

Portable Patient Symptom Management - Project Specification

“Portable Patient Symptom Management” is a smartphone-based application which provides a more reliable method to assess symptoms in near real-time. It provides patients a simple way to work with their healthcare providers to improve management of pain and quality of care.

Considerations:

Patient:

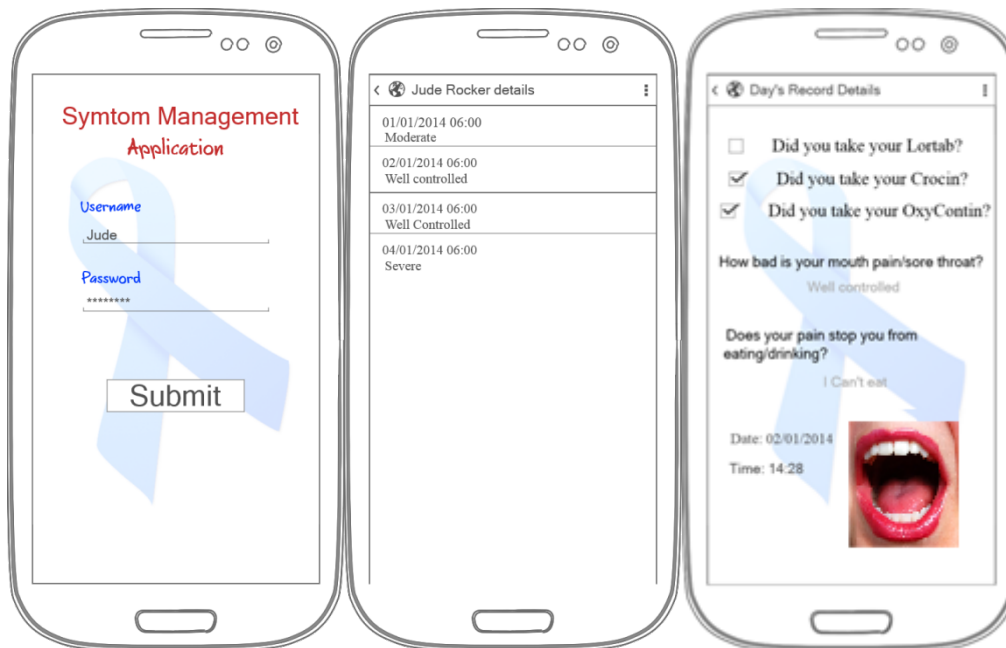
- Patient will have First name, Last name, DOB and medical number. Medical number is considered to be unique throughout the history of the patient.
- Total Duration is 7 weeks
- User will be able to login and see only if he is assigned a tracking job by the doctor.
- Primary interaction of patient with the system is through Notification service.
- Patients are Hard coded
- Login will be active till the user logs out manually or the user uninstalls the app.
- Whenever patient opens app, data is synched from the server- any new medicine is updated and details of previous check-ins are displayed.

Doctor

- Doctor sees his patient by making a server side search.
- Doctors are Hard coded
- Doctor is alerted based the hours counted backwards from the current check-in of the patient
- Doctors update the medical list of individual patient whenever he wants during the treatment

Wireframes

Workflow 1: patient login & seeing the check-in details



Workflow 2: Patient checking in after notification



Workflow 3: Patient changing notification timings



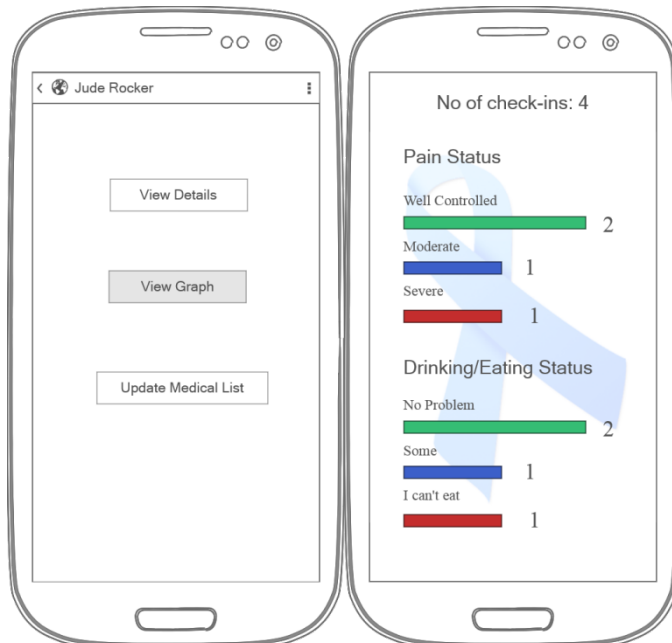
Workflow 4: Doctor logging in to check his patient status



