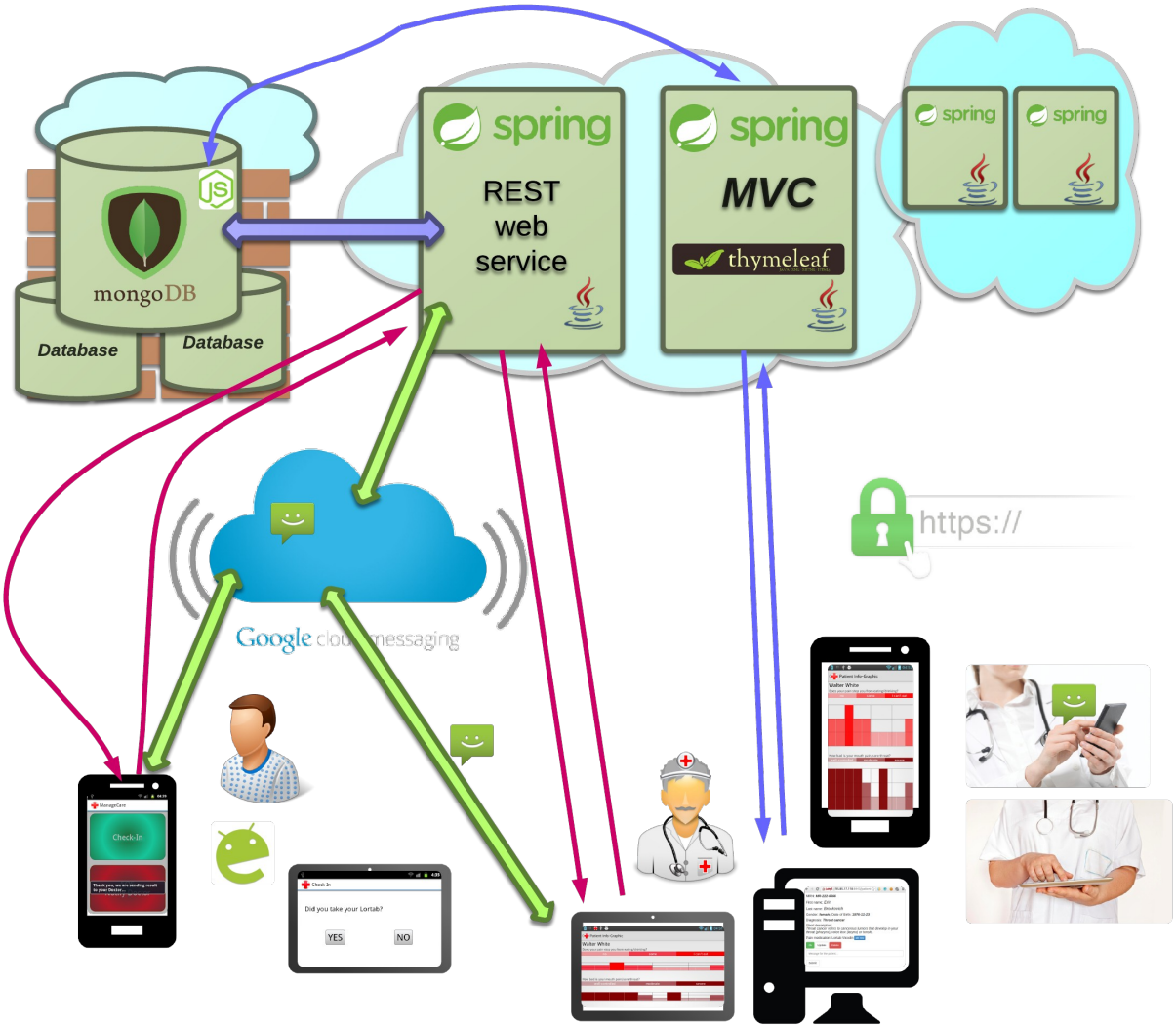
“ManageCare“

medical system

System scheme



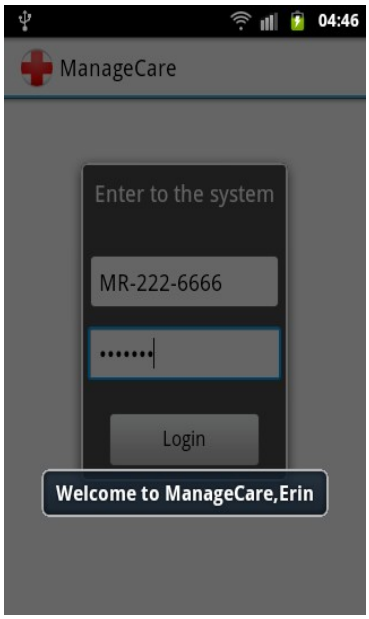
I hope the scheme explains pretty much design of system infrastructure. All data transfer via https.

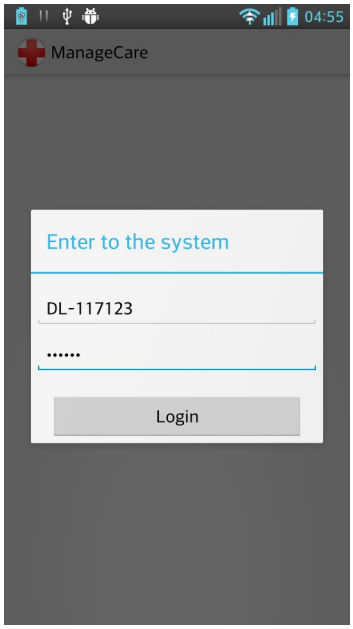
One user can access his account from several devices (even simultaneously) – from Android powered devices, and only doctor can enter to server-part of system from any device.

