Project: Add PACS communication

Timezone: Indian Standard Time (UTC +5:30)

Project Description

Add PACS communication for InVesalius

InVesalius currently does not support loading DICOM images from a PACS server. The idea is to have a way to search for images in a PACS, download the images and load the images.

Deliverables:

- Tool to communicate with PACS server
- Save the surfaces and masks in the PACS
- Invesalius Dicom Conformance statement (Optional)

Mentor: Thiago Franco de Moraes

Synopsis

InVesalius is an open source computer tomography and magnetic resonance images reconstruction software. InVesalius accepts Dicom, Analyze 7.5, NifT1 1, PAR/REC, TIFF, BMP, JPG and PNG formats as inputs to import images to the software. These images have to be imported from the local hard drive or networked drive or removable drives.

In clinical settings most of the images are generated as Dicom images/studies on the modalities (CT,MRI,Xray machines). In clinical settings Dicom images are transferred directly from the modalities to the radiology workstation/PACS using the Dicom networking protocol over the LAN.

InVesalius as of now does not have the ability to receive Dicom images from the modalities/PACS/radiology workstations nor does it have the ability to send back Dicom Images generated(surfaces/masks) to radiology workstations or PACS.

This proposal is a humble attempt to make InVesalius a Dicom 3.0 Networking compliant with ability to send and receive Dicom images along with ability to query other Dicom nodes and retrieve studies/images.

Benefits to Community

InVesalius has been serving the community since 2001 and supports Linux, Windows and Mac operating systems. InVesalius has been widely deployed in the research and student community. InVesalius is one of the most versatile multi platform open source platforms with capabilities which can match commercial software. Due to the lack of Dicom Networking capabilities deployment in clinical/hospital settings has been slow. This project will be able to close the gap and help InVesalius gain wider acceptability and get more traction in clinical settings.

Technical Proposal

Digital Imaging and Communications in Medicine (DICOM) is the standard for the communication and management of medical imaging information and is used to store, exchange, and transmit medical images.

This project can be divided into 4 logical units.

1. InVesalius Storage Service Class Provider (receives Dicom images from other Dicom Nodes)

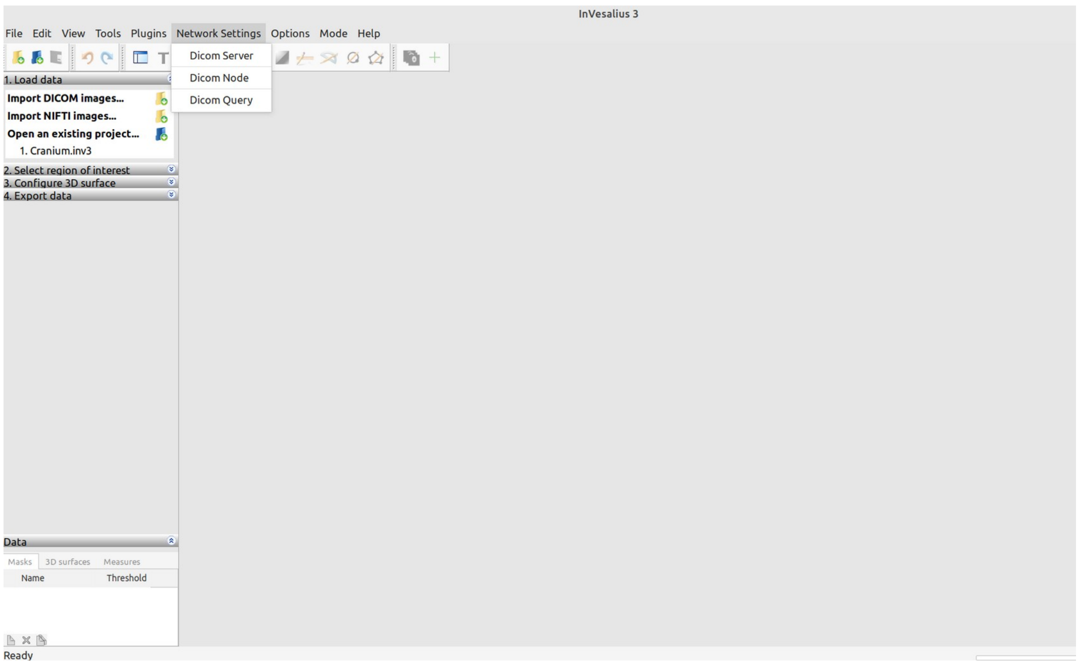
2. InVesalius Service Class User (sends Dicom images, in our case surfaces and masks using C-STORE)

3. InVesalius DICOM Query/Retrieve service (uses C-Find to find list of studies from Dicom Node and download studies using C-Move)

4. InVesalius Dicom conformance statement and testing with DVTK (Optional)

DICOM is based on two-way communication, which means that there is always one device that sends and one that receives. Or in DICOM terminology, one device that invokes an operation and the other one that performs it. In DICOM these roles are called Service Class User (SCU) and Service Class Provider (SCP).