Workflow 5: Doctor viewing the check-in details of the patient



Workflow 6: Doctor viewing graph of the patient to determine his status



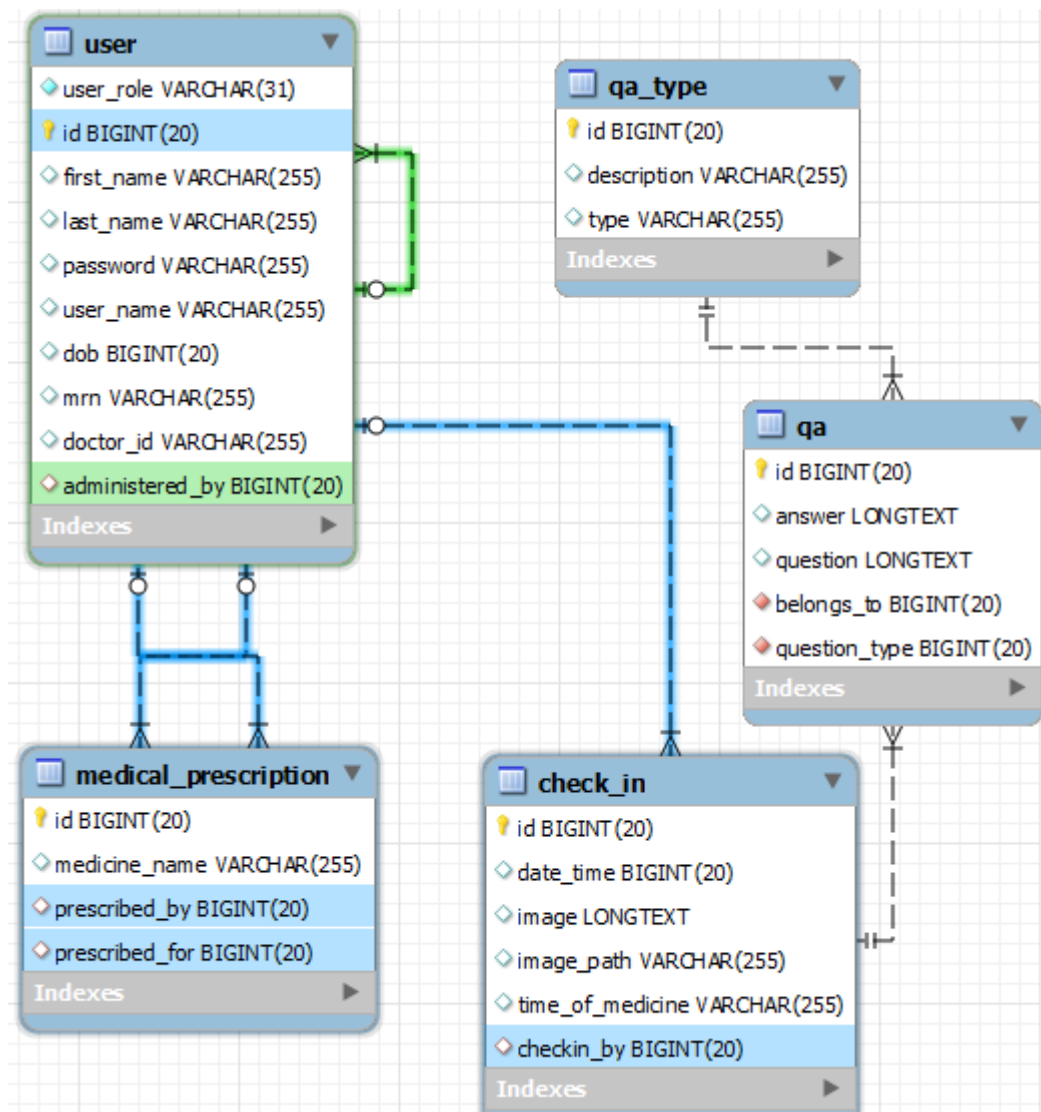
Workflow 7: Doctor updating medical list of patient



