Android Capstone final specification

Option C: Symptom Management

Deliverables

All documents are saved under the Google drive link

https://drive.google.com/folderview?id=0B6gEbOeTEmOAMFNwbS1fbW9lM0k&usp=sharing

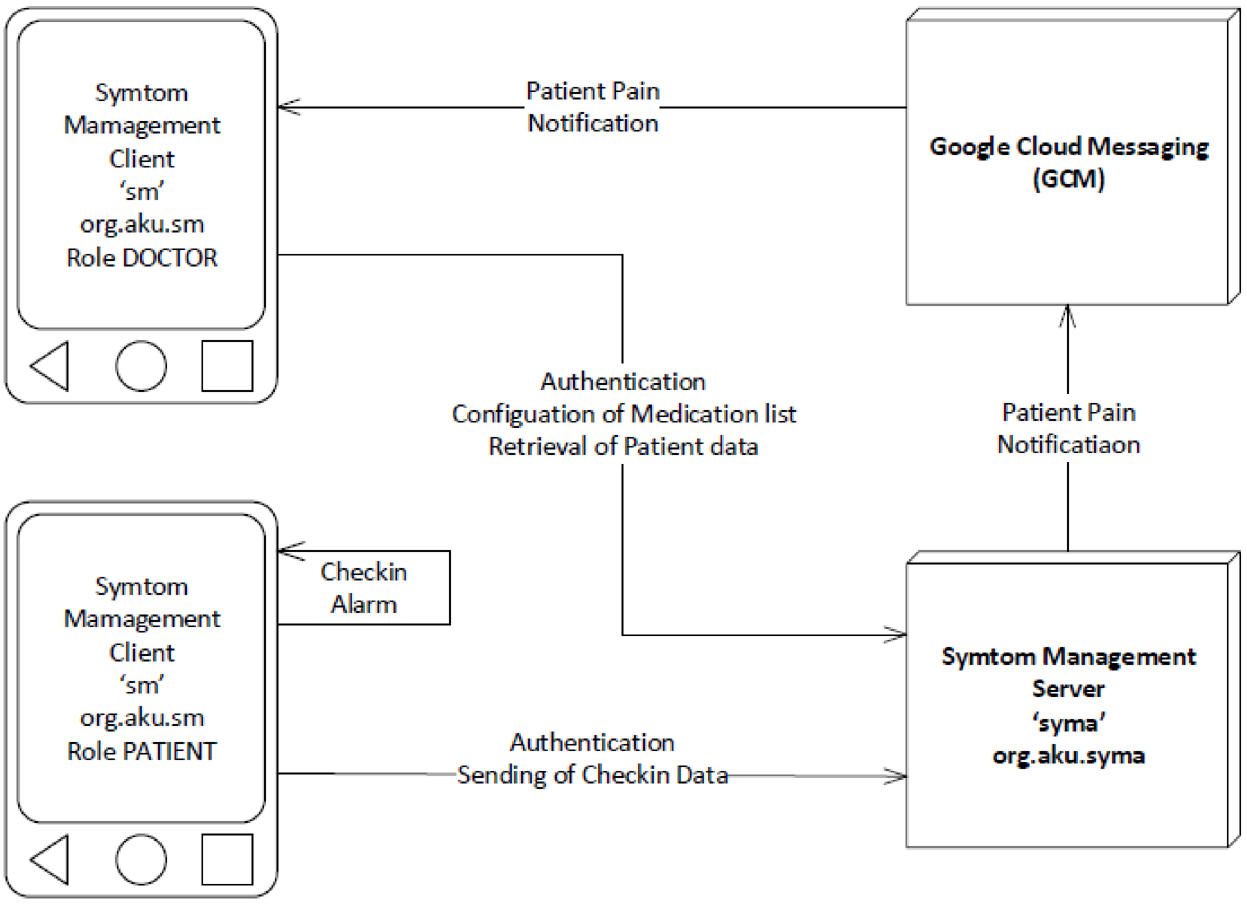
Document	**Description**
coursera_syma_final.pdf	The final specification, this document.
coursera_syma_capture.mp4	The film
syma_server_src.zip	The server source zip file. This is an Eclipse STS project.
Syma.jar	The server as executable jar file
tomcathost.jks	Self-signed server certificate to start with SSL. Didn't test with other host-name than mine which is aba.home.
sm_client_src.zip	The Android source zip file. This is a Android Studio project
sm-debug.apk	The Android APK file
syma_midterm.pdf	The mid term specification Just for information, not to review

Introduction

The document consists of three parts

1. Short introduction to the design
2. Screen-shots of application
3. Evaluation of Coursera requirements

Architecture Overview and Deployment Diagram

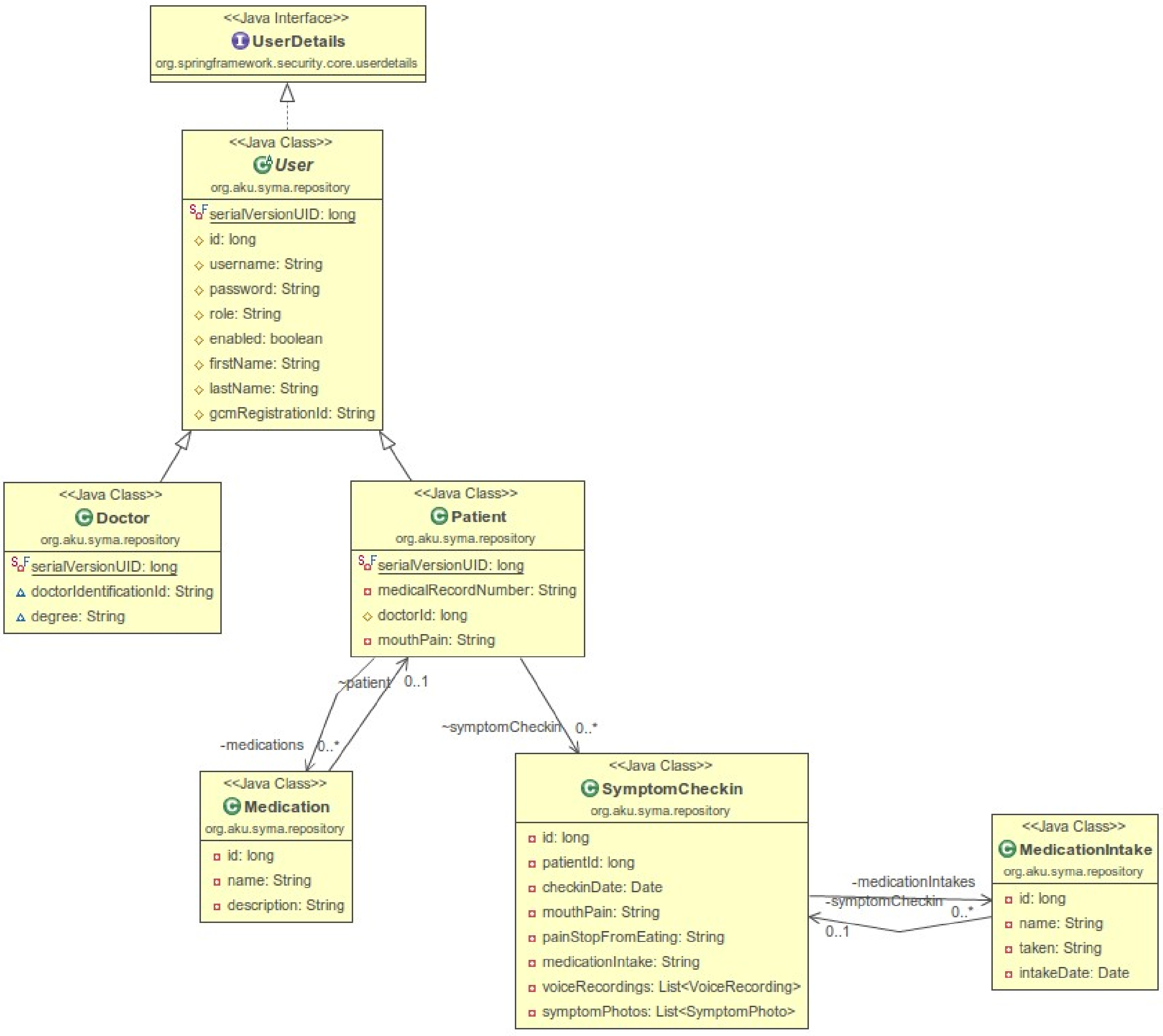


There is one APK file, at startup it performs :

- GCM registration
- Login to the server to submit the GCM id and retrieve the role.