It is proposed as shown in the figure below to add the following new drop down in the menu called Network Settings.



The Network Setting will have 3 Options

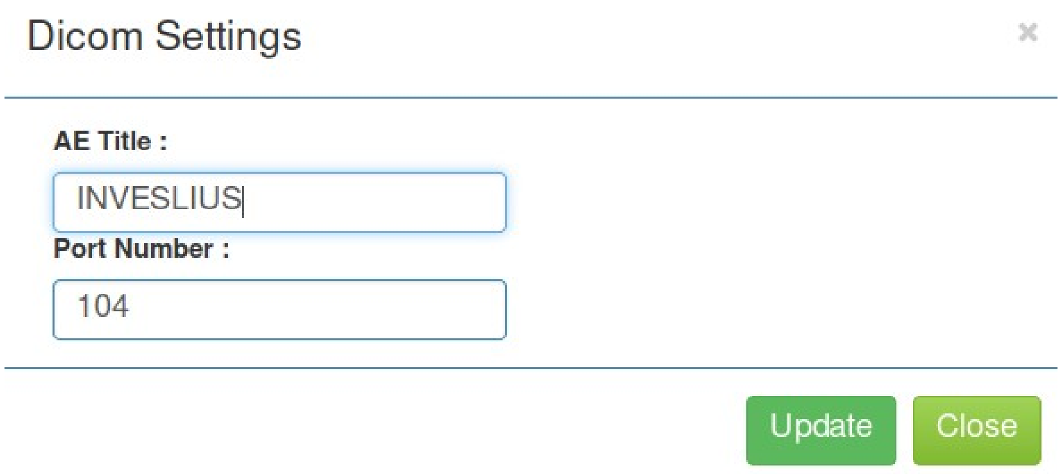
1.Dicom Server

2.Dicom Node

3.Dicom Query

Dicom Server

On clicking of Dicom server the following modal is launched as shown in the below sample figure which will be implemented in wxwidgets.


Dicom Settings

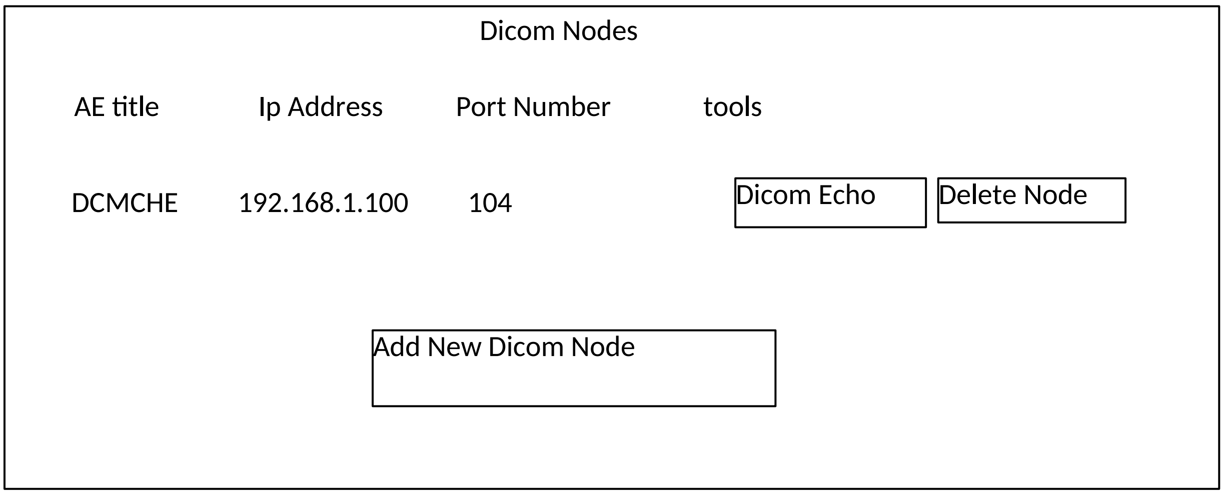
AE Title : INVESLIUS

Port Number : 104

This will be used to save and update the Application Entity(AE) title for SCP along with the port number. This will be described in more detail in the Storage SCP section.

