UI-thread.

The *Reminder* alarm will use Androids AlarmManager and will be a background Service. When the alarm goes off a notification will show up on the patients android device and when clicking on that, the app will launch and the patient answers the questions.