The server may be started locally. There is a deployed version at Pivotal under the URL http://syma.cfapps.io.

Class Diagram



- A patient is assigned to a doctor. This relation is not modelled by a JPA relation, is is handled by the application. The attribute for the relation is Patient.doctorId.
- A Patient has Medication list assigned. The Medication list is updted by the assigned Doctor.
- A Patient a SymptomCheckinList assigned. The list entries can only be created by a Patient.
- A SymptomCheckin has a MedicationIntake list assigned. The MedicationIntake holds the answers of a patient about intake of Medications.

Implementation

The Spring based Server is implemented under the package `org.aku.syma.` Deliverd is the ZIP file syma.zip which is an Eclipse STS project.

The Android client is implemented under the package `org.aku.sm.` Deliverd is the ZIP file sm.zip which is an Android Studio project.

Test

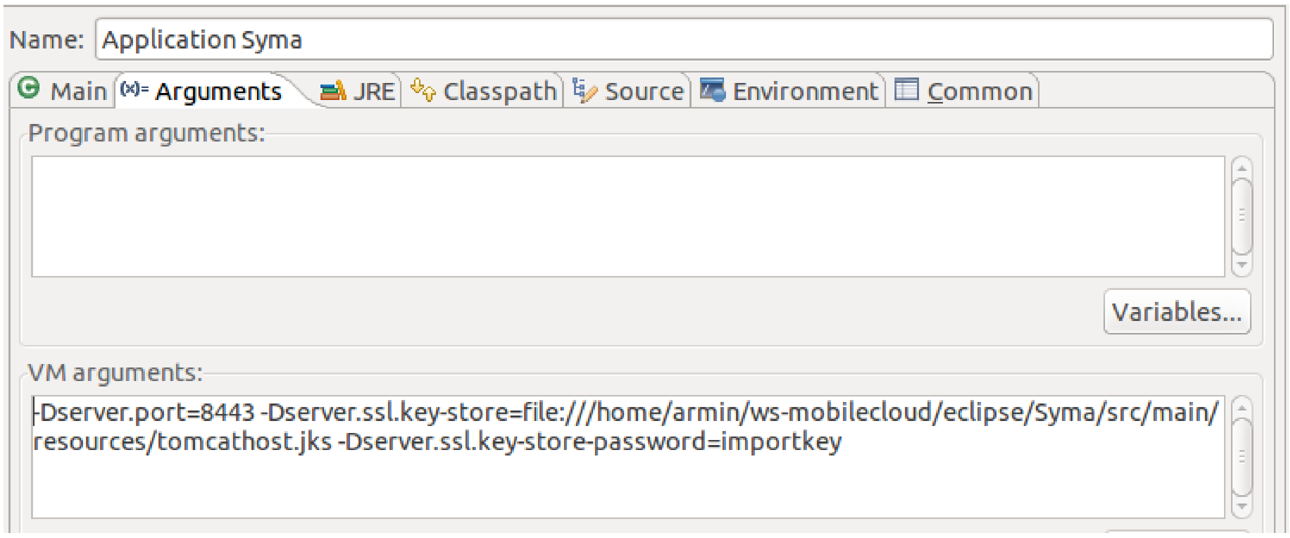
On startup the server is loading test-data, see class `InitializerService`. `Some` test-clients are

User name	Password	Role	Assigned to doctor	Patient has medication list	Patient has ckeck-ins
aflower	p0	ROLE_DOCTOR	---	---	---
btree	p0	ROLE_DOCTOR	---	---	---
cblack	p0	ROLE_PATIENT	aflower	Yes	Yes
dblue	p0	ROLE_PATIENT	aflower	Yes	Yes
jcarmine	p0	ROLE_PATIENT	aflower	No	No
...	...	...		...	...

Example to start the server executable jar under UNIX

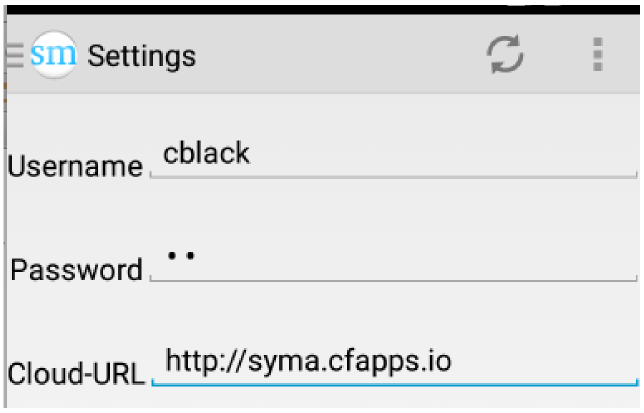
```
java -Dserver.port=8443 \
    -Dserver.ssl.key-store=file:///home/armin/ws-mobilecloud/eclipse/Syma/src/main/resources/tomcathost.jks \
    -Dserver.ssl.key-store-password=importkey -jar Syma.jar
```

To run from Eclipse use the java definitions from above and adjust:



```
-Dserver.port=8443 -Dserver.ssl.key-store=file:///home/armin/ws-mobilecloud/eclipse/Syma/src/main/resources/tomcathost.jks -Dserver.ssl.key-store-password=importkey
```

There is a non SSL-deployment of the server at Pivotal. The URL to configure under 'Settings' → 'Cloud-URL' is http://syma.cfapps.io

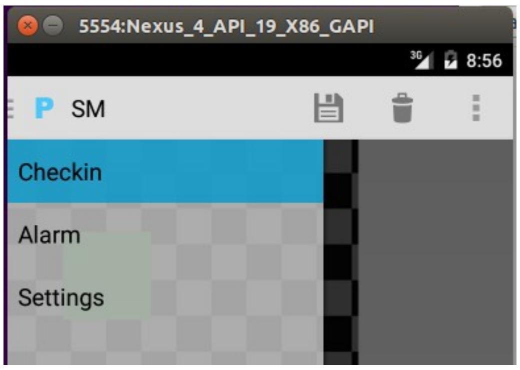
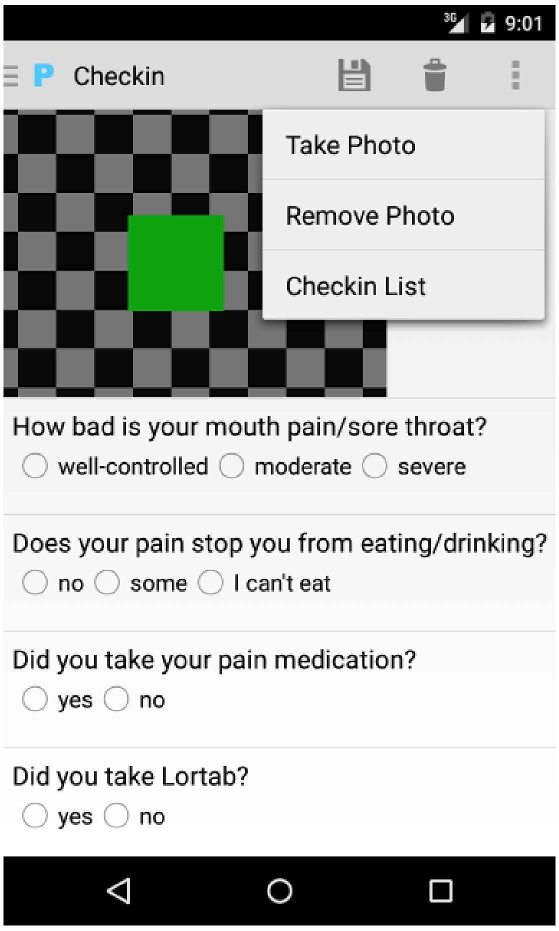


