Specification for Next Sprint

User Stories

(a) As a radiologist, I need to be able to communicate with the field worker, so that any queries, issues or misunderstandings related to patient's diagnosis can be answered quickly.
(b) As a radiologist, I want to be able to access and list all the diagnosis done by me, so that I can follow up with the patient's medical condition, if need be.
(c) As a fieldworker, I need to be able to receive feedback or treatment recommendations for a specific patient from the radiologists so that I can make proper treatment decisions.
(d) As a manager, I require a way to check if the field worker has received the diagnosis, so that we measure the effectiveness of our system.

Justification

As of the current sprint, we have designed the interface for both the Android application and the URMH client, and have added most of the functionality of the Android application, and included basic functionality of the URMH client. In the upcoming sprint, we aim to add the remaining required functionality to both the application and client. This includes enabling communication between the application and the client, and allowing the users to check the status and history of some of the content being communicated.

Our primary focus of this project is to develop a basis for a two-way communication between fieldworkers using the Android application and radiologists using the URMH client. We aim to complete this goal by the end of the next sprint. Thus far, we have been able to establish a one-way communication between the fieldworker and radiologist. Using the Android application, fieldworkers are able to insert information into a database, and radiologists are able to retrieve this information using the URMH client. For our final sprint, we wanted to allow radiologists to respond to this information, and we want to be able to relay this response back to the fieldworker. In order to do this, we must implement user story (a) which will allow radiologists to submit responses, user story (c) to relay this response back to the fieldworker, and (d) to check whether or not this process was successfully completed. In order to better manage these responses, we also decided to implement user story (b), which will allow radiologists to access their previous responses. Once we successfully implement these use stories, we would have finally completed the main objective of our project.

Ordering & Contingency Plan

User story (a) should be completed before any of the other stories allocated to this sprint, since that will allow us to populate our database with responses that are required for completing the subsequent user stories. After completing (a), we can work on either (b) or (c), however (c) should be completed before we can begin (d), since in order to assess whether or not a response has been received, it should first be possible to receive a response.

Since (a) should be completed before implementing the remaining user stories, and (c) should be completed before implementing (d), taking longer-than-expected times to complete these stories may hinder the progress of other stories. In order to allow team members to work on user stories that depend on (a) or (c), if implementing either of these stories takes a long time, then we can add mock entries into our database that takes the form of the expected insertions from (a) and (c).

If one of our members completes their assigned tasks early, they are encouraged to help other members with their tasks, and if no assistance is required, members can take stories from the product backlog and work on those.

Goals & Success Metrics

- Successfully implement selected user stories.
- Check that the responses provided from the radiologist are properly added to the database, and that information is properly received by fieldworkers.
- Check that radiologists are able to see all of the responses they have previously submitted, and that these responses only pertain to them (and not other users).
- Check that we are able to determine whether a fieldworker has successfully received a response, and that it is displayed correctly in real-time.
- Create a clear, simple, and intuitive interface for both the Android application and the URMH client.