Mobile App

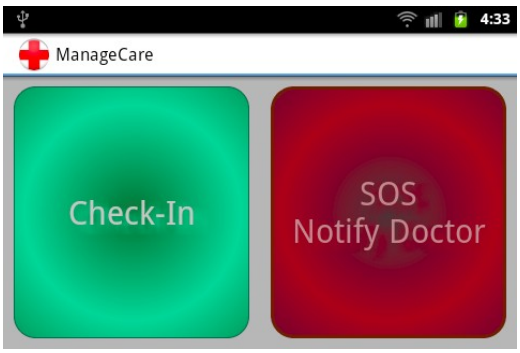
App supports old versions of Android OS since Android 2.3.* (api level 9, version name "GINGERBREAD") It covers (approximately) 99% active devices.

Patients and medical staff use the same App, but depending on credentials program looks and works absolutely different way.

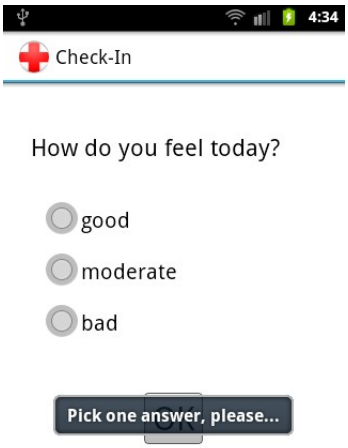


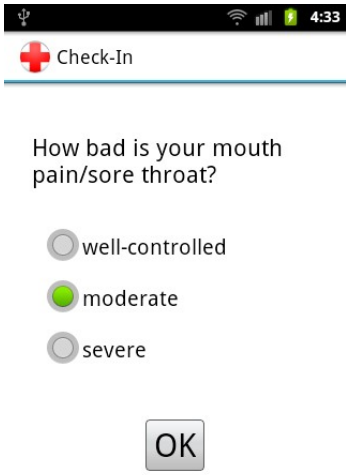


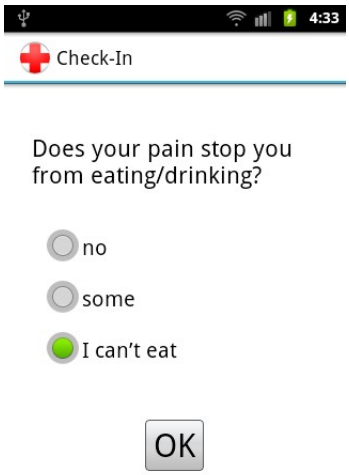
I tried to keep things for users as simple as possible, so Patients part “start screen” has only 2 buttons that allow user to answer Check-In or send an urgent message to Doctor.