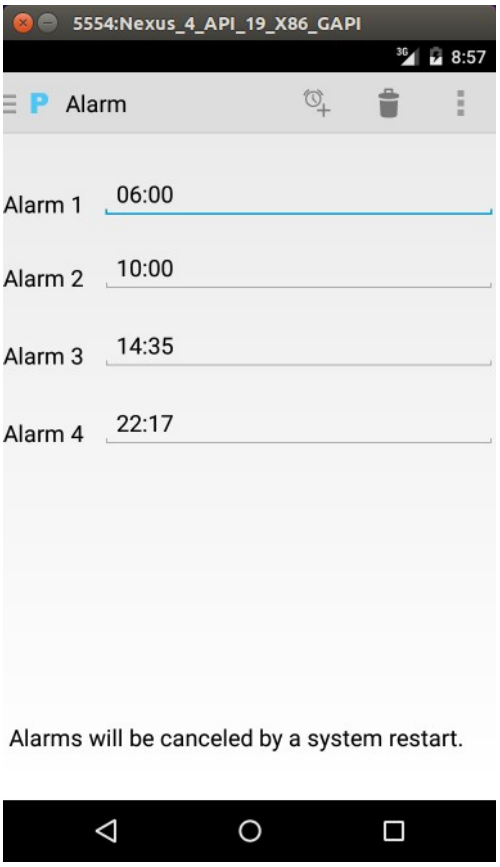
Username cblack

Password ••

Cloud-URL http://syma.cfapps.io

User Inteface Role Patient

 5554:Nexus_4_API_19_X86_GAPI 8:56 P SM Checkin Alarm Settings	**P: 1. Menu of a role Patient** Navigation of a patient A patient may select three screens from the menu • Checkin to add a new checking • Alarm to define the alarm settings • Settings to define the connection settings The blue **P** left of SM indicates role patient. The role is derved from login and thefore delivered from the server.
 9:01 P Checkin Take Photo Remove Photo Checkin List How bad is your mouth pain/sore throat? ◯ well-controlled ◯ moderate ◯ severe Does your pain stop you from eating/drinking? ◯ no ◯ some ◯ I can't eat Did you take your pain medication? ◯ yes ◯ no Did you take Lortab? ◯ yes ◯ no	**P: 2. Check-in** Main menu to record a checkin • Menu save to store entered value • Menu delete to clear input • Double tab on input are to capture a photo Overflow menu • To take or remove a photo • Checkin List to view the checkins already entered


5554:Nexus_4_API_19_X86_GAPI

8:57

Alarm

Alarm 1 06:00

Alarm 2 10:00

Alarm 3 14:35

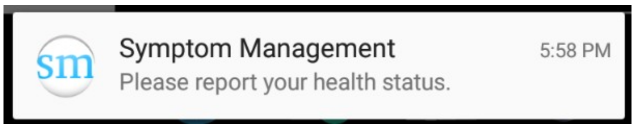
Alarm 4 22:17

Alarms will be canceled by a system restart.

P: 3. Alarms

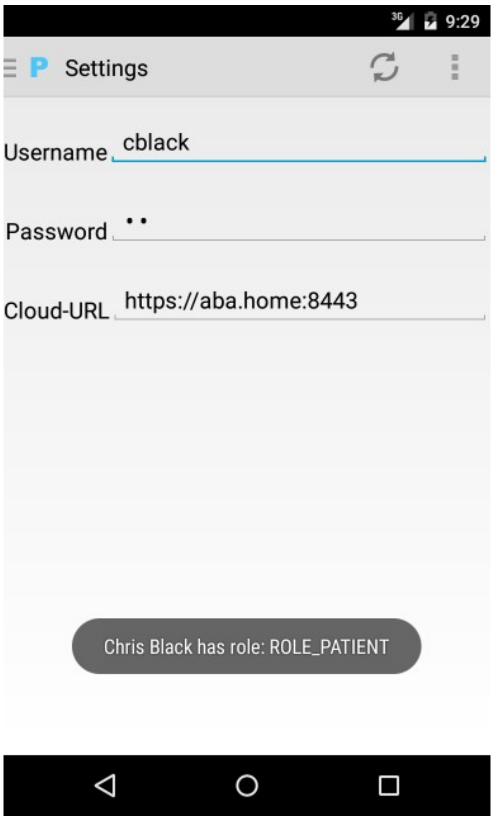
Four alarms may be defined. Only valid time values can be entered.

When an alam occurs a notification will be displayed.


Symptom Management 5:58 PM

Please report your health status.

Tab on the notification will start the application and display the input screen.


9:29

Settings

Username cblack

Password ••

Cloud-URL https://aba.home:8443

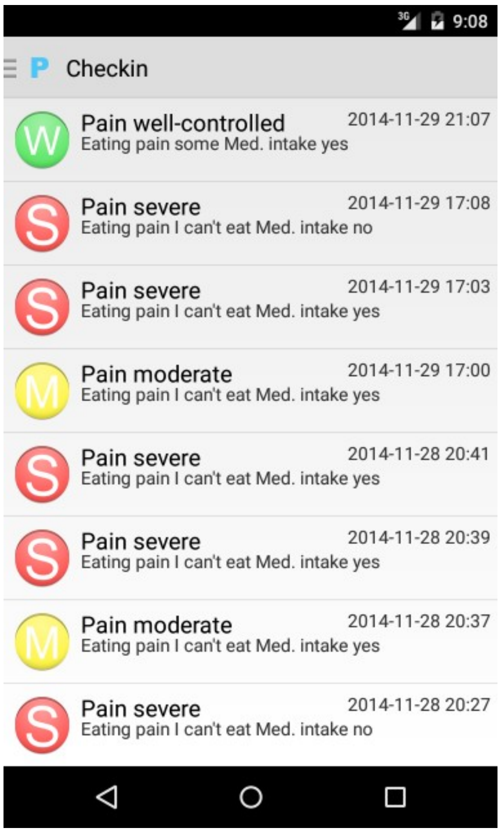
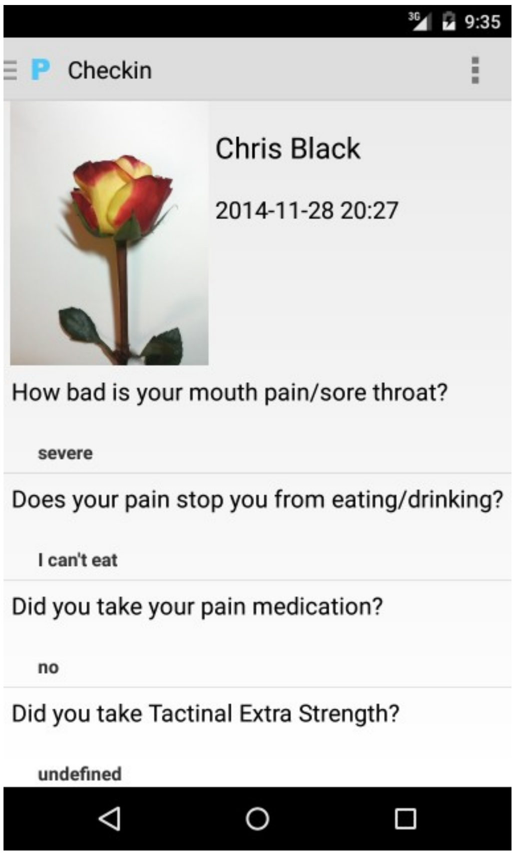
Chris Black has role: ROLE_PATIENT

