

# SYMPTOM MANAGEMENT

## System overview

All communication between the app and backend goes through https. Below is a table describing the different services, http-method, who (patient or doctor) uses it and what it does. The services are secured, so that a only a patient can perform a check-in, and only a doctor can update patient medications. All services are implemented in Java Spring.

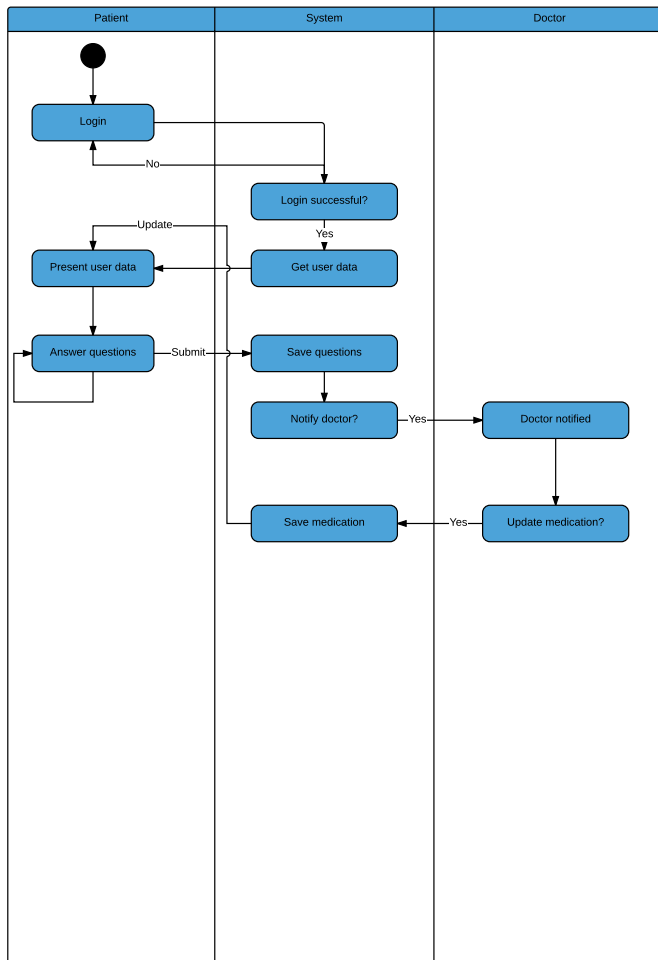
Service name	HTTP-method	User	Description
<b>/oauth/token</b>	POST	Patient/doctor	Log in path
<b>user/</b>	GET	Patient/doctor	Get information about the logged in user
<b>patient/{username}</b>	GET	Patient/doctor	Information about the patient. Name, date of birth, medical record number, doctor, pain medications
<b>patient/search?name={name}</b>	GET	Doctor	Search for a patient by name
<b>medication/</b>	GET	Doctor	Get a list of all available pain medications
<b>medication/{username}</b>	PUT	Doctor	Updates a patient pain medication
<b>doctor/{username}/patient</b>	GET	Doctor	Get a list of doctor patients
<b>checkin/</b>	POST	Patient	Submit a patient check-in
<b>checkin/{username}</b>	GET	Doctor	Get a patient check-ins
<b>checkin/{username}/alerts</b>	GET	Doctor	Get List of alerts for a patient
<b>question/</b>	GET	Patient	Get all questions and answers the patient is supposed to answer

## Login

When launching the application for the first time a login screen is shown. The patient or doctor logs in with his/her credentials and the app starts. On login the app will either start the doctor or patient UI depending on what role the user have. The backend uses OAuth2 so the user doesn't have to login every time he or she starts the

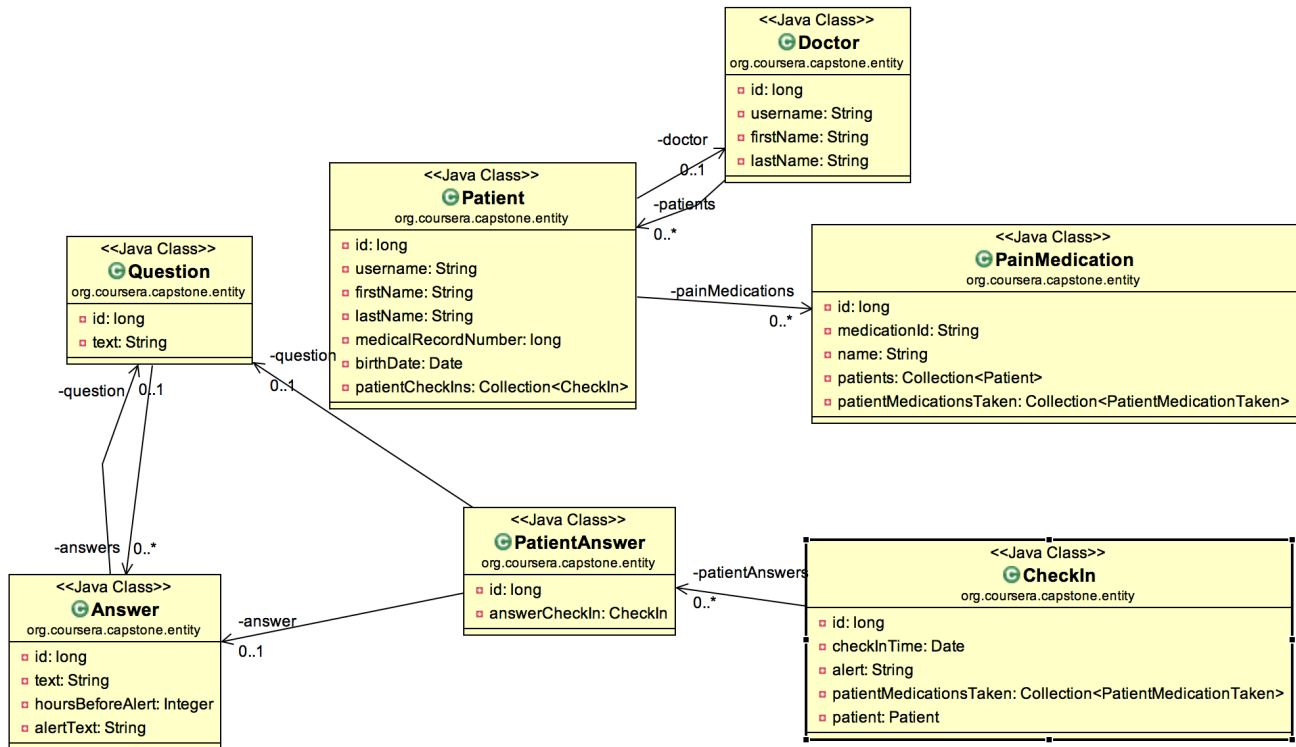
app. After a successful login the patient can configure the reminders and/or begin to answer the questions. The doctor sees a list of his/her patients, where it's possible to search for patients and to go in and see details for a specific patient. In the patient details page, patient alerts, check-ins and medications are shown. The doctor can update a patient medication. The patient search is made on server-side.

### 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 stores 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 doctor receives this via an Android Service that polls the server every half hour. 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. 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 the demo movie for the app UI. The app is realised with two main activities, one for doctor and one for patient, that each have different fragments.

Communication with backend are done in background services (AsyncTask).

The *Reminder* alarm use Androids AlarmManager and is a background Service. When the alarm goes off a notification shows up on the patients android device and when clicking on that, the app launches the check-in flow.