Workflow 8: Doctor receiving notification about emergency situation

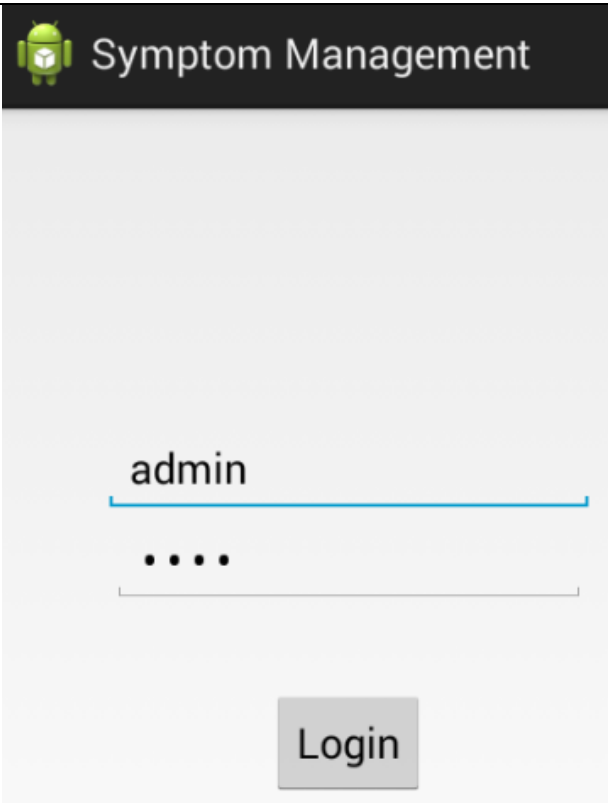
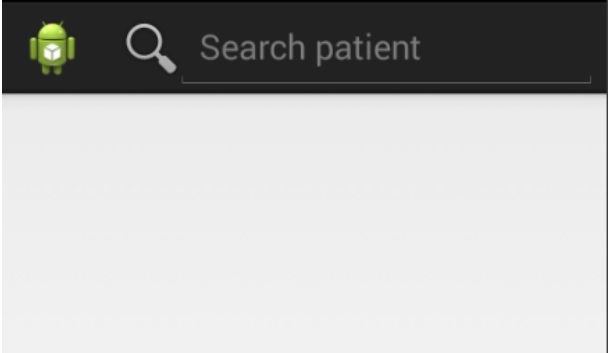


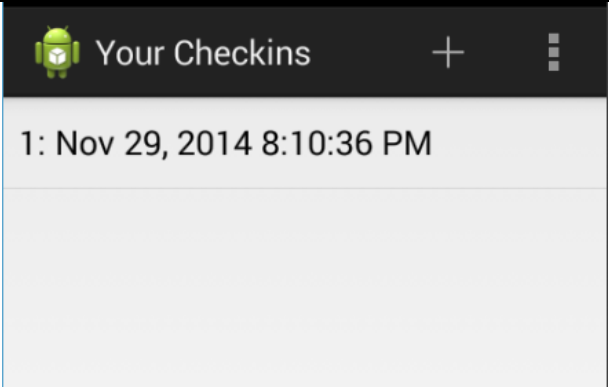
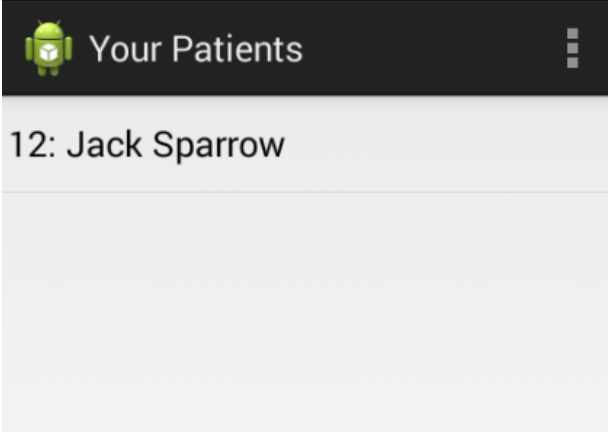
Database Design:

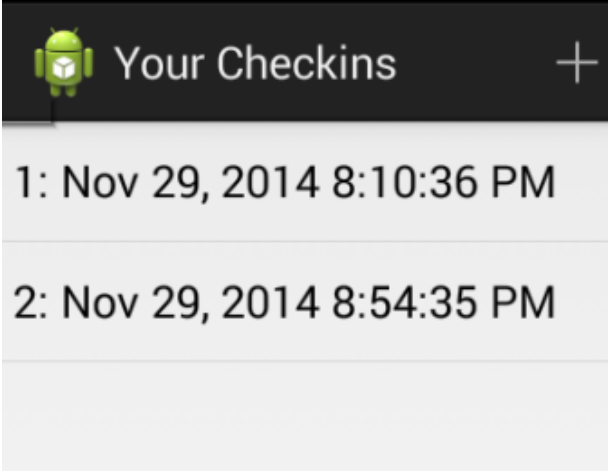
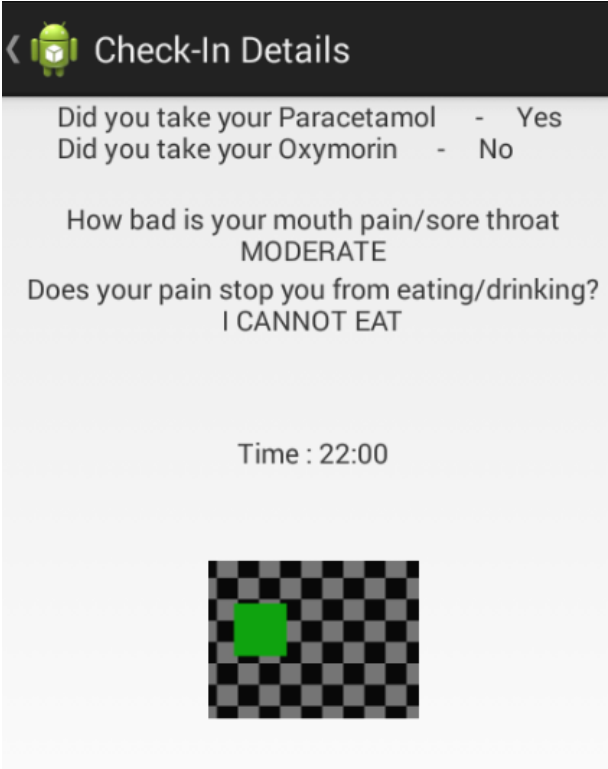