P: 4. Settings

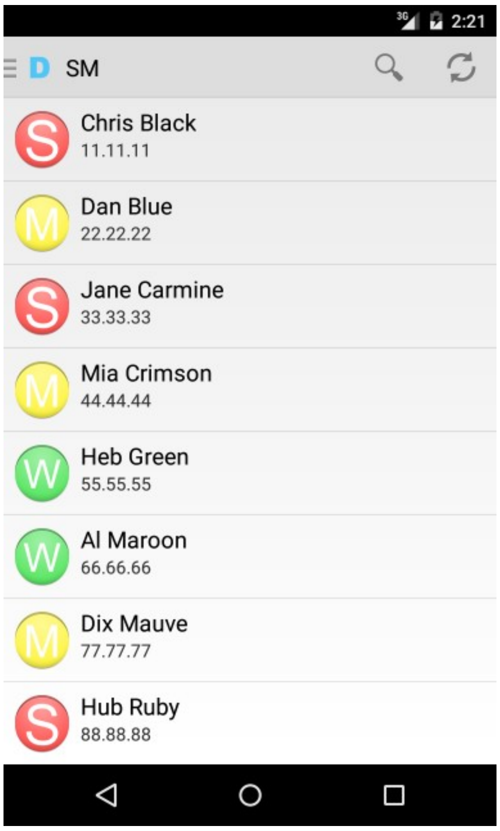
Settings screen to enter values to connect to the server.

Tab on refresh button performs reconnect and displays the current role.

Not the password is saved unencrypted in the private settings and on the server. This is not save but selected for ease of implementation.

 9:08 Checkin W Pain well-controlled 2014-11-29 21:07 Eating pain some Med. intake yes S Pain severe 2014-11-29 17:08 Eating pain I can't eat Med. intake no S Pain severe 2014-11-29 17:03 Eating pain I can't eat Med. intake yes M Pain moderate 2014-11-29 17:00 Eating pain I can't eat Med. intake yes S Pain severe 2014-11-28 20:41 Eating pain I can't eat Med. intake yes S Pain severe 2014-11-28 20:39 Eating pain I can't eat Med. intake yes M Pain moderate 2014-11-28 20:37 Eating pain I can't eat Med. intake yes S Pain severe 2014-11-28 20:27 Eating pain I can't eat Med. intake no	**P: 1. Check-in List** View of already entered check-ins. • Tab on a check-in to display the details.
 9:35 Checkin Chris Black 2014-11-28 20:27 How bad is your mouth pain/sore throat? severe Does your pain stop you from eating/drinking? I can't eat Did you take your pain medication? no Did you take Tactinal Extra Strength? undefined	**P: 5. Check-in Details** Check-in details to display the details of a symptom check-in Not yet implemented: Double tab on image to start image view.

User Interface Role Doctor



Chris Black 11.11.11
Dan Blue 22.22.22
Jane Carmine 33.33.33
Mia Crimson 44.44.44
Heb Green 55.55.55
Al Maroon 66.66.66
Dix Mauve 77.77.77
Hub Ruby 88.88.88

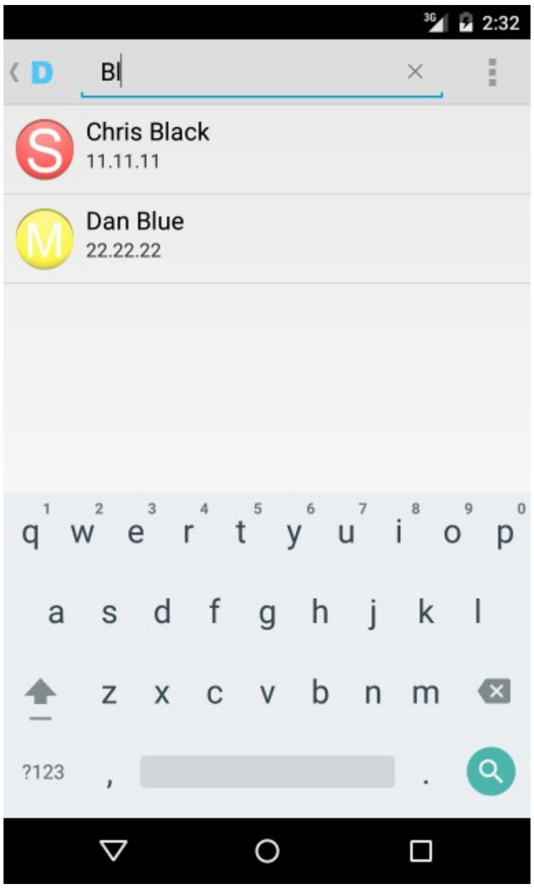
D: 1. Patient List

The list of the patients is loaded at startup from the server. The list is ordered descending by check-in date.

The pain level of the patient's last check-in is displayed as a colored icon

- S – severe pain
- M – moderate pain
- W – well controlled pain

Click on magnifying glass to display search input field.

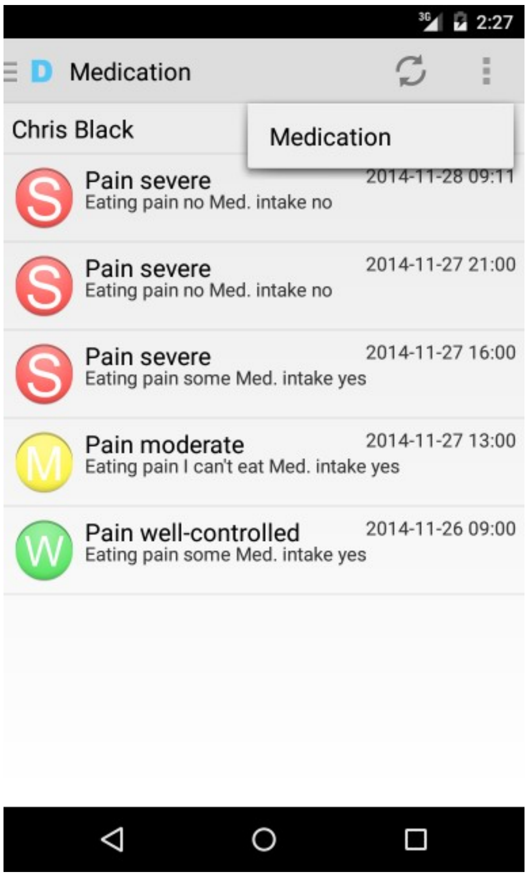


Chris Black 11.11.11
Dan Blue 22.22.22

D: 2. Search Patient

Click on magnifying glass of the navigation bar to display search input field. The list is sorted alphabetically by the last name.

