(https://courserahelp.zendesk.com/hc/requests/new)

Peer Assessments (https://class.coursera.org/androidcapstone-001/human_grading/)
/ Symptoms Management: Final

Help (https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://courserahelp.zendesk.com/hc)

Submission Phase

1. Do assignment ☑ (/androidcapstone-001/human_grading/view/courses/972699/assessments/8/submissions)

Evaluation Phase

2. Evaluate peers ☑ (/androidcapstone-001/human_grading/view/courses/972699/assessments/8/peerGradingSets)

Results Phase

3. See results ☑ (/androidcapstone-001/human_grading/view/courses/972699/assessments/8/results/mine)

Your effective grade is **44**

Your unadjusted grade is 44, which is simply the grade you received from your peers.

See below for details.

Develop a final project deliverable containing the following:

1. Design documentation explaining in detail how you implemented the *Symptoms Management App* specification. This documentation should be a refinement of your deliverable for Mid-Point peer assessment assignment.
2. The source code for the server-side must use Java Spring and the source code for the client-side must use Java Android. Since peer reviewers are not required to run the code to grade it, you must provide a PDF with a table that maps the project requirements from the grading rubric to specific classes and/or lines in the source code that the peer reviewer can use to help find where requirements are satisfied without having to hunt through thousands of lines of code and dozens of classes.
3. The complete project source code and associated files needed to build the project. Note that all software will ultimately need to be released in open-source form to facilitate peer assessment. Students are responsible for not including proprietary software in their solutions. Students who aren't willing to release their solutions in open-source form can still participate in the Capstone project, but their solutions won't be evaluated and they won't receive credit for passing the Capstone.
4. A screencast video showing the working project (submitted as a link to YouTube or similar site where your video can be watched).

Items 1, 2, and 3 should be included as a single ZIP file.
Item 4 should be submitted separately as a link pointing to where the video can be viewed.

To pass the Capstone MOOC, the average of all the Assessor reviews must be >= 70%

Title your app -- if you were to present your app to someone or describe it on your resume, what would you call it?

SymplyTracked

Submit project deliverable items 1, 2, and 3 as a single ZIP file:

1. Design documentation explaining in detail how you implemented the *Symptoms Management App* specification. This documentation should be a refinement of your deliverable for Mid-Point peer assessment assignment.
2. The source code for the server-side must use Java Spring and the source code for the client-side must use Java Android. Since peer reviewers are not required to run the code to grade it, you must provide a PDF with a table that maps the project requirements from the grading rubric to specific classes and/or lines in the source code that the peer reviewer can use to help find where requirements are satisfied without having to hunt through thousands of lines of code and dozens of classes.
3. The complete project source code and associated files needed to build the project. Note that all software will ultimately need to be released in open-source form to facilitate peer assessment. Students are responsible for not including proprietary software in their solutions. Students who aren't willing to release their solutions in open-source form can still participate in the Capstone project, but their solutions won't be evaluated and they won't receive credit for passing the Capstone.

Symptom Management Final (https://s3.amazonaws.com/coursera-uploads/user-c5536711d9cf0474c05074c4/972699/asst-8/9c3c6de076bd11e48a811d810a42adb8.zip)

Submit project deliverable item 4: a link to your screencast video showing the working project.

These links are both to the **SAME** video just two different ways to view them.

http://www.screencast.com/t/kbIgoBUst (http://www.screencast.com/t/kbIgoBUst)

https://www.dropbox.com/s/olohgpmsh3qbzdc/symptom%20management%20final%2011-30.mp4?dl=0 (https://www.dropbox.com/s/olohgpmsh3qbzdc/symptom%20management%20final%2011-30.mp4?dl=0)

Overall evaluation/feedback

Note: this section can only be filled out during the evaluation phase.

1. Basic Project Requirement:
App supports multiple users via individual user accounts

Rubric:
0-Needs Improvement: Does not support multiple users
1-Partially Complete: Supports multiple users via individual accounts, but cap on number of users is too small
1-Partially Complete: Supports multiple users, but without individual accounts
2-Complete: Supports multiple users via individual accounts (cap on number of users is reasonable)

Score from your peers: **2**

2. Basic Project Requirement:
App contains at least one user facing function available only to authenticated users

Rubric:
0-Needs Improvement: No user facing function requires authenticated user account
1-Partially Complete
2-Complete: Contains one or more user facing functions requiring authenticated user account

Score from your peers: **2**

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

- Activity
- BroadcastReceiver
- Service
- ContentProvider

Rubric:
0-Needs Improvement: Uses no components
1-Partially Complete: Uses at least one instance of only one component
2-Complete: Uses at least one instance each of at least 2 components

Score from your peers: **2**

4. Basic Project Requirement:
App interacts with at least one remotely-hosted Java Spring-based service

Rubric:
0-Needs Improvement: App does not interact with a Java Spring-based service
1-Partially Complete
2-Complete: App interacts with 1 or more remotely-hosted Java Spring-based service(s)

Score from your peers: **2**

5. Basic Project Requirement:
App interacts over the network via HTTP

Rubric:
0-Needs Improvement: App does not interact over the network via HTTP
1-Partially Compete
2-Complete: App interacts over the network via HTTP

