Peer Testing Project Report

Demo Explanation and Setup Summary (Daniil)
In the demo and in our testing we created a two server setup, one server to hold un-anonymized DICOM images and one server to hold anonymized DICOM images. When the project is implemented the un-anonymized server will communicate directly with the imaging modalities to receive scans. Currently, we act as an imaging modality and send a DICOM image via the DIMSE protocol. Upon reception, the DICOM image gets stored using the DIMSE protocol C-STORE command. Once stored the un-anonymized DICOM images can be queried. Once stored this also kicks off another function that sends a copy of the DICOM image to the anonymized server. Upon reception, the anonymized server performs anonymization. These two servers are connected together on the same network and communicate with each other via the DIMSE protocol. This is the current setup/pipeline and what was shown in the demo.

Checked that images can work (Kshitij)
We uploaded images to the server and saw them process all the way. It did anonymize the specific tags that we have mentioned in the code at the moment. We will be adding more tags as we go along with the project.

Industry-standard (Kshitij)
According to the industry standards we need a functional pipeline for x-ray images to provide mammography scans to experts to analyze using AI for Cancer prediction of the breast tissue. Currently our project does meet the in progress industry standards set for the pipeline we are trying to build.

What part of the system is most dangerous (Kshitij)
During testing the most dangerous part of the system that we discovered is the sending of un-anonymized DICOM images. The issue being t

Add more unit tests (Kshitij)
There are several unit tests that we must continue to add such as checking the anonymity of data, as well as correctness of data compared to the original un-anonymized DICOM file. These will be one of the first implementations for unit tests, since they are crucial to the integrity of the files as the current set of features allow An un-anonymized PACS server has the ability to store DICOM images, create a copy, and automatically send/pipeline the images to the anonymized server

What parts of the code to test (Daniil)
The most important part of the codes to test would be the anonymization process as we need to check that for edge cases to make sure that the software still works and doesn't break even if there is an error caused by incorrect type of information or incorrect information itself

The current set of features (Daniil)

- An Un-anonymized PACS server has the ability to store DICOM images, create a copy, and automatically send/pipeline the images to the anonymized server.
- The anonymized server has the ability to automatically anonymize incoming DICOM images.
- Both servers are on the same network and can communicate together.
- Both servers can be queried.
- Can upload DICOM images to PACS server

Mention activities that we did during the week (Jace)

We finished the Peer Testing Report
We worked on Testing by uploading the images on the DICOM Server and followed the anonymization process step by step
Discussed how the original server interacts with the new exported data.
We also worked on the Weekly Report Document and made some more progress with anonymization coding

Issues discovered during tests (Jace)

- tested both servers
- try-catch failure
- anonymization failure
- patient id mismatch
- If certain things lacked a data and read a null, export fails

Prioritization (Jace)

- Finishing the code for all the tags
- Testing the code thoroughly for edge cases and making sure that the code doesn't crash
- Finishing all the documentation for the code in time
- Delivering the code in time with all the functionality completely
- Complete the function that inserts crosswalk identification values
- Implement all the main features discussed by PM