Click on the magnifying glass of the keyboard to execute a server side search. The search is an SQL like search, the server-side is appending the SQL wild card character '%' to the search input.


2:27

Medication

Chris Black

Medication

S	Pain severe Eating pain no Med. intake no	2014-11-28 09:11
S	Pain severe Eating pain no Med. intake no	2014-11-27 21:00
S	Pain severe Eating pain some Med. intake yes	2014-11-27 16:00
M	Pain moderate Eating pain I can't eat Med. intake yes	2014-11-27 13:00
W	Pain well-controlled Eating pain some Med. intake yes	2014-11-26 09:00

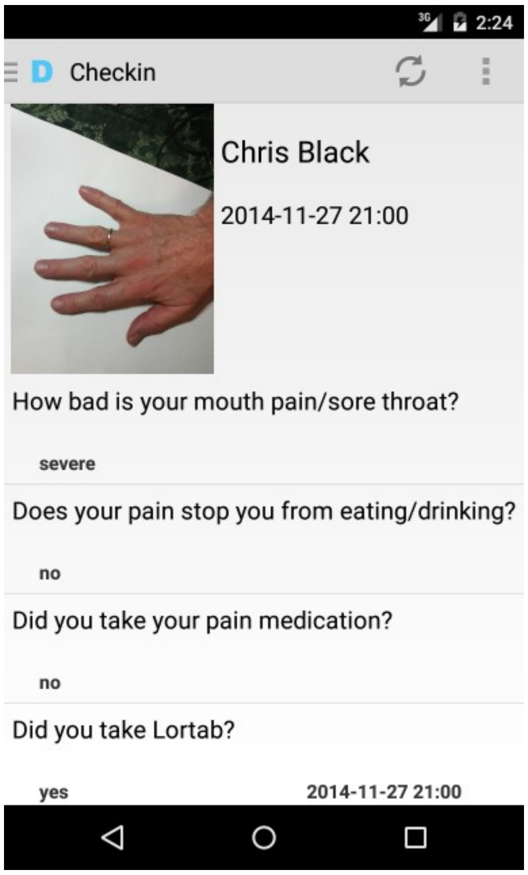
D: 3. Check-in List

Tab on patient opens the patient's check-in list. The check-in list is sorted descending by check-in date

The pain level is displayed as a colored icon

- S – severe pain
- M – moderate pain
- W – well controlled pain

The navigate bar overflow menu entry Medication leads to the patient's medication list.


2:24

Checkin

Chris Black

2014-11-27 21:00

How bad is your mouth pain/sore throat?

severe

Does your pain stop you from eating/drinking?

no

Did you take your pain medication?

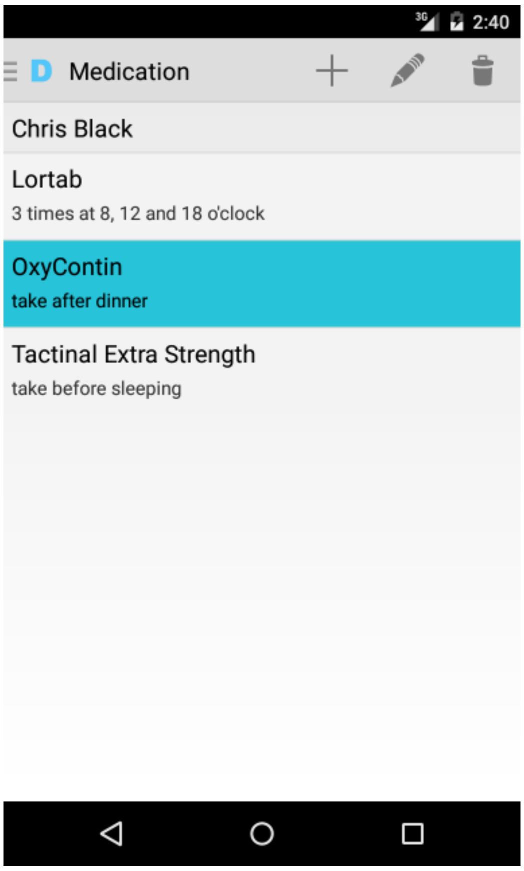
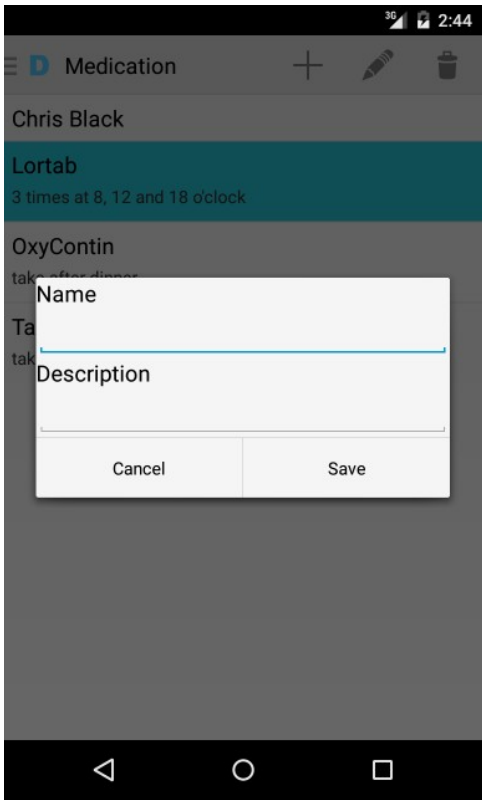
no

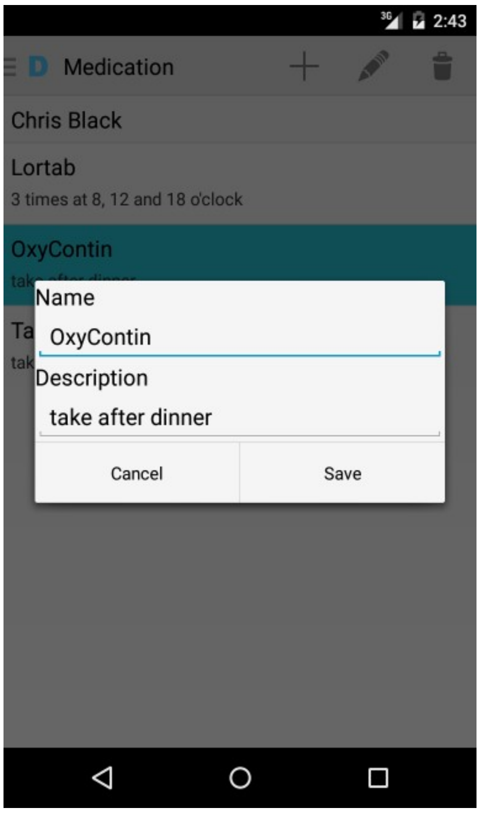
Did you take Lortab?

yes 2014-11-27 21:00

D: 4. Check-in Detail

- Same screen as the patient
- First the common check-in questions
- Following the medication questions

 2:40 Medication Chris Black Lortab 3 times at 8, 12 and 18 o'clock OxyContin take after dinner Tactinal Extra Strength take before sleeping	**D: 5. Medication List** Three icons • + to add a medication • pen to edit a medication • bin to delete a medication
 2:44 Medication Chris Black Lortab 3 times at 8, 12 and 18 o'clock OxyContin Name Description Cancel Save	**D: 6. Add Medication** • Tab + icon • Enter name and description • Save



