Android Capstone Project

Symptoms Management

Final Assingment

December 1, 2014

1 Introduction

The intent of this document is to outline what will be the application Symptoms Management for the Android Capstone Project.
For easier evaluation a section will be dedicated detailing how each items in the rubric list is intented to be implemented.
This project will consist on server in the cloud to process requests sent by the Android application.

2 Notes from Student

I have to apologize in advance because due to the lack of time I could not implement all the item in the rubric list. This document also is not very detailed because of this.
To further improve my skills on developing and depolying applications I chose to use a PostgreSQL database and a Tomcat server on a DigitalOcean server. IP: 178.62.10.164
The database is open for anyone who wants to connect to it using the tool pgAdmin. This can be downloaded from `http://www.pgadmin.org/download/`.
However not everyone may not want to install the software for this. The content of the database is availabe using the url `http://178.62.10.164:8080/SymptomsManagement-0.1.0/getAllData`. The website `http://jsonlint.com/` can be used to help view the data.
Only the android application has to to installed for testing purposes.

3 Android Applicaton

This application will allow for both doctors and patient perform most of the actions asked for this project.

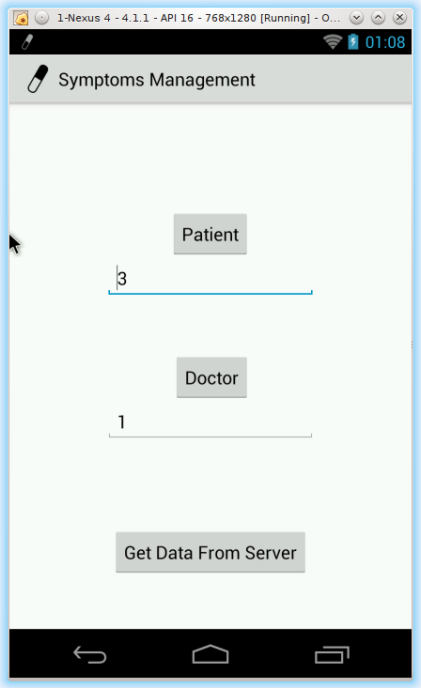

Figure 1: Main Activity

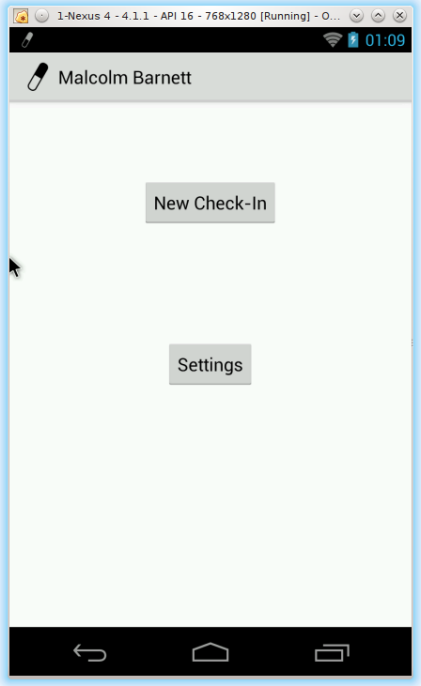

Figure 2: Patient Home Activity

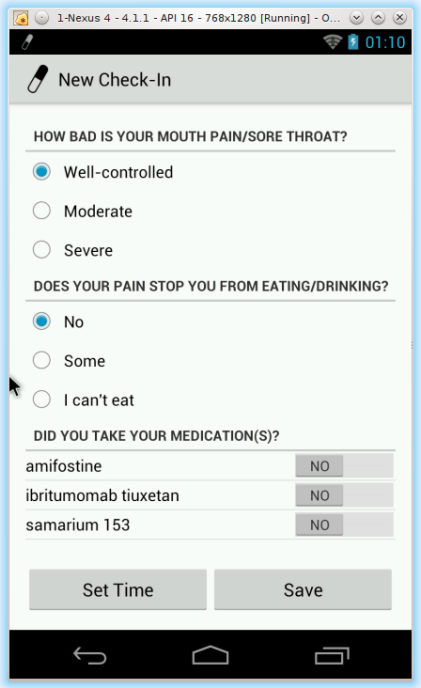

Figure 3: Patient Check-In Activity

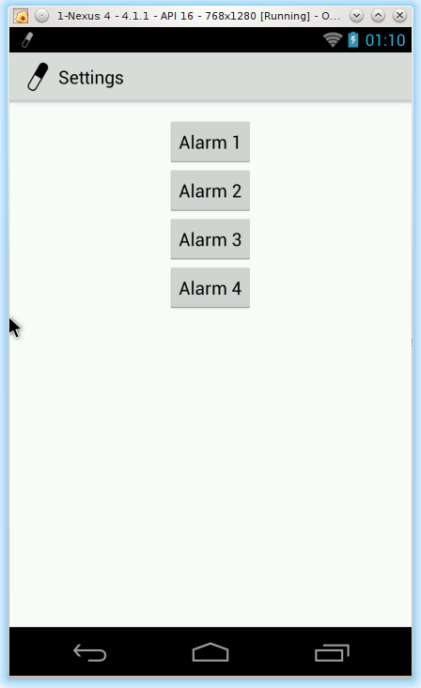

Figure 4: Patient Settings Activity

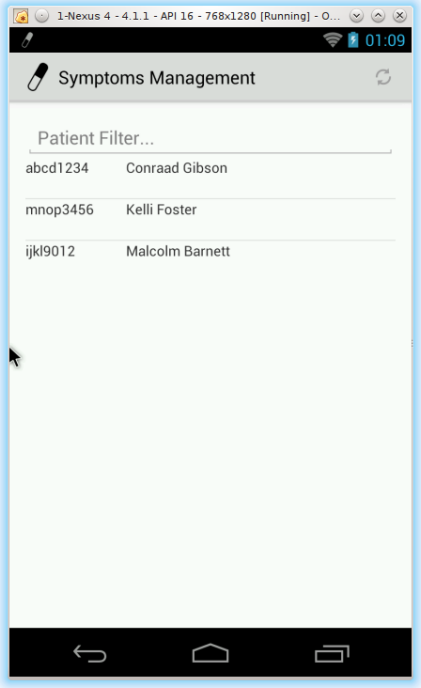

Figure 5: Patients List Activity

Figure 6: Patien Detail Activity

3.1 Database

A sql file is included in the zip package where this docuemnt was contained that is the database creation script.