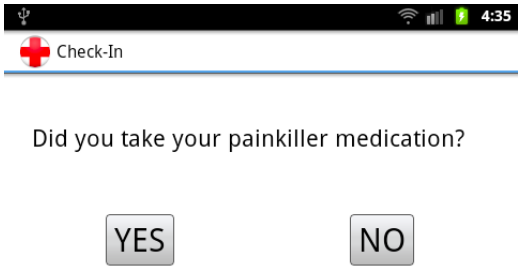


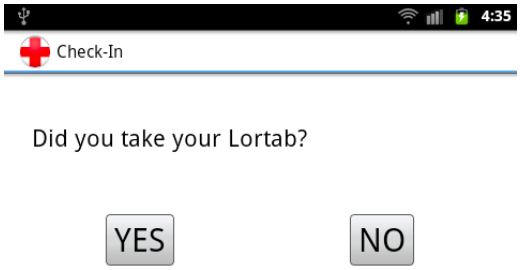
Android Capstone Project 2014

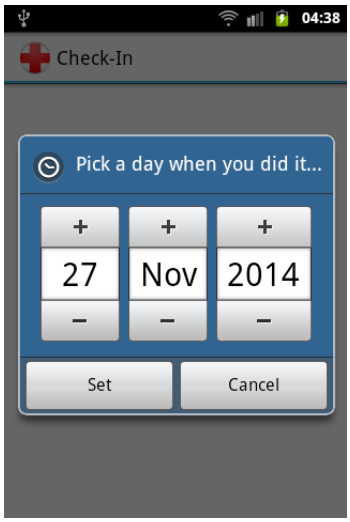


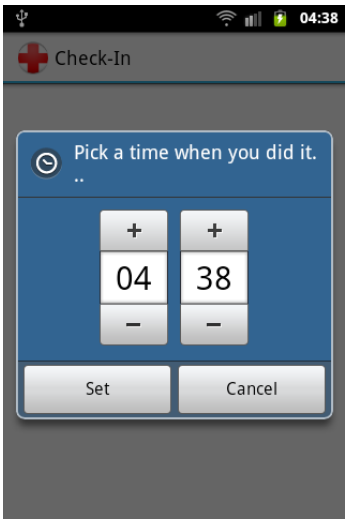


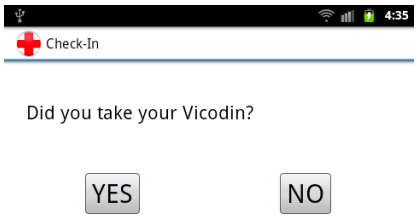


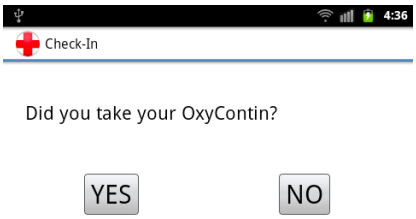


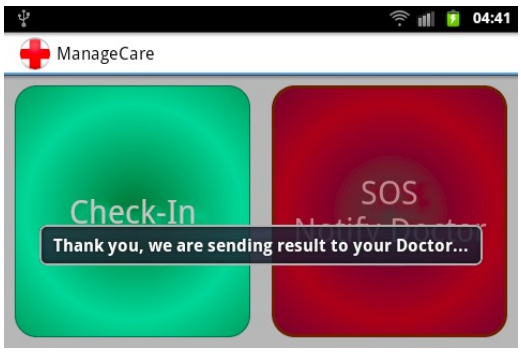


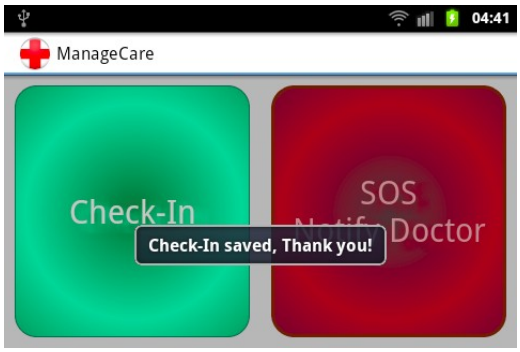




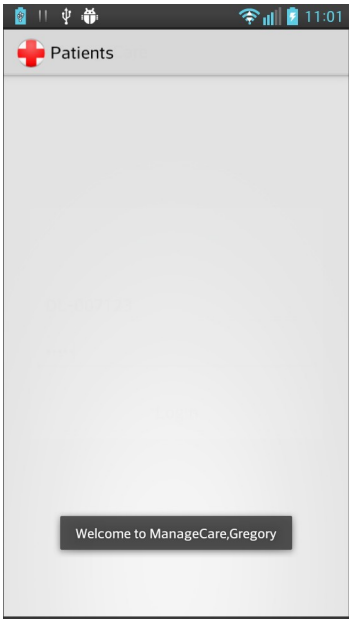


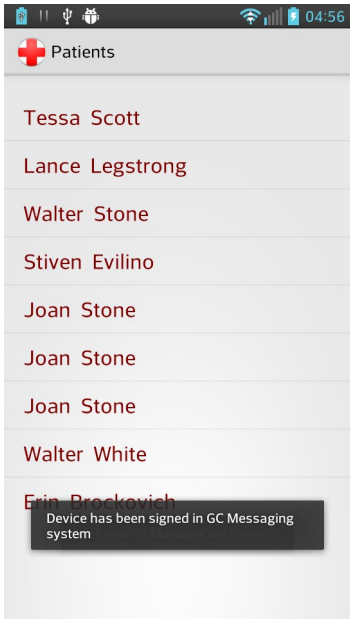


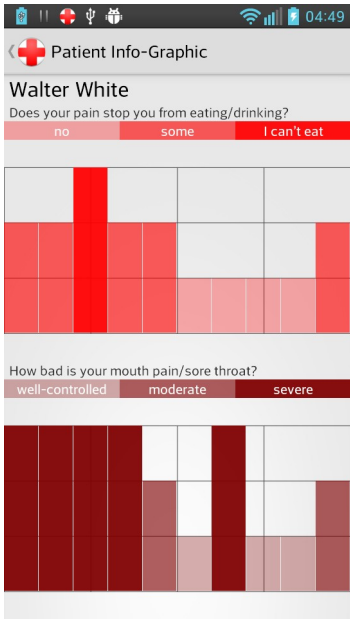


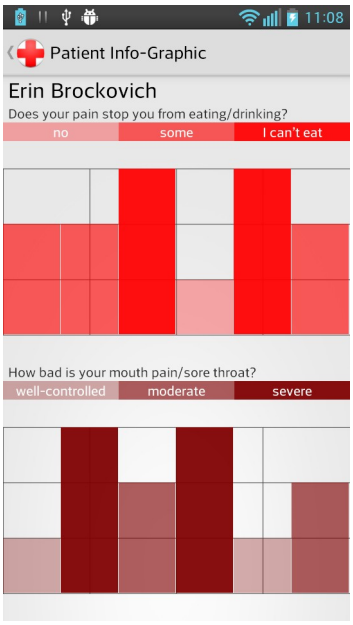


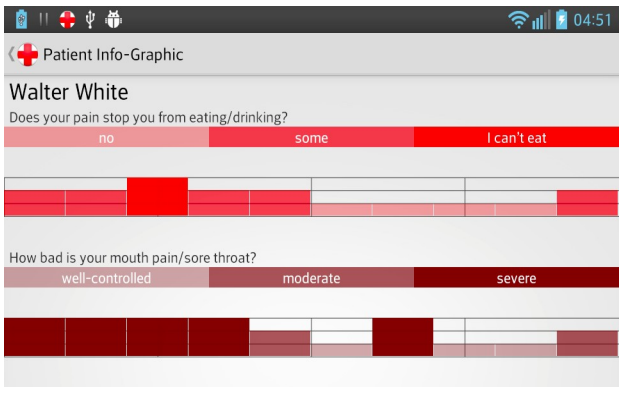
Medic part starts with list of all Doctor's patients and after pressing on the Patient open Info-graphic screen that provide info about patient's condition.