Rubric Tabular Column:

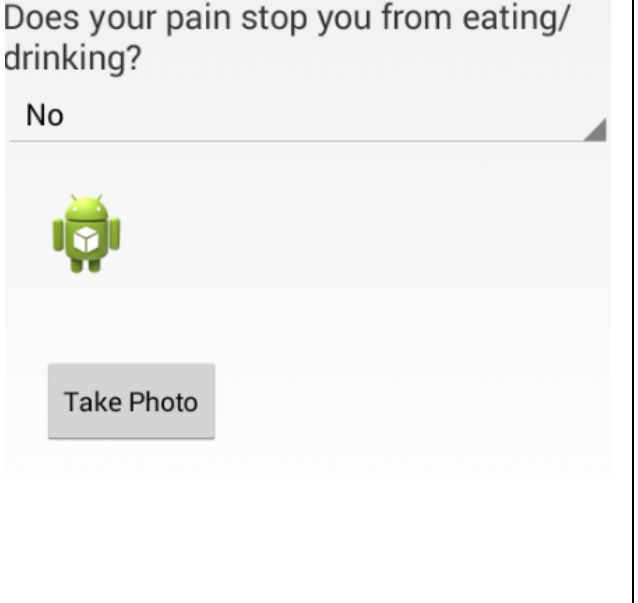
Basic Project Requirement:

Requirement	Details, code reference(<i>Italics</i>)	Screenshots
App supports multiple users via individual user accounts	<ul style="list-style-type: none"> App supports multiple users through role based system. No new registration is possible in this setup but the workflow can be easily added. <p><i>Line 201 of server: /Symptom Management/src/main/java/com/geo/sm/auth/OAuth2SecurityConfiguration.java</i></p> <ul style="list-style-type: none"> Only patients administered by a doctor will be able to use the app. The <i>user</i> table has a field called <i>administered_by</i> which tells whether a user has the permission to use the app or not. <p><i>Line 120 of mobile: /SymptomManagementApp/src/com/geo/sm/client/LoginScreenActivity.java</i></p> <ul style="list-style-type: none"> Upon login, app identifies whether the user is a patient or doctor based on the user type saved in the database and the workflow behaves accordingly <p><i>Line 95 & 159 of mobile: /SymptomManagementApp/src/com/geo/sm/client/LoginScreenActivity.java</i></p>	
App contains at least one user facing function available only to authenticated users	<ul style="list-style-type: none"> Users can use the app only if they are registered users and the login screen is the first one to appear Doctor has a search bar as the home page <p><i>Home screen : Doctor - /SymptomManagementApp/src/com/geo/sm/doctor/PatientListActivity.java</i></p> <ul style="list-style-type: none"> Patient has list of Check-Ins as the home page <p><i>Patient - /SymptomManagementApp/src/com/g</i></p>	<p>Doctor:</p>  <p>Patient:</p>