Dicom Node

On clicking the Dicom Node option a model will opened where all previously added Dicom nodes are listed and new Dicom Nodes can be added.

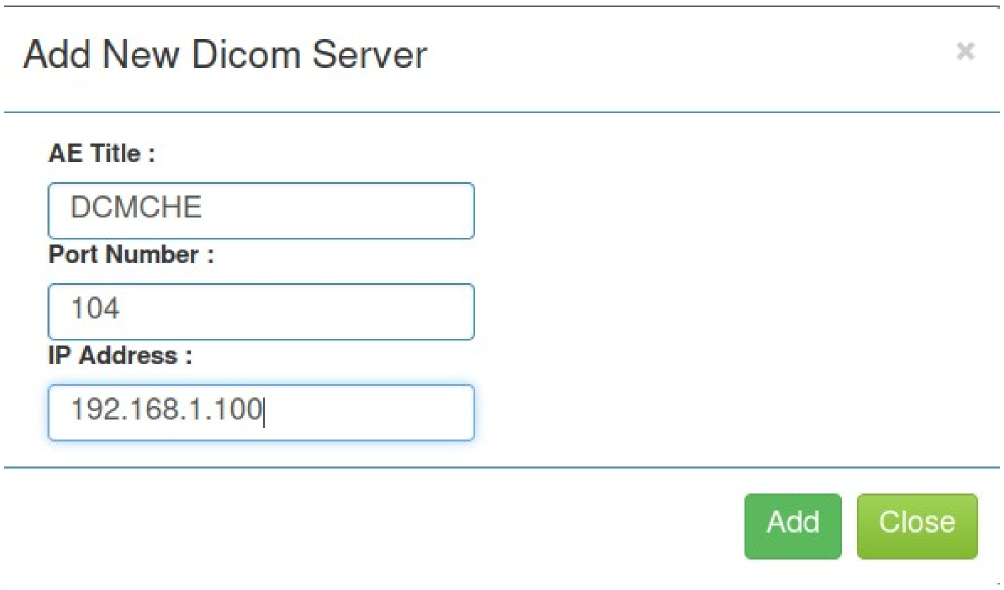


Dicom Nodes

AE title	Ip Address	Port Number	tools	
DCMCHE	192.168.1.100	104	Dicom Echo	Delete Node

Add New Dicom Node

On clicking add new Dicom Node the following wxwidget model is launched and delete Node will delete the Dicom Node. Dicom Echo will do C-Echo to the Dicom Node and indicate success or failure.



Add New Dicom Server

AE Title : DCMCHE

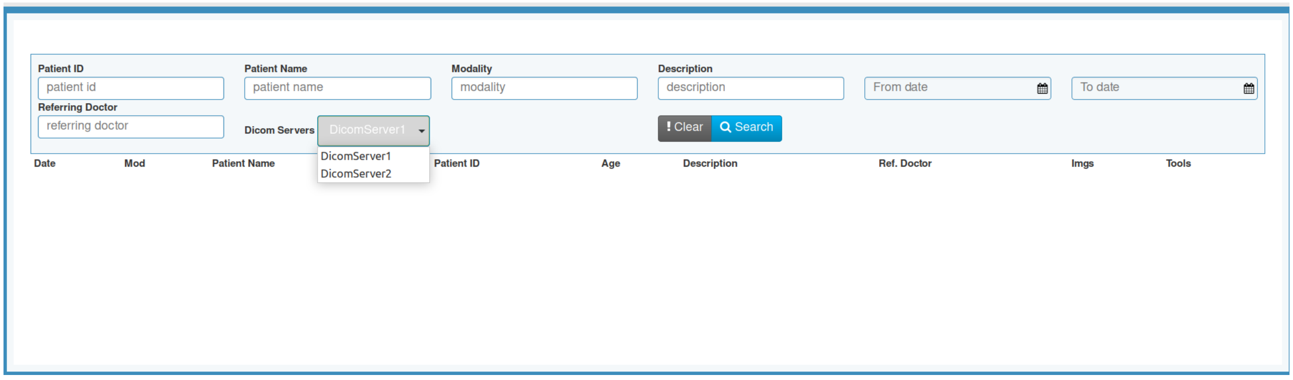
Port Number : 104

IP Address : 192.168.1.100

Using this modal we can add a New Dicom Node

Dicom Query

Onclick of Dicom query a model is opened where you need to select the dicom node from which you want to query as a drop down as shown below and will be implemented in wxwidgets. On clicking on the search the query results will be shown and in the tools there will be an option called download.