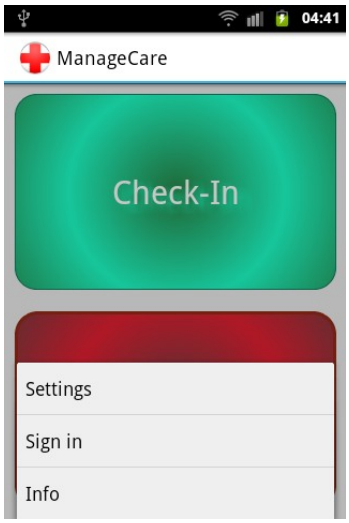


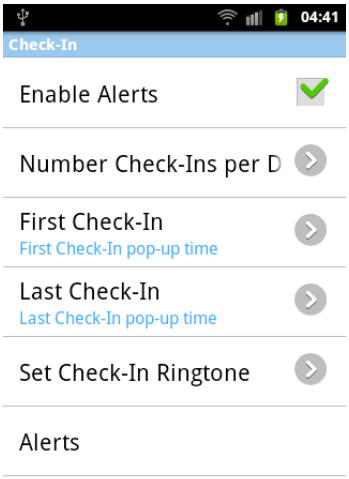


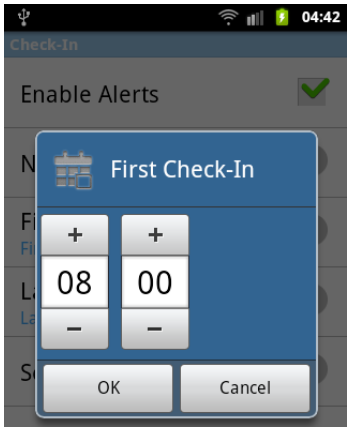


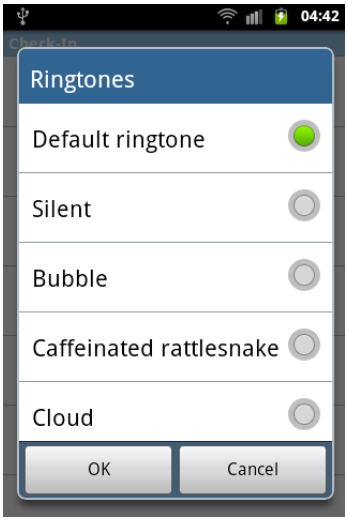


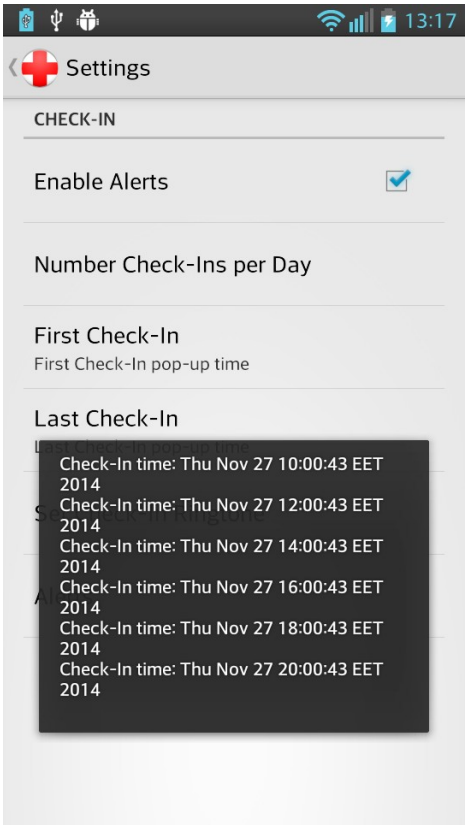
The setting also are as easy as possible. Patient can reenter, get info about himself or to adjust Check-In alerts schedule, pick a ring-tone for alerts. There are also default settings (5 per/day)

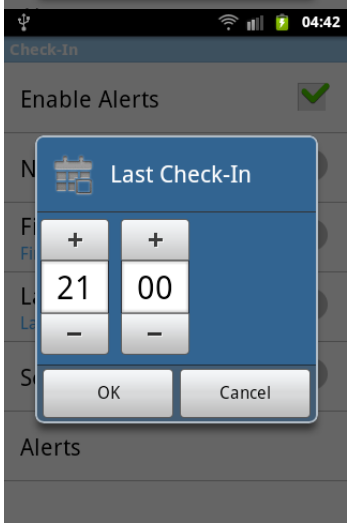








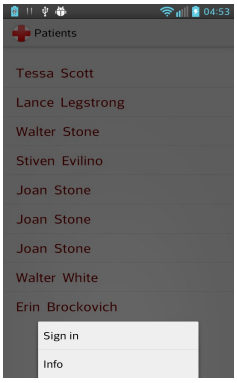


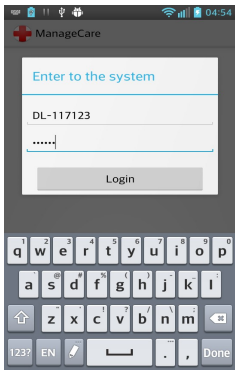


Alert pop ups Check-In screen, and Patient just has to answer 4 – 10 questions. The system trying not to use exact hours time to avoid server overhead.