Chris Black

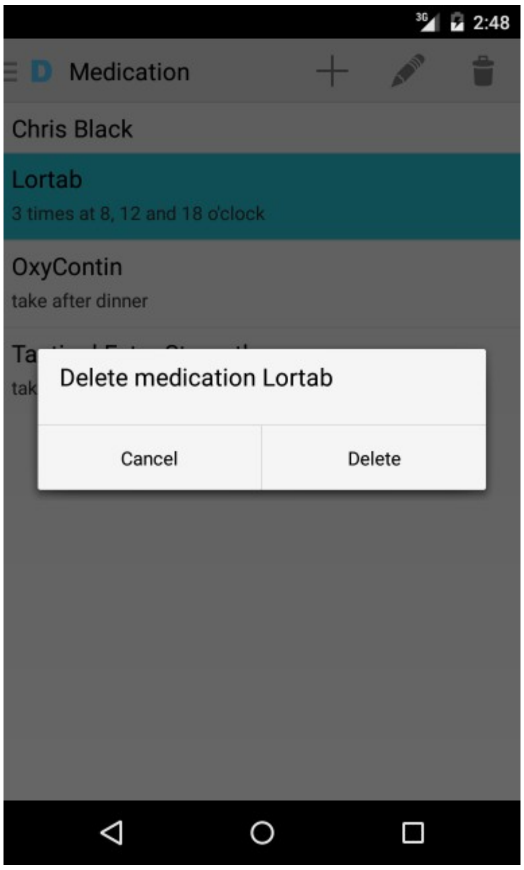
Lortab
3 times at 8, 12 and 18 o'clock

OxyContin

Name
OxyContin
Description
take after dinner

D: 7. Change Medication

- Select a medication list entry
- Tab the pen icon
- Modify name and / or description
- save



Chris Black

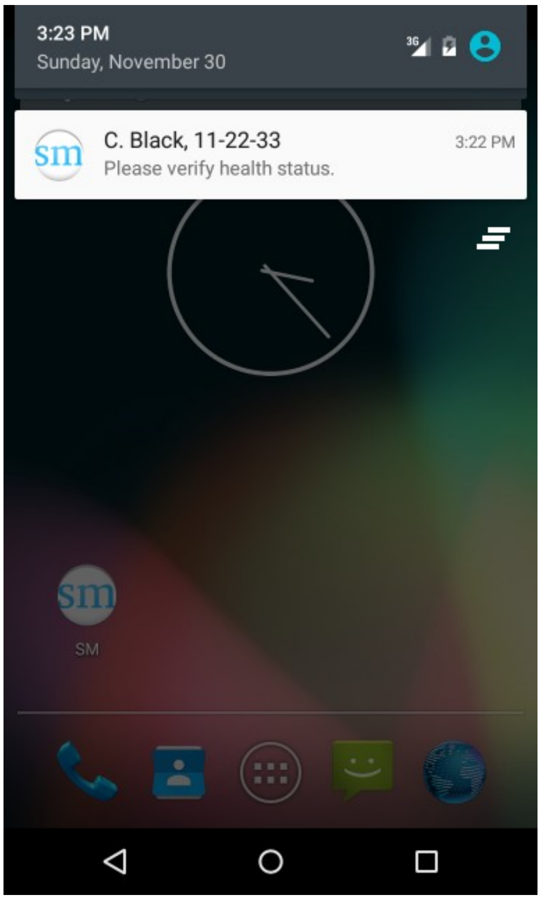
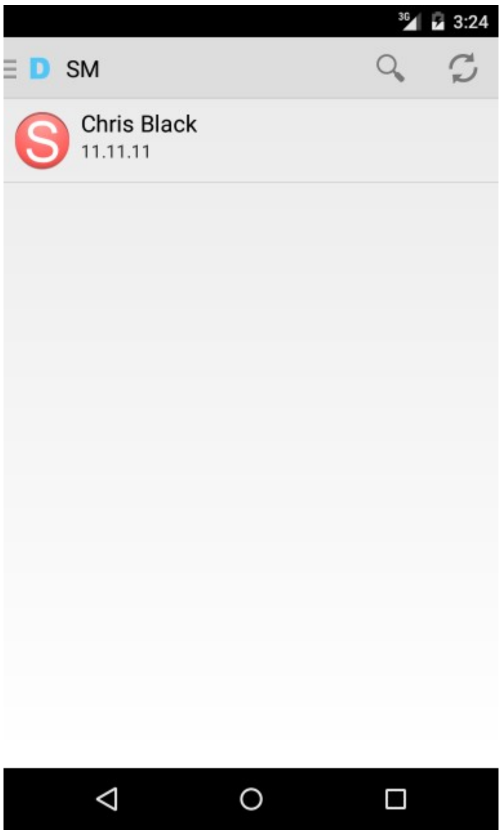
Lortab
3 times at 8, 12 and 18 o'clock

OxyContin
take after dinner

Delete medication Lortab

D: 8. Delete Medication

- Select a medication list entry
- Tab the bin icon
- delete

 3:23 PM Sunday, November 30 C. Black, 11-22-33 Please verify health status. 3:22 PM	**D: 9. Notification** When a patient performs a check-in and the rules for a notification of the doctor are true the assigned doctor is notified. • Notification is performed by GCM – Google Cloud Messaging. To display the patient of the notification tab the sm-notification entry.
 3:24 Chris Black 11.11.11	**D: 10. Display patient of notification** Tabbing the sm-notification entry opens the system management application and displays the patient list with the patient of the notification. To display the check-in list of the patient tab the list entry. For further navigation please see 'D: 3 Check-in List' and following.

Requirements overview

Basic		
1	syma_server_src	org.aku.syma.auth.JpaUserDetailService
2	syma_server_src	org.aku.syma.controller.MedicationService. createMedication
3.1	sm_client_src	org.aku.sm.MainActivity
3.2	sm_client_src	org.aku.sm.alarm.AlarmBroadcastReceiver
3.3	sm_client_src	org.aku.sm.gcm.GcmIntentService
3.4	sm_client_src	org.aku.sm.checkin.contentprovider
4	syma_server_src	org.aku.syma.controller.UserService org.aku.syma.controller.MedicationService
5	sm_client_src	org.aku.sm.service
6	sm_client_src	org.aku.sm.patient org.aku.sm.checkin org.aku.sm.medication org.aku.sm.settings
7	sm_client_src	org.aku.sm.checkin.NewCheckinFragment org.aku.sm.checkin.GestureImageView
8	sm_client_src	org.aku.sm.checkin.CheckinListFragment.LoadCheckingListTask org.aku.sm.medication.MedicationListAdapter.MedicationCommandLooperThread
Functional		
1	syma_server_src	org.aku.syma.repository.Patient
2	sm_client_src	org.aku.sm.alarm.AlarmFragment
3	sm_client_src	org.aku.sm.alarm.AlarmFragment org.aku.sm.checkin.NewCheckinFragment
4	sm_client_src	org.aku.sm.checkin.NewCheckinFragment org.aku.sm.checkin.adapters.CheckinQuestionListAdapter
5	sm_client_src	org.aku.sm.checkin.NewCheckinFragment org.aku.sm.checkin.adapters.CheckinQuestionListAdapter
6	sm_client_src	org.aku.sm.checkin.NewCheckinFragment org.aku.sm.checkin.adapters.CheckinQuestionListAdapter
7	sm_client_src	org.aku.sm.checkin.NewCheckinFragment org.aku.sm.checkin.adapters.CheckinQuestionListAdapter
8	sm_client_src	org.aku.sm.checkin.NewCheckinFragment org.aku.sm.checkin.adapters.CheckinQuestionListAdapter
9	syma_server_src	org.aku.syma.repository.Doctor
10	sm_client_src	org.aku.sm.patient.PatientListFragment
11	sm_client_src	org.aku.sm.patient.PatientListFragment.onOptionsItemSelected() with id R.id.menu_search