Score from your peers: **2**

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

Rubric:
0-Needs Improvement: Users can only access 1 (or no) interface screens at runtime
1-Partially Complete: Users can navigate between only 2 user interface screens at runtime
2-Complete: Users can navigate between 3 or more user interface screens at runtime

Score from your peers: **2**

7. Basic Project Requirement:
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.

Rubric:
0-Needs Improvement: App does not use an advanced capability or API from the MoCCA list
1-Partially Complete
2-Complete: App uses at least one advanced capability or API from the MoCCA list above

Score from your peers: **2**

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

Rubric:
0-Needs Improvement: The app does not support any operations off the UI Thread
2-Complete: The app does support at least one operation off the UI Thread

Score from your peers: **2**

Provide your peer with constructive feedback about their **Basic Project** components.

peer 1 → *[This area was left blank by the evaluator.]*

peer 2 → All requirements have been met.

peer 3 → *[This area was left blank by the evaluator.]*

peer 4 → *[This area was left blank by the evaluator.]*

peer 5 → *[This area was left blank by the evaluator.]*

peer 6 → 2

peer 7 → Awesome!

1. 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.

Rubric:
0-Needs Improvement: Patient is not defined.
0-Needs Improvement: Missing both complete Patient information and patient cannot login.
1-Partially Complete: Patient has a (unique) medical record number, a first name, a last name, and a date of birth; but patient can't login
1-Partially Complete: Patient can login but the Patient associated information is missing one or more of the following: first name, last name, date of birth
2-Complete: Patient can log in to their Patient account associated with a (unique) medical record number, a first name, a last name, dob

Score from your peers: **2**

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

Rubric:
0-Needs Improvement: Reminders are not defined
0-Needs Improvement: Patient cannot set the time of a reminder but the app doesn’t support at least four per day
1-Partially Complete: Patient cannot adjust times, but app supports at least four reminders per day
2-Complete: Patient can set the time of a reminder and the app supports at least four reminders per day

Score from your peers: **2**

3. Functional Description and App Requirement:
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 (items 4-8) at that date and time.

Rubric:
0-Needs Improvement: Reminder does not trigger Check-In; associated data does not include (at least) a Patient, date, time, & patient responses
1-Partially Complete: Reminder triggers Check-In; associated data does not include (at least) a Patient, a date, a time, and patient responses
1-Partially Complete: A Reminder does not trigger a Check-In, but a Check-In has associated data including (at least) a Patient, a date, a time, and patient responses
2-Complete: A Reminder triggers a Check-In and a Check-In has associated data including (at least) a Patient, a date, a time, and that patient's responses

Score from your peers: **2**

4. Functional Description and App Requirement:
Check-In includes the question, “How bad is your mouth pain/sore throat?” to which a patient can respond, “well-controlled,” “moderate,” or “severe.

Rubric:
0-Needs Improvement: Question is not asked or not asked with defined answers
0-Needs Improvement: Question is not asked with defined answers and is not associated with Check-In
1-Partially Complete: Question is asked with defined answers, but not associated with the Check-In
2-Complete: Question is asked with defined answers and answer is associated with Check-In

Score from your peers: **2**

5. Functional Description and App Requirement:
Check-In includes the question, “Did you take your pain medication?” to which a Patient can respond “yes” or “no”.

Rubric:
0-Needs Improvement: Question is not asked or not asked with defined answers
0-Needs Improvement: Question is not asked with defined answers and is not associated with Check-In
1-Partially Complete: Question is asked with defined answers, but not associated with Check-In
2-Complete: Question is asked with defined answers and answer is associated with Check-In

Score from your peers: **2**

6. Functional Description and App Requirement:
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.”

Rubric
0-Needs Improvement: The question is not asked at all
0-Needs Improvement: Neither all of the appropriate questions are asked with defined answers, nor are the answers associated with the Check-In
1-Partially Complete: All of the appropriate questions (based on associated medications) are asked with defined answers, but not associated wi
1-Partially Complete: Only some of the appropriate questions are asked with defined answers
2-All of the appropriate questions (based on associated medications) are asked with defined answers, and associated with the Check-In

Score from your peers: **2**

7. Functional Description and App Requirement:
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.

Rubric:
0-Needs Improvement: If a patient indicates he or she has taken a pain medication he or she is not prompted to enter the time and date
0-Needs Improvement: The patient is prompted for some but not all needed info about taken medication & not associated correctly at Check-In
1-Partially Complete: The patient is prompted to enter pain medication information, but not the time and date
1-Partially Complete: The patient is prompted to enter the time and date medication was taken, but it is not associated correctly in Check-In
2-Complete: The patient isprompted to enter the time and date and that data is associated with the appropriate pain medication in the Check-In

Score from your peers: **2**

8. Functional Description and App Requirement:
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.

Rubric:
0-Needs Improvement: Question is not asked
0-Needs Improvement: Question is not asked with defined answers and is not associated with Check-In
1-Partially Complete: Question is not asked with defined answers, but answer is associated with Check-In
1-Partially Complete: Question is asked with defined answers, but answer is not associated with Check-In
2-Complete: Question is asked with defined answers and answer is associated with Check-In