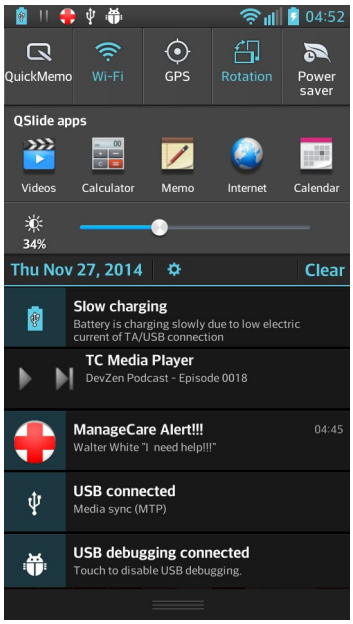
After rebooting device Check-In pop-up tasks schedule persists, and user do not need to re-adjust it (by default, all alarms are canceled when a device shuts down). I did it using intent “android.intent.action.BOOT_COMPLETED” and handling it in the BroadcastReceiver.

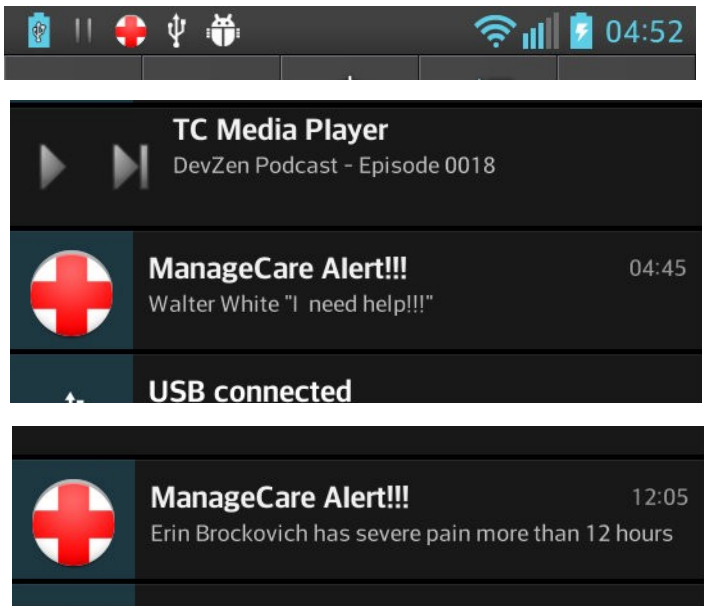
And nothing to adjust for Doctor, only reenter and info.



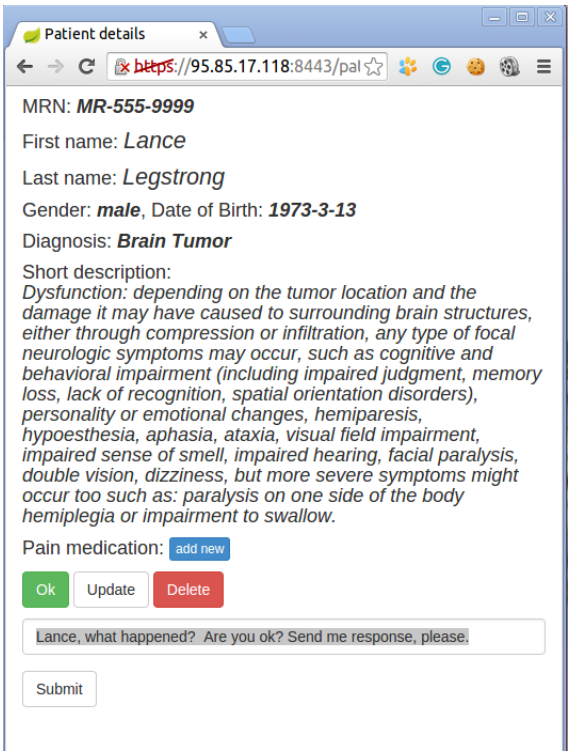


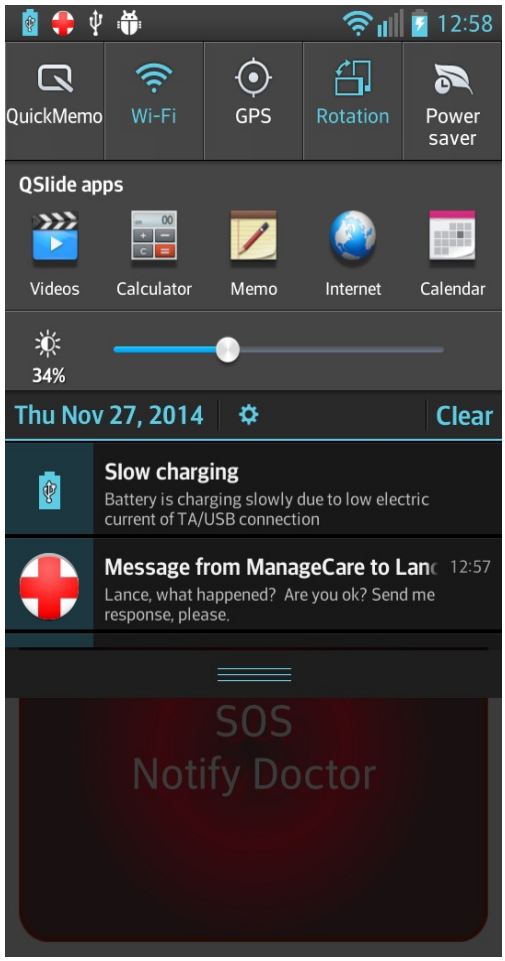
Doctor also gets Notifications (I used GCM) if something wrong accordingly project specification about patient's answers. And Doctor is receiving messages from his/her Patients.