	<code>eo/sm/patient/CheckInListActivity.java</code>	
App comprises at least 1 instance of each of at least 2 of the following 4 fundamental Android components: Activity, BroadcastReceiver, Service, ContentProvider	<ul style="list-style-type: none"> Activity is primarily used for every screen in the application <p><i>All classes ending with activity are Activities. Present in com.geo.sm.client, com.geo.sm.patient, com.geo.sm.doctor</i></p> <ul style="list-style-type: none"> BroadcastReceiver along with Alarm Manager and Notification service is primarily used for Reminder settings <p><i>/SymptomManagementApp/src/com/geo/sm/client/AlarmNotificationReceiver.java</i></p>	
App interacts with at least one remotely-hosted Java Spring-based service	<p>The server is a Java Spring Boot instance</p> <p><i>Entire server code. Spring Application starts from /SymptomManagement/src/main/java/com/geo/sm/Application.java</i></p>	<pre>:: Spring Boot :: (v1.0.2.RELEASE) 2014-11-30 06:37:45.836 INFO 5568 --- [main] com.geo.sm.Application : Starting Application on Forest with PID 5568 (started \workspace-cloud\SymptomManagement)</pre>
App interacts over the network via HTTP	<ul style="list-style-type: none"> Client Android app makes all interactions with the server via HTTP through REST calls <p><i>/SymptomManagement/src/main/java/com/geo/sm/client/CheckInSvcApi.java</i></p> <p><i>/SymptomManagement/src/main/java/com/geo/sm/client/SMFuncSvcApi.java</i></p> <p><i>/SymptomManagement/src/main/java/com/geo/sm/client/UserSvcApi.java</i></p>	<pre>---> HTTP GET https://10.0.2.2:8443/patient/: 1 Authorization: Bearer 85978c49-3d6f-4eb0-9ac ---> END HTTP (no body) <--- HTTP 200 https://10.0.2.2:8443/patient/: 1 (335ms) Server: Apache-Coyote/1.1 X-Content-Type-Options: nosniff X-XSS-Protection: 1; mode=block</pre>

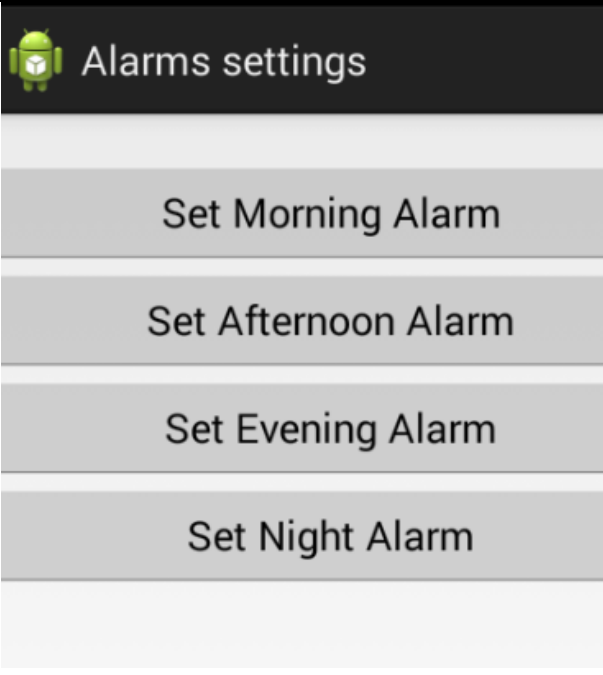
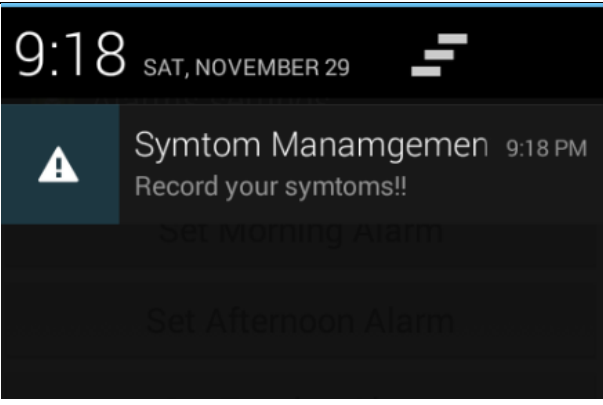
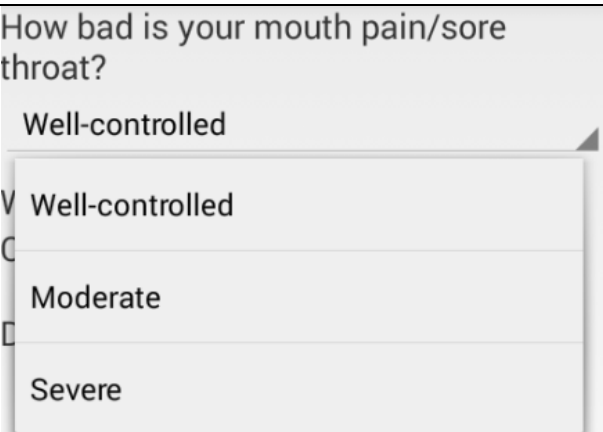
<p>App allows users to navigate between 3 or more user interface screens at runtime</p>	<p>Three primary views:</p> <ul style="list-style-type: none"> Details view to see the check-in details <p><i>/SymptomManagementApp/src/com/geosm/client/CheckInDetailsActivity.java</i></p> <ul style="list-style-type: none"> Notifications & App settings to see settings screen <p><i>Line 145 of mobile:</i></p> <p><i>/SymptomManagementApp/src/com/geosm/patient/CheckInListActivity.java</i></p> <ul style="list-style-type: none"> List view to see the check-ins on both the doctor and patient side <p><i>/SymptomManagementApp/src/com/geosm/patient/CheckInListActivity.java</i></p> <ul style="list-style-type: none"> Graph view to quickly understand the status of a patient <p><i>/SymptomManagementApp/src/com/geosm/doctor/GraphActivity.java</i></p>	<p>List view:</p>  <p>Details view:</p>  <p>Settings View:</p>
---	---	---