12	sm_client_src	org.aku.sm.medication
13	syma_server_src	org.aku.syma.controller.MedicationService.painLevelExceeded `org.aku.syma.gcm`
13	syma_server_src sm_client_src	`org.aku.syma.controller.AuthenticationTest` org.aku.sm.service.SslSetup

Basic Project Requirements

1. App supports multiple users via individual user accounts

- Only registered user can access the system after login.
- A user has either the role ROLE_DOCTROR or ROLE_PATIENT.
- System is using Spring Security for access control and implementing Spring `UserDetailsService`
- See Class `org.aku.syma.auth.JpaUserDetailService`

2. App contains at least one user facing function available only to authenticated users

- Any access to the system must be authenticated
- See class `org.aku.syma.auth.SecurityConfiguration`

```
    http.authorizeRequests()
        .antMatchers("/**", "/login/**").hasAnyRole("DOCTOR", "PATIENT")
        .and()
        .httpBasic();

http.authorizeRequests().antMatchers("/doctor/**").hasRole("DOCTOR");
http.authorizeRequests().antMatchers("/patient/**").hasAnyRole("DOCTOR", "PATIENT");
http.authorizeRequests().antMatchers(HttpMethod.POST, "/symptomCheckin", "/checkin").hasRole("PATIENT");
http.authorizeRequests().antMatchers(HttpMethod.GET,  "/symptomCheckin", "/checkin").hasAnyRole("DOCTOR",
"PATIENT");
http.authorizeRequests().antMatchers(HttpMethod.POST,  "/notify/**").hasAnyRole("DOCTOR", "PATIENT");
```

- Example for role based authorization
- See `org.aku.syma.controller.MedicationService`

```
@Secured("ROLE_PATIENT")
@RequestMapping(value="/checkin/{symptomCheckinId}/image",
                                method=RequestMethod.POST)
public @ResponseBody Medication createMedication(@PathVariable long patientId,
                        @RequestBody Medication medication) {
```

Example for principal based authorization
`org.aku.syma.repository.PatientRepository`

```
@PostAuthorize("returnObject.Id == principal.id || returnObject.doctorId == principal.id")
Patient findByIdAndDoctorId(long id, long doctorId);
```

3. App comprises at least 1 instance of each of at least 2 of the following 4 fundamental Android components

3.1 Activity

- The main start-up gui of the application is an activity.
- See org.aku.sm.MainActivity

3.2 BroadcastReceiver

- Alarms defined by a patient are retrieved by a BroadcastReceiver
- See org.aku.sm.alarm.AlarmBroadcastReceiver

3.3 Service

- IntentService is used to handle Google Cloud Messages (GCM).
- See org.aku.sm.gcm.GcmIntentService

3.4 ContentProvider

- The patients check-in data is stored by a content provider, Allowing the patient to see old checkin.
- See org.aku.sm.checkin.contentprovider
- See AndroidManifest.xml <provider> tags

4. App interacts with at least one remotely-hosted Java Spring-based service

All the data for the application is managed by a Java Spring-based application server. Access is provided by the means of Spring-based services (JSON over HTTP).

There are two Spring controllers to process the data
1. UserService – Handling user login and Doctor / Patient data.
   See `org.aku.syma.controller.UserService`
2. MedicationService – Handling symptom check-in data
   See `org.aku.syma.controller.MedicationService`

5. App interacts over the network via HTTP

See point 4.)
See org.aku.sm.service

6. App allows users to navigate between 3 or more user interface screens at runtime

The symptom management app hat two modes: doctor- or patient-mode. The role a user gets will be decided during login.

The user interface screens of a doctor are:
- Patient List – Package org.aku.sm.patient
- Symptom Checkin List – Package org.aku.sm.checkin
- Symtom Checking Detail – Package org.aku.sm.checkin
- Medication List – Package org.aku.sm.medication
- Settings - Package org.aku.sm.settings

The user interface screens of a patien are:
- Symptom Checkin Input – Package org.aku.sm.checkin
- Symtomp Checking Detail – Package org.aku.sm.checkin

- Symptom Checkin List – Package org.aku.sm.checkin
- Alarm - Package org.aku.sm.alarm
- Settings – Package org.aku.sm.settings

7. App uses at least one advanced capability or API from the following list (covered in the MoCCA Specialization): multimedia capture, multimedia playback, touch gestures, sensors, animation.**

**Learners are welcome to use ADDITIONAL other advanced capabilities (e.g., BlueTooth, Wifi-Direct networking, push notifications, search), but must also use at least one from the MoCCA list.

The symptom management app makes use of camera to report the health status. Double tap on the image starts the camera.
See org.aku.sm.checkin.NewCheckinFragment
See org.aku.sm.checkin.GestureImageView

8. App supports at least one operation that is performed off the UI Thread in one or more background Threads of Thread pool.

Data exchange by the SM-Android client with the Syma application server is performed using the Android AsyncTask framework.
There are several task implemented, example

- org.aku.sm.checkin.CheckinListFragment.LoadCheckingListTask

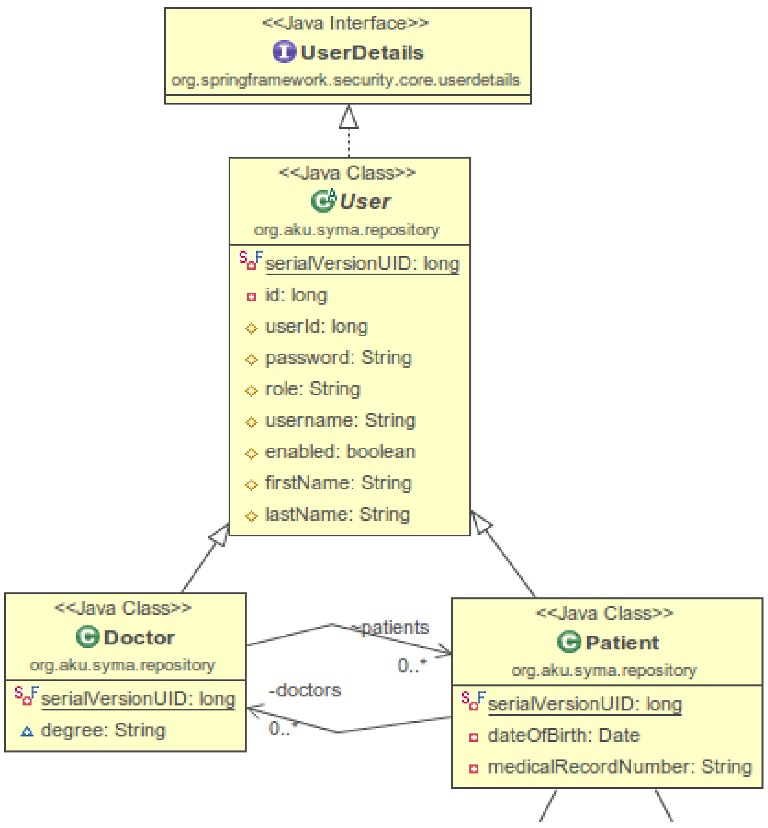
The medication list is updated by a Looper thread

- See org.aku.sm.medication.MedicationListAdapter.MedicationCommandLooperThread

Functional Description and App Requirement

1. App identifies a Patient as a user with first name, last name, date of birth, a (unique) medical record number, and possibly other identifying information). A patient can login to their account.