Doctor can also send a messages to any of his/her patients via web interface:

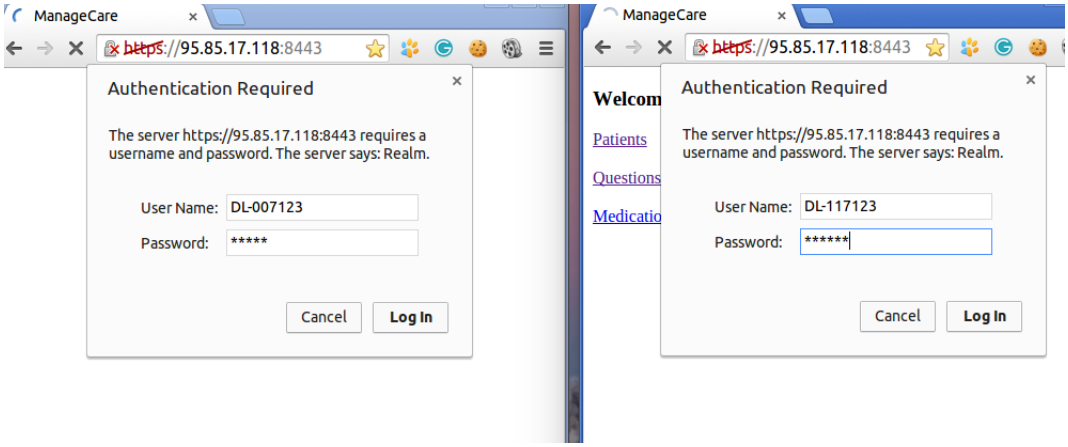




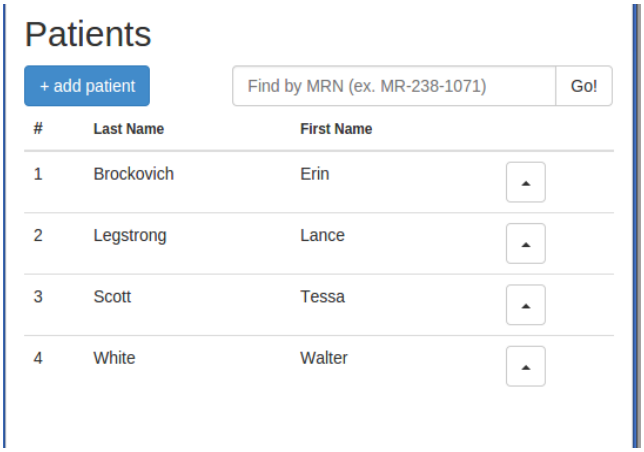
Web App

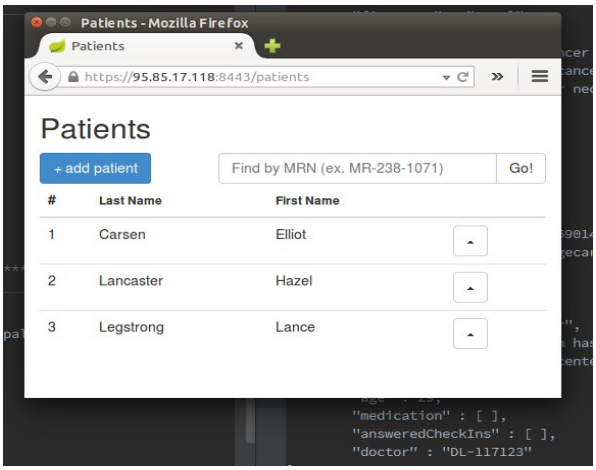
A lot of things Doctor can do via web console with practically any device (even phone).

Browse a list all his/her patients (below two different medics enter to the App with their credentials:

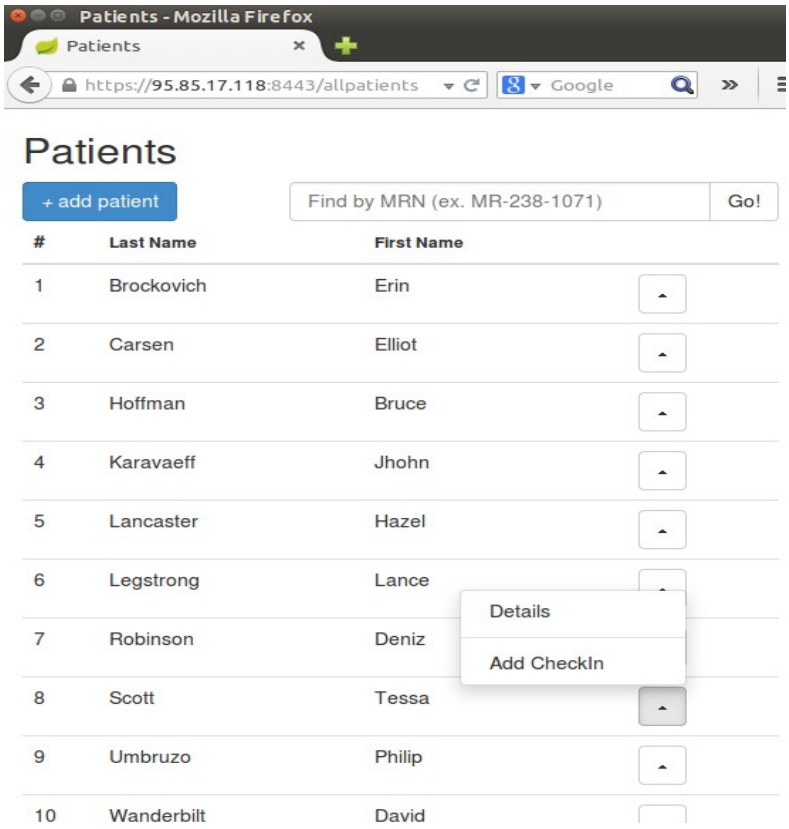


And can see his/her Patients:

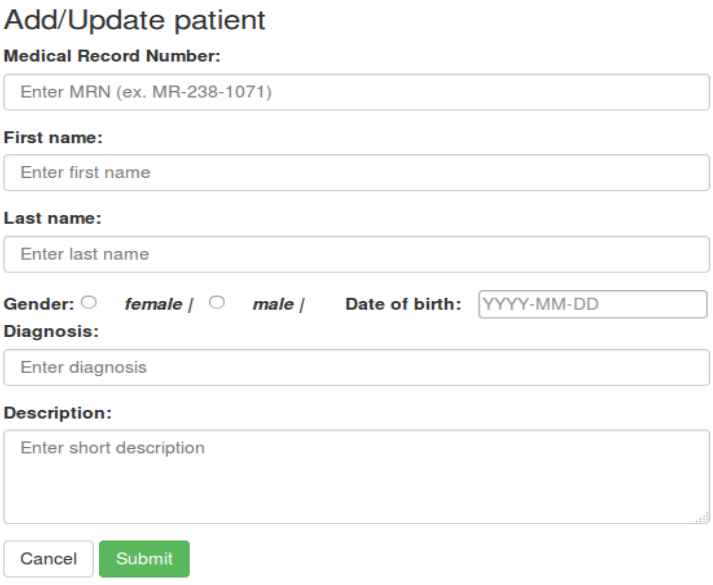




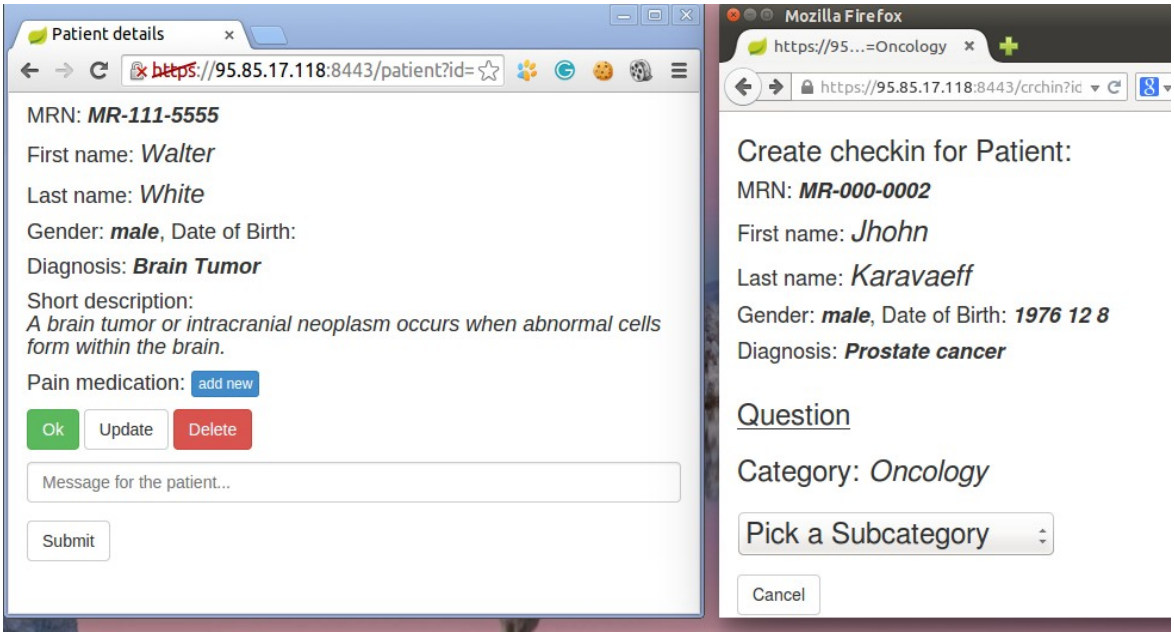
And as you've seen above one Patient can belong to many doctors. If the user knows “secret” he/she can see all patients)).