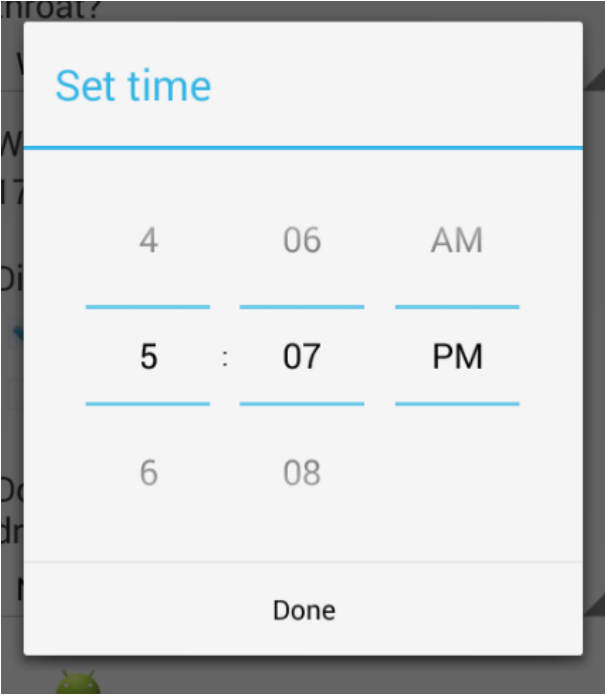
		<div><div><div><div><div>Your Checkins</div><div><div></div><div></div><div></div></div></div><div><div>1: Nov 29, 20</div><div>2: Nov 29, 20</div></div><div><div>Refresh</div><div>Logout</div><div>Alarm Settings</div></div></div><div><div><div>Alarms settings</div><div><div>Set Morning Alarm</div><div>Set Afternoon Alarm</div><div>Set Evening Alarm</div><div>Set Night Alarm</div></div></div><div><div>Graph View:</div><div><div><div>Graph</div><div>No of Check-Ins : 2</div><div>Pain Status</div><div>Well Controlled</div><div>Moderate</div><div>Severe</div></div></div></div></div></div></div>
--	--	---

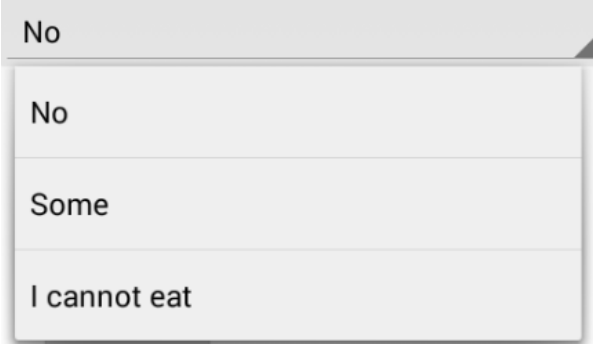
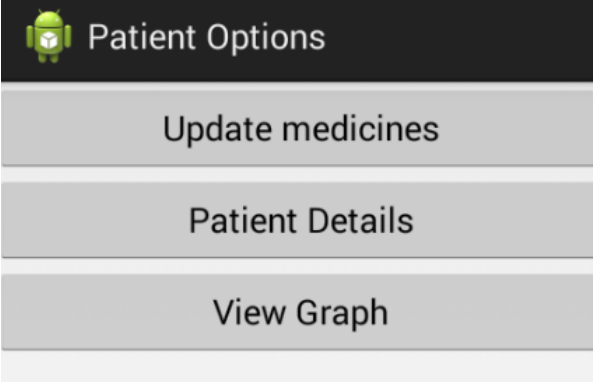
App uses at least one advanced capability or API from the following list (covered in the MoCCA Specialization): multimedia capture, multimedia playback, touch gestures, sensors, animation	<p>This application uses camera to take pictures of throat or any needed areas</p> <p><i>Line 412 to 426 of mobile :</i> <i>/SymptomManagementApp/src/com/g eo/sm/patient/CheckInFormActivity.jav a</i></p>	<p>Does your pain stop you from eating/ drinking?</p> <p>No</p> 
App supports at least one operation that is performed off the UI Thread in one or more background Threads of Thread pool.	<ul style="list-style-type: none"> All REST calls are made through the Async Task framework <p><i>Used is all fetch and add operations. Sample at Line 322 of mobile:</i> <i>/SymptomManagementApp/src/com/g eo/sm/patient/CheckInFormActivity.jav a</i></p>	<pre> licineView.setText("The check in id is " + checki UserSvcApi svc = UserSvc.getOrShowLogin(this); c != null) { llableTask.invoke(new Callable<Collection<Qa>>() @Override public Collection<Qa> call() throws Exception return svc.getQasByCheckInId(checkin_id); } new TaskCallback<Collection<Qa>>() { @Override public void success(Collection<Qa> result) { StringBuffer str = new StringBuffer(); for (Qa q : result) { </pre>