Score from your peers: **2**

9. Functional Description and App Requirement:
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.

Rubric:
0-Needs Improvement: Doctor is not defined
0-Needs Improvement: Doctor does not contain all required information and cannot view a list of Patients and cannot login
1-Partially Complete: Doctor is missing 1 or 2 of the following 3 requirements: has all required info; can view list of Patients; can login
2-Complete: All of the above. Doctor contains all required information and can view a list of Patients and can login

Score from your peers: **2**

10. Functional Description and App Requirement:
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).

Rubric:
0-Needs Improvement: Doctor cannot monitor Check-Ins
1-Partially Complete: Doctor can monitor Check-Ins, but not information is not displayed graphically
1-Partially Complete: Doctor can monitor Check-Ins and information is displayed graphically, but data is not updated at appropriate interval
2-Complete: Doctor can monitor Check-Ins & info is displayed graphically. Data is updated at some appropriate interval (ex. after Check-In)

Score from your peers: **2**

11. Functional Description and App Requirement:
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…”)

Rubric:
0-Needs Improvement: A doctor cannot search for a given Patient’s Check-In data by the patient’s name
1-Partially Complete: A doctor can search for Check-In data by the patient’s name (an exact text search) but it is hosted client side
2-Complete: A doctor can search for a given Patient’s Check-In data by the patient’s name (an exact text search hosted server-side)

Score from your peers: **2**

12. Functional Description and App Requirement:
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).

Rubric:
0-Needs Improvement: A doctor cannot update a list pain medications associated with a Patient
1-Partially Complete: A doctor can update a Patient's list of pain medications, but it doesn’t update the tailored questions listed above (6)
2-Complete: A doctor can update a Patient's list of pain medications. This data updates the tailored questions listed above (6)

Score from your peers: **2**

13. Functional Description and App Requirement:
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.”

Rubric:
0-Needs Improvement: No doctor alerts are provided for any of these scenarios
1-Partially Complete: At least one of these scenarios is not supported or not supported according to the specifications
2-Complete: A doctor is alerted if a patient experiences all listed scenarios

Score from your peers: **2**

14. Functional Description and App Requirement:
A patient’s data should only be accessed by his/her doctor(s) over HTTPS.

Rubric:
0-Needs Improvement: A doctor’s access of patient data does not occur over HTTPS (or Doctor is not implemented to allow patient data access).
2-Complete: Complete: A patient’s data should only be accessed by his/her doctor over HTTPS

Score from your peers: **2**

Provide your peer with constructive feedback about their **Functional Description and App** components.

peer 1 → *[This area was left blank by the evaluator.]*

peer 2 → Regarding #2 - if I got it right from your design document, you decided not to implement "at least four reminders per day" part. It's kinda against the requirement (though I would personally agree that this restriction doesn't bring much value), however I'm still giving you full credit for this. Regarding #12 - I feel like it would be nice to give doctors a way not only to add medications, but to remove them as well.

peer 3 → Your functional description is very complete and covers all the project aspects

peer 4 → *[This area was left blank by the evaluator.]*

peer 5 → *[This area was left blank by the evaluator.]*

peer 6 → *[This area was left blank by the evaluator.]*

peer 7 → Fantabulous!

Provide your peer with constructive feedback about any **Implementation Considerations** they included in their app that you thought were particularly interesting or well implemented, or any improvements they may want to make in those implementations, and why.

peer 1 → *[This area was left blank by the evaluator.]*

peer 2 → The UI is really awesome, with all those icons, and colors, and graphs. I also like that you have managed to incorporate all four Android fundamental components.

peer 3 → I have nothing to add to your implementation, I think you have covered all the requirements in it. What I have liked the most is how you have used the sync adapter in the application and the oauth. Only one consideration, other alternative to server pull could have been server push, using google cloud messaging, in order to send a notification to a doctor, when a patient completes a check-in and and one of the requirements is fulfilled, but is only an alternative, I think your solution is very good.

peer 4 → *[This area was left blank by the evaluator.]*

peer 5 → *[This area was left blank by the evaluator.]*

peer 6 → *[This area was left blank by the evaluator.]*

peer 7 → Really beautiful work! The UI looks amazing!

Provide your peer with constructive feedback on their **project overall.**

peer 1 → It's by far the best project I have reviewed. I don't think I could give any valuable feedback.

peer 2 → Very impressive overall. Hands down this is a great submission and it does really deserve to be nominated for prizes.

peer 3 → Impressive job! I have nothing to add, the documentation and demo are very complete, and the additional features are very good, I think that with your work you have completed and surpassed all the requirements and that your implementation could be very helpful to doctors and patients.

peer 4 → Great work!!!! Congratulations!!!

peer 5 → Well without any doubt one of the top 10 projects. I have nothing to say, your app fill all requirements, the UI looks awesome. Simple and smooth app. I only have one doubt, when a doctor searches a patient by name, I think you are filtering any patient, but you should get only the own patients of that doctor. Good luck.

peer 6 → Great Job

peer 7 → Best project I reviewed by an order of magnitude!