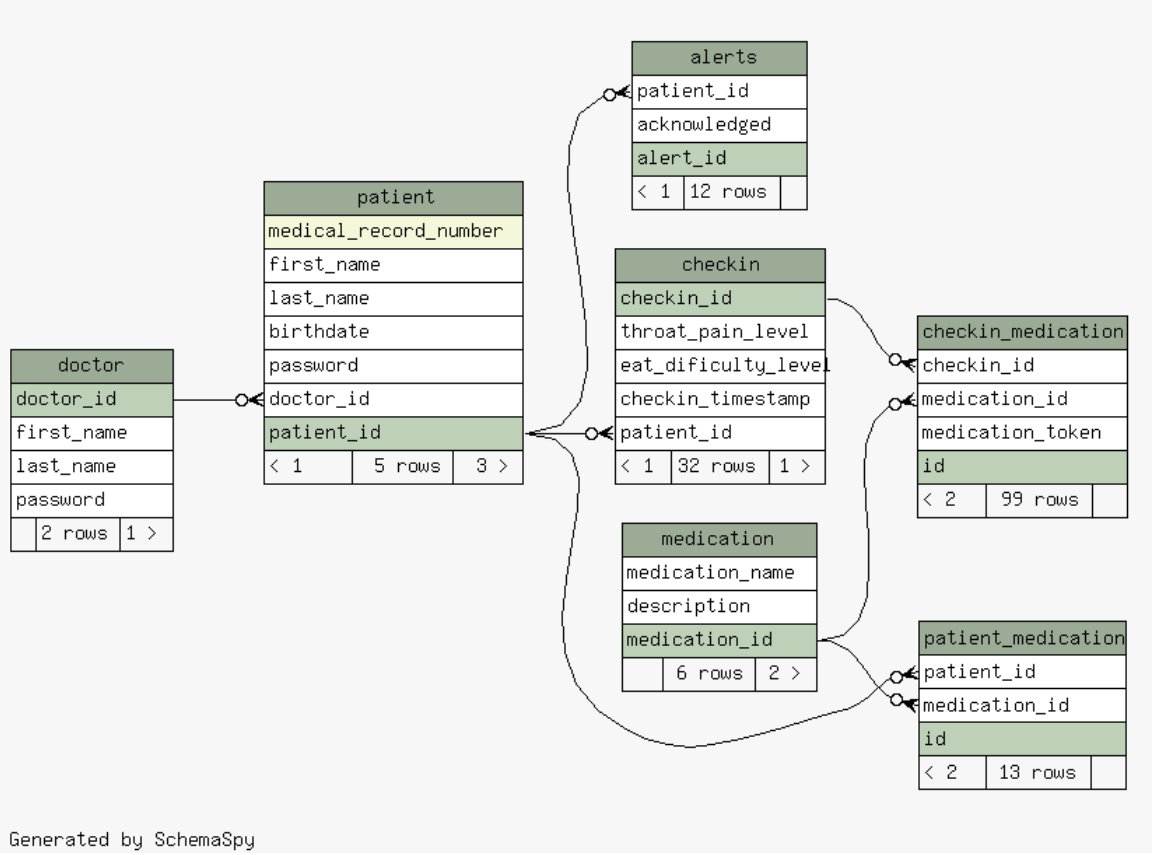

Figure 7: Database Scehma

4 Rubric List

This section is dedicated to inform how each element of the rubric list meets its requirements.

4.1 Basic Project Requirement

4.1.1 App supports multiple users via individual user accounts

Yes but login was not implemented. The are solely identified by the id on the database.

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

The login was no implemente but all actions are performed as if they were.

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

Several activities where implemented as well as Broadcast Receivers for the alarms.

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

The Spring server is hosted on the DigitalOcean server.

4.1.5 App interacts over the network via HTTP

Yes. most of the actions performed interact over the network. Add/remove medications, patient checkin, refresh patient data, etc.

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

Yes

4.1.7 App uses at least one advanced capability or API

No.

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

All actions that interact over the network are performed as background tasks.

4.2 Functional Description and App Requirement

4.2.1 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

Yes, but the login method is only by selecting the patient id on the database.

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

Yes. I defined four daily alarms that can be set on the patient settings activity (figure 4). The reminder is a notification on the notification tray.

4.2.3 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 patients responses to various questions (items 4-8) at that date and time.

Yes. Clickin the notification launches the Check-In activity for the patient that set the alarm.

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

Yes. See figure 3.

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

Yes, I decided to include the medications the patient has as a list comprised of switches (figure 3).

4.2.6 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.

Yes. As told in the previous item.

4.2.7 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.

Only the time was included for the checkin. It can be set using the button on figure 3. All medications will be associated with that time.

4.2.8 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 cant eat.

Yes. See figure 3.

4.2.9 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.

Yes, but the login method is only by selecting the doctor id on the database.

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

A doctor can only monitor the Check-Ins.

4.2.11 A doctor can search for a given Patients Check-In data by the patients name (an exact text search hosted server-side).

The patients List associated to the doctor is stored in a SQLite database on the Android application. This data is donwloaded from the server.

4.2.12 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

Yes. A doctor can add a medication using the button on the upper right corner in figure 6. The medication can be removed by long pressing the medication on the same figure.

4.2.13 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 cant eat.

A mechanism was implemented on the server side that on each patient Check-In will analyse previous Check-Ins find if the patient is experiecing problems. If so a new entry will be insert on the table alerts. In figure 5 a doctor can update the patient data from the server. If an alert is present and not acknowledged a new notification will appear and, on click, will launch the corresponding patient detail activity and acknowledge the alert on the server.

4.2.14 A patients data should only be accessed by his/her doctor(s) over HTTPS.

No.