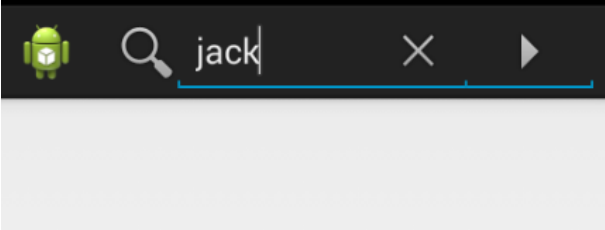
Functional Description and App Requirement:

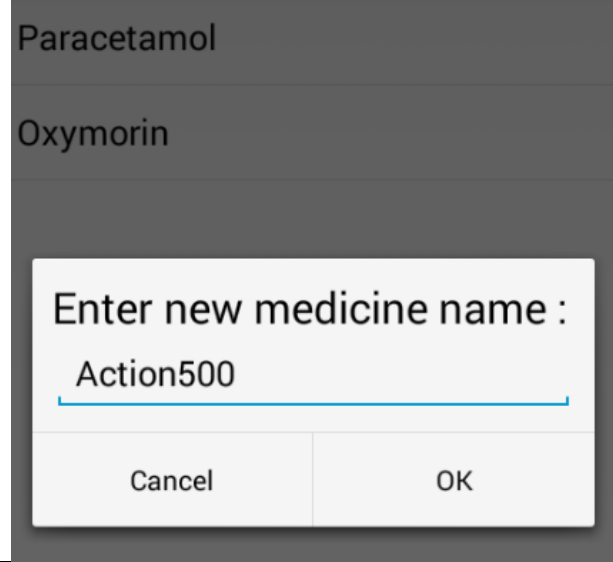
App identifies a Patient as a user with first name, last name, date of birth, a (unique) medical record number, and possibly other identifying information). A patient can login to their account.	<p>A patient is a user role with first name, last name, dob and medical number. They can login using the first page of the app</p> <p><i>Refer database schema above basic requirements table</i></p>	<ul style="list-style-type: none"> ▶ ◆ user_role ▶ ◆ id ▶ ◆ first_name ▶ ◆ last_name ▶ ◆ password ▶ ◆ user_name ▶ ◆ dob ▶ ◆ mrn ▶ ◆ doctor_id ▶ ◆ administered_by
--	--	---

<p>App defines a Reminder as an alarm or notification which can be set to patient-adjustable times (at least four times per day).</p>	<ul style="list-style-type: none"> Here there are four notifications period. Morning, afternoon, evening and night. <p><i>/SymptomManagementApp/src/com/geo/sm/client/Alarm CreateActivity.java</i></p> <ul style="list-style-type: none"> During those periods , patient can set the time at which he wishes to get notification. <p><i>Line 117 – 156 of Mobile: /SymptomManagementApp/src/com/geo/sm/client/Alarm CreateActivity.java</i></p>	
<p>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.</p>	<ul style="list-style-type: none"> When the user clicks the notification, it automatically opens the check in list page. <p><i>Line 38 – 57 of Mobile: /SymptomManagementApp/src/com/geo/sm/client/Alarm NotificationReceiver.java</i></p> <ul style="list-style-type: none"> User clicks the '+' icon to add a new entry <p><i>Line 158 of Mobile: /SymptomManagementApp/src/com/geo/sm/patient/CheckInListActivity.java</i></p>	
<p>Check-In includes the question, “How bad is your mouth pain/sore throat?” to which a patient can respond, “well-controlled,” “moderate,” or “severe.</p>	<p>Check-in involves a spinner through which the option is selected</p> <p><i>Line 21 of mobile: /SymptomManagementApp/res/layout/activity_checkin_form.xml</i></p>	