Login to the Syma server is realized using HTTP basic authentication and Spring Authentication. Both doctors and patients must login to the Syma server to gain access to the data. There are be three entities to model this relation. A generic User entity used by the login manager and a Doctor and Patient entity to hold doctor and patient specifc data. Doctor and Patient entity are derived from the User entity.



2. App defines a Reminder as an alarm or notification which can be set to patient-adjustable times (at least four times per day).

See chapter, User Inteface Role Patient, P: 3. Alarms
See org.aku.sm.alarm.AlarmFragment

3. A Reminder triggers a Check-In, which is defined by the app as a unit of data associated with a Patient, a date, a time, and that patient's responses to various questions at that date and time.

See chapter, User Inteface Role Patient, P: 3. Alarms and P: 2 Check-in
See org.aku.sm.checkin.NewCheckinFragment

4. Check-In includes the question, "How bad is your mouth pain/sore throat?" to which a patient can respond, "well-controlled," "moderate," or "severe.

See chapter, User Inteface Role Patient, P: 2 Check-in
See org.aku.sm.checkin.NewCheckinFragment
See org.aku.sm.checkin.adapters.CheckinQuestionListAdapter

5. Check-In includes the question, "Did you take your pain medication?" to which a Patient can respond "yes" or "no".

See chapter, User Inteface Role Patient, P: 2 Check-in
See org.aku.sm.checkin.NewCheckinFragment
See org.aku.sm.checkin.adapters.CheckinQuestionListAdapter

6. A Check-In for a patient taking more than one type of pain medication includes a separate question for each medication (e.g., “Did you take your Lortab?” followed by “Did you take your OxyContin?”). The patient can respond to these questions with “yes” or “no.”

See chapter, User Inteface Role Patient, P: 2 Check-in
See org.aku.sm.checkin.NewCheckinFragment
See org.aku.sm.checkin.adapters.CheckinQuestionListAdapter

7. During a Check-In, if a patient indicates he or she has taken a pain medication, the patient will be prompted to enter the time and date he or she took the specified medicine.

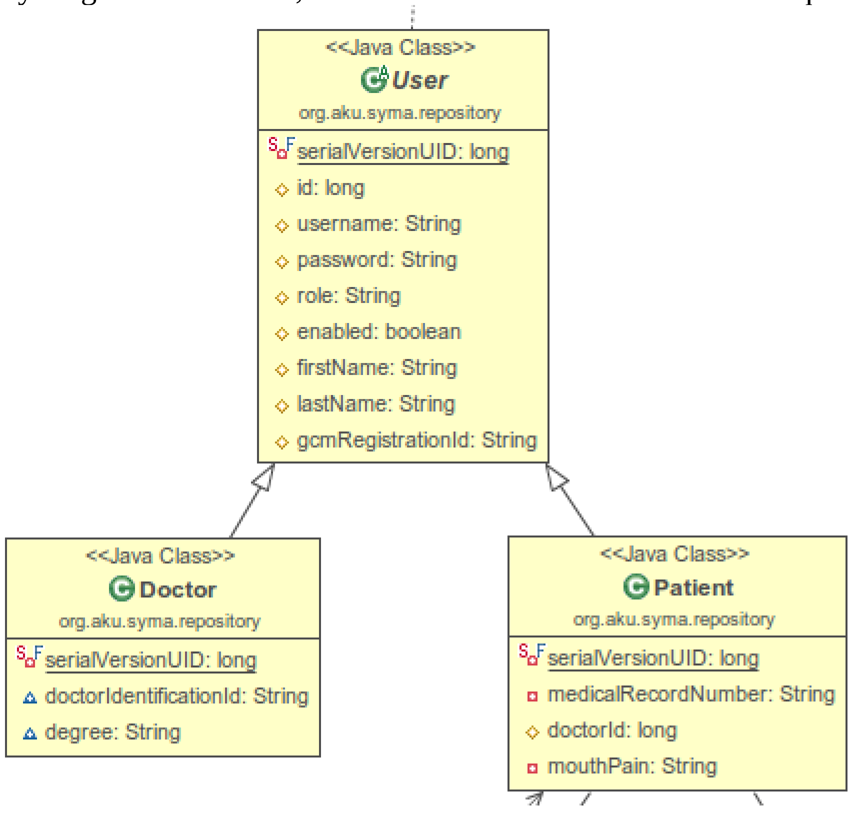
See chapter, User Inteface Role Patient, P: 2 Check-in
See org.aku.sm.checkin.NewCheckinFragment
See org.aku.sm.checkin.adapters.CheckinQuestionListAdapter

8. During a Check-In, the patient is asked “Does your pain stop you from eating/drinking?” To this, the patient can respond, “no,” “some,” or “I can’t eat.

See chapter, User Inteface Role Patient, P: 2 Check-in
See org.aku.sm.checkin.NewCheckinFragment
See org.aku.sm.checkin.adapters.CheckinQuestionListAdapter

9. App defines a Doctor as a different type of user with a unit of data including identifying information (at least first name, last name, and a unique doctor ID) and an associated list of Patients that the doctor can view a list of. A doctor can login.

The class hierarchy diagram of the User, Doctor and Patient entities with the required attributes:



- A patient is assigned to a doctor by the attribute Patient.doctorId.
- Any user has to login to access the system.
- A doctor can only view data of patients assigned to him.
- See `org.aku.syma.repository.Doctor`

10. App allows a patient's Doctor to monitor Check-Ins, with data displayed graphically. The data is updated at some appropriate interval (perhaps when a Check-In is completed).

The checkin data of the doctor is update when starting the symptom management application or tabing the refresh button of the patient list GUI. The patient list is ordered by descending check-in date.

Display of data see

- User Interface Role Doctor, D: 1 Patient.
- D3: Check-in list
- D4: Check-in detail

See org.aku.sm.patient.PatientListFragment

11. A doctor can search for a given Patient's Check-In data by the patient's name (an exact text search hosted server-side). Note: Non-exact text searching is not required (e.g. you don't have to suggest, "Did you mean…")

See chapter User Interface Doctor, D2: Search Patients

See org.aku.sm.patient.PatientListFragment.onOptionsItemSelected() with id R.id.menu_search

12. A doctor can update a list of pain medications associated with a Patient. This data updates the tailored questions regarding pain medications listed above in (6).

See User Inteface Role Doctor, D: 3. Check-in List
See User Inteface Role Doctor, D: 5. Medication List
See User Inteface Role Doctor, D: 6. Add Medication
See User Inteface Role Doctor, D: 7. Change Medication
See User Inteface Role Doctor, D: 8. Delete Medication
See classes in package org.aku.sm.medication.

13. A doctor is alerted if a patient experiences 12 of “severe pain,” 16 or more hours of “moderate” or “severe pain” or 12 hours of “I can’t eat.”

The duration of a patients pain is calculated on the server when the patient performs a check-in. If the notification condition is true the doctor is notified by Google Cloud Messaging.
See See User Interface Role Doctor, D: 9 Notification
See See User Interface Role Doctor, D: 10 Display Patient notification

Implementation see: `org.aku.syma.controller.MedicationService.painLevelExceeded`
Test see: `org.aku.syma.controller.PainLevelTest.testMouthPain`

14. A patient’s data should only be accessed by his/her doctor over HTTPS.

Any access to the server is using HTTPS
See Android org.aku.sm.service.SslSetup
See Spring startup options

```
-Dserver.port=8443 \
    -Dserver.ssl.key-store=file:///home/armin/ws-mobilecloud/eclipse/Syma/src/main/resources/tomcathost.jks \
    -Dserver.ssl.key-store-password=importkey
```

Exmaple: test `org.aku.syma.controller.AuthenticationTest`
and chapter test