Please take a look at the screen cast. Most of this info is covered there!

Basic Project Requirements

	Requirement	Code Reference
1	App supports multiple users via individual user accounts	1.server: **/controllers/DmController.java** 2. server: **/schema-mysql.sql** 3 client: **ui/AuthActivity.java** many more....
2	App contains at least one user facing function available only to authenticated users	1. client: any fragment/activity other than LogInFragment 2. server: any method in **/controllers/DmController.java**
3	App comprises at least 1 instance of each of at least 2 of the following 4 fundamental Android components: * Activity * BroadcastReceiver * Service * ContentProvider	1. client: **ui/MainActivity.java** 2. client: **receiver/CheckInPublisher.java** 3. client: **service/NetOpsService.java** 4. client: **provider/DmProvider.java** Many more in the client code, almost all fragment have broadcast receivers.
4	App interacts with at least one remotely-hosted Java Spring-based service	1. **submission/app/acp_server** is a remotely-hosted Java Spring-based service
5	App interacts over the network via HTTP/HTTPS	1. server: **/Application.java : customize(...) method**
6	App allows users to navigate between 3 or more user interface screens at runtime	1. client: **/ui/** contains 17 classes, most of them represent a separate screen
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.	1. client: **/ui/FeedbackFragment.java** uses gestures for the chart, most of the network operations are done via GCM (push notifications)
8	App supports at least one operation that is performed off the UI Thread in one or more background Threads of Thread pool.	1. client: **/sync/SyncAdapter.java** 2. client: **/service/NetOpsService.java** 3. client: cursor loaders in the fragments

Functional Description and App Requirement

	Requirement	Code Reference
1	The Teen is the primary user of the mobile app. A Teen is represented in the app by a unit of data containing the core set of identifying information about a diabetic adolescent, including (but not necessarily limited to) a first name, a last name, a date of birth, and a medical record number.	1. client: **/ui/SignUpFragment.java** 2. client: **/ui/SignUpTabx.java** 3. client: **/model/UserInfo.java** 4. server: **/schema-mysql.sql**
2	The Teen will receive a Reminder in the form of alarms or notifications at patient-adjustable times at least three times per day.	1. client: **/ui/RemindersFragment.java** 2. client: **/receiver/CheckInPublisher.java**
3	Once the Teen acknowledges a Reminder, the app will open for a Check-In. A Check-In is a unit of data associated with that Teen, a date, a time, and can include the user's responses to the following set of Questions at that date and time: *a lot of questions here...*	1. client: **/ui/CheckInActivity.java** 2. client: **/res/layout/activity_check_in.xml** 3. client: **/receiver/CheckInPublisher.java**
4	Feedback is the mechanism by which Check-In data is summarized and provided to the user in a meaningful way. A Teen is able to monitor their Feedback data that is updated at some appropriate interval (e.g., when a Check-In is completed, daily, weekly, or when requested by Followers). The Feedback data can be viewed graphically on the mobile device.	1. client: **/ui/FeedbackFragment.java** 2. client: **/sync/SyncAdapter.java** 3. client: **/provider/*.java**
5	A Follower is a different type of user (e.g., a parent, clinician, friend, etc.) who does not the ability to perform Check-Ins, but who can receive Check-In data shared from one or more Teens. A Teen can be a Follower for other Teens.	1. client: **/util/MiscUtils.java** info about Teen/Follower is saved during log in and used to determine the app behavior and appearance 2. client: **/ui/MainActivity.java** 3. client: **/ui/FeedbackFragment.java**
6	The Teen can choose what part(s) of their data to share with one or more Followers.	1. client: **/ui/FollowingFragment.java** 2. client: **/ui/FollowerFragment.java** 3. client: **/ui/UserSettingsDialogFragment.java**
7	Teen data should only be disseminated to authorized/authenticated Followers and accessed over HTTPS to enhance privacy and security of the data.	No data can be received via HTTP or by unauthorized users. 1. server: **/Application.java**