<p>Check-In includes the question, “Did you take your pain medication?” to which a Patient can respond “yes” or “no”.</p>	<p>This question is showed if only one medication is available.</p> <p><i>Line 342 mobile:</i> <code>/SymptomManagementApp/src/com/geo/sm/patient/CheckInFormActivity.java</code></p>	<p>Did you take your pain medication?</p> <p><input checked="" type="checkbox"/> Paracetamol</p> <p><input type="checkbox"/> Oxymorin</p>
<p>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.”</p>	<p>If multiple medicines are suggested then questions are iterated through all the medicines</p> <p><i>Line 342 mobile:</i> <code>/SymptomManagementApp/src/com/geo/sm/patient/CheckInFormActivity.java</code></p>	<p>Did you take your pain medication?</p> <p><input checked="" type="checkbox"/> Paracetamol</p> <p><input type="checkbox"/> Oxymorin</p>
<p>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.</p>	<p>Here since all the questions are populated in the same activity, the patient always needs to enter the date and time</p> <p><i>Line 369 of mobile:</i> <code>/SymptomManagementApp/src/com/geo/sm/patient/CheckInFormActivity.java</code></p>	<p>Time Picker</p>  <p>Chosen time:</p> <p>What time did you take medicines? 17:07</p>

<p>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.</p>	<p>This question is asked in the same activity and is again a spinner</p> <p><i>Line 64 of mobile:</i> <i>/SymptomManagementApp/res/layout/activity_checkin_form.xml</i></p>	<p>Does your pain stop you from eating/drinking?</p> 
<p>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.</p>	<p>A doctor is a separate role and doctor has first name, last name and unique id . The doctor can only search his patients which is governed by the <i>administered_by</i> field in the <i>user</i> table</p> <p><i>Refer database schema above basic requirements table</i></p>	<p>User is the super class of Patient and Doctor:</p> <pre>@Entity @Inheritance(strategy = InheritanceType.DISCRI- MINATOR) @DiscriminatorColumn(name = "user_role") public class User { @Id @GeneratedValue(strategy = GenerationType.IDENTITY) private long id; private String firstName; private String lastName; @Column(unique = true) private String userName; private String password; }</pre> <p>Unique doctor id attrib:</p> <pre>@Entity public class Doctor extends User { @Column(unique = true) private String doctorId; }</pre>
<p>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).</p>	<p>The doctor will be able to see check-in data in detail or graphically in the doctor workflow.</p> <p><i>/SymptomManagementApp/src/com/geo/sm/doctor/CheckInDetailsActivityDoctor.java</i></p> <p><i>/SymptomManagementApp/src/com/geo/sm/doctor/CheckInListActivityDoctor.java</i></p> <p><i>/SymptomManagementApp/src/com/geo/sm/doctor/GraphActivity.java</i></p>	<p>Doctor can search and get different patient views:</p>  <p>CheckIn displayed graphically:</p>

		<p>No of Check-Ins : 3</p> <p>Pain Status</p> <p>Well Controlled</p> <p>Moderate</p> <p>Severe</p> <p>Drinking/Eating status</p> <p>No Problem</p> <p>Some</p> <p>I Cannot Eat</p>
<p>A doctor can search for a given Patient's Check-In data by the patient's name (an exact text search hosted server-side).</p>	<ul style="list-style-type: none"> Primary home screen for doctor is to search for a patient <p><i>/SymptomManagementApp/src/com/geo/sm/doctor/SearchActivity.java</i></p> <ul style="list-style-type: none"> Search returns patient list with exact first name or last name <p><i>Line 58 of server: /SymptomManagement/src/main/java/com/geo/sm/UserService.java</i></p>	

<p>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.</p>	<p>Doctor can update medicines to the current patient through the app menu.</p> <p><i>/SymptomManagementApp/src/com/geo/sm/doctor/MedicineDetailsActivityDoctor.java</i></p>	
<p>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.”</p>	<p>Though not fully explored , planning to notify using GCM based on a logic in the server</p>	
<p>A patient’s data should only be accessed by his/her doctor over HTTPS.</p>	<p>All rest calls are encrypted using HTTPS by a self trusted certificate.</p> <p><i>/SymptomManagement/src/main/java/com/geo/sm/auth/OAuth2SecurityConfiguration.java</i></p>	<pre> @Override public void customize(ConfigurableEmbeddedServletContainer container) { TomcatEmbeddedServletContainerFactory tomcat = (TomcatEmbeddedServletContainerFactory) container; tomcat.addConnectorCustomizers(new TomcatConnectorCustomizer() { @Override public void customize(Connector connector) { connector.setPort(8443); connector.setSecure(true); connector.setScheme("https"); Http11NioProtocol proto = (Http11NioProtocol) connector .getProtocolHandler(); proto.setSSLEnabled(true); proto.setKeystoreFile(absoluteKeystoreFile); proto.setKeystorePass(keystorePass); proto.setKeystoreType("JKS"); proto.setKeyAlias("tomcat"); } }); } </pre>