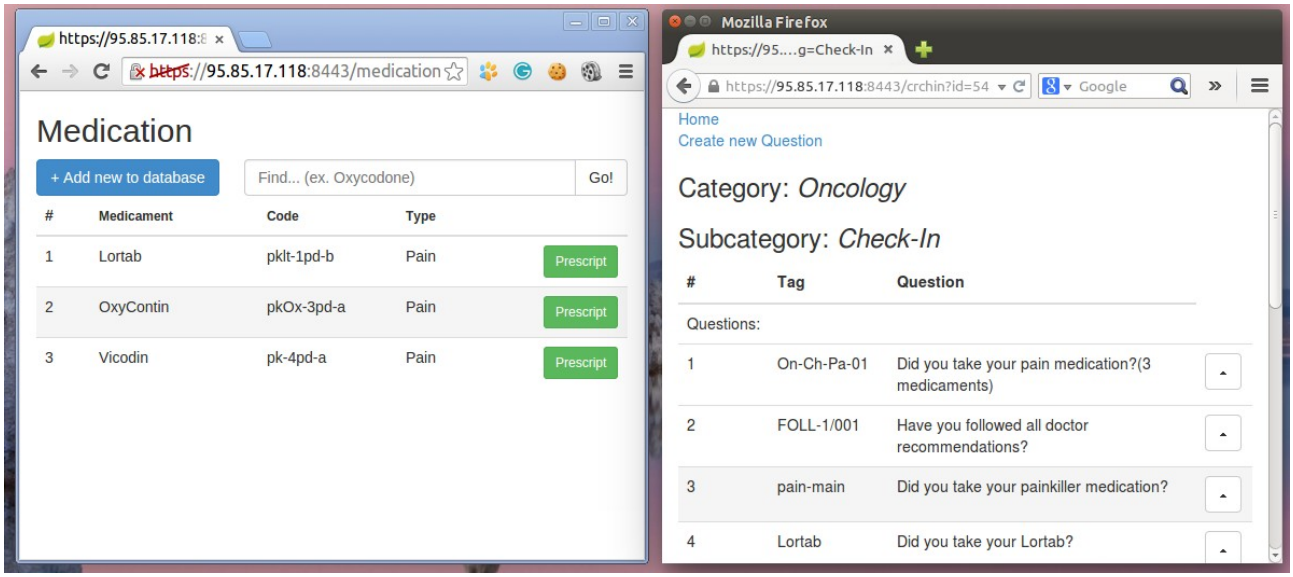


Doctor can add, update, delete Patients:

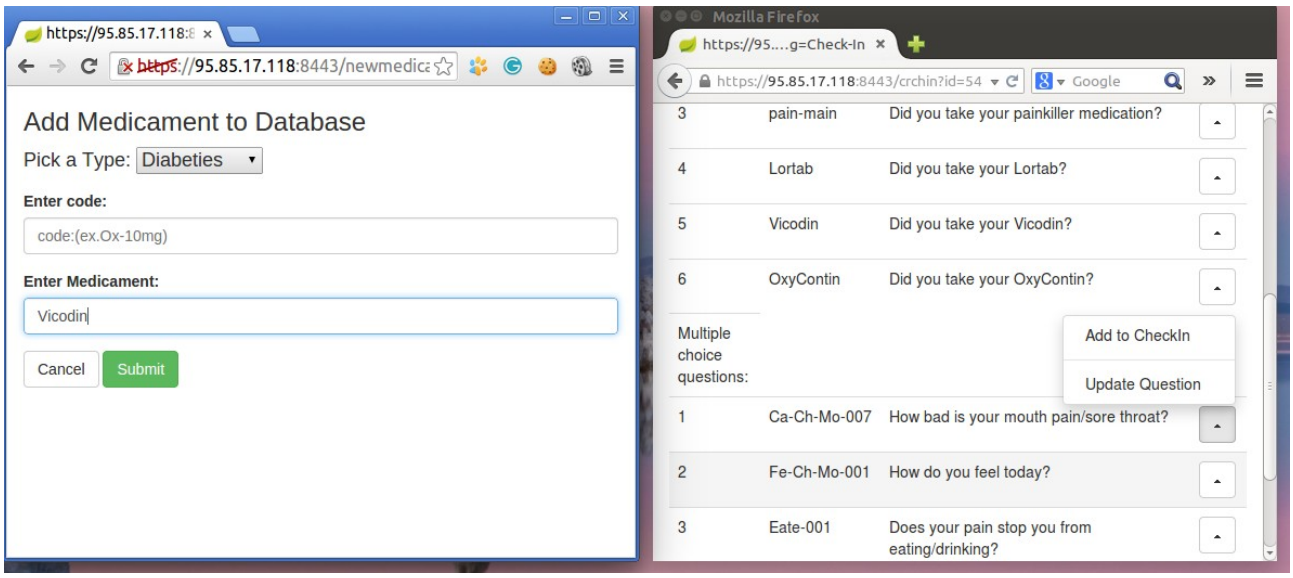


He/she can see detailed information, prescript new medication, send messages, create individual Check-Ins:





Add new questions to database, update questions, add new medication to DB:



When doctor prescripts new medication the question like “Did you take your Vicodin?” goes to the patient's Check-In automatically and immediately!

Conclusions

Utilizing a Spring and MongoDB in the project make it easy maintainable, scaling (and vertically and horizontally), pretty secure. For security purposes I do not keep any data on the mobile, only on the server side.

Mongodb allows add new data fields, change data, create very sophisticated queries without headache. I'll show how it works in the screen-cast.

System supports same user from different devices at the same moment and multiple users from many gadgets.

PS. If something won't unzip or load or … here you'll find all links and files:

https://github.com/ruan65/capstone2014

If there something is unclear don't hesitate to ask questions: ruan65@gmail.com

PSS.))