On click of Download the study will be downloaded from the Dicom Node using C-Move. This can be also implemented using C-Get but as not many PACS/Dicom implementations don't support it we can use C-Move which is easier to implement.

Dicom Operations(composite objects)

This section describes a few dicom operations which we plan to use in this project.

1.C-STORE The most basic DICOM operation, otherwise known as “DICOM Push” allows an SCU to send a Composite Instance to an SCP. For instance, it is used to send images from InVesalius to PACS/Dicom Node.

2.C-ECHO Sometimes called Dicom echo is to check if a Dicom node is alive or network connectivity is fine.

3.C-FIND is a Query/Retrieve service similar to a SQL Query using which a data set is passed on to the SCU from SCP.

4. C-MOVE This requests the C-MOVE SCP to act as a C-STORE SCU and to copy composite instances to a requested AE Title.

GDCMSCU(GDCM) or PynetDicom

GDCM library supports C-ECHO (SCU), C-FIND (SCU), C-STORE (SCU) and C-MOVE (SCU/SCP) C-MOVE operation are executed using two different ports (one for the SCU and one for the SCP). But GDCM does not have Storage SCP implementation.

PynetDicom has C-ECHO , C-FIND , C-STORE ,C-MOVE and Storage SCP implementation.

As I have done work on pynetdicom I am biased towards using it for the whole project. I understand that GDCM is extensively used for invesalius and the mentor can make a call if we can use pynetdicom for Storage SCP and GDCM can be used for C-ECHO (SCU), C-FIND (SCU), C-STORE (SCU) and C-MOVE. Pre-existing code in the net directory and import_network_panel.py can be enhanced.

Storage SCP or Dicom Server

Storage SCP *a.k.a* Dicom Server can be a process or service which responds to Dicom echo and receives Dicom images from other Dicom Nodes. The Supported Transfer syntaxes and presentation contexts will be decided before the project is started. AE title and port number will be stored as part of settings and the service or process will be launched with those settings.

To implement Storage SCP open source pynetdicom and pydicom have to be added to the InVesalius project as GDCM does not support this. A sample implementation has been provided along with the proposal as a github gist named dicomlistner.py which will run as a process.

A window service implementation using win32serviceutil, servicemanager, win32event and win32service python modules has been done for window os for Storage SCP. This will run as a windows service and the reference implementation as gihub gist is named as invesalius_service.py.

The Storage SCP when it receives a Dicom file will store it in the temp directory and import the file to InVesalius and then delete it.

Service Class User (C-Store SCU)

To implement this feature both gdcm and pynetdicom can be used. The will allow edited or generated Dicom files in InVesalius to be pushed or sent to other dicom nodes. Using the list of Dicom Nodes added a study or image can be selected and using the context menu can give a send option and from there the user can select a node and send the study or the image. Mentor can make a call if gdcm or pynetdicom can be used for this feature.

DICOM Query/Retrieve service

To implement the feature a wxwidget model shown above model with a drop down of Dicom nodes and search fields can be used or existing ui code can be used to implement this feature. C-FIND will get the list of studies from PACS/Dicom Node which will be shown in the UI. On clicking Download a C-Move for the selected study can be implemented using pynetdicom or gdcm.

Schedule / Deliverables:

Brief Timeline

- (Phase 0) Till 03 May: Pre-GSoC Period
- (Phase 1) 04 May - 28 May: Community Bonding Period
- (Phase 2) 29 May - 09 July: Coding Period 1
- (Phase 3) 10 July - 14 July: Phase 1 Evaluations
- (Phase 4) 14 July - 20 August: Coding Period 2
- (Phase 5) 21 August - 4 September: Phase 2 Evaluations

Detailed Project Timeline

Phase 0 [Pre-GSoC Period]

- **4 weeks (04 April to 03 May)**

I will be spending time learning Wxwidgets and spending time on understanding the InVesalius code base and start experimentation on the existing code base to add UI as documented in the proposal. I will be going though gdcm documentation and pynetdicom documentation and writing small test applications. I will go through the InVesalius documentation and user manual to better understand the capabilities and features.

Phase 1 [Community Bonding Period]

- **3 weeks (04 May to 28 May)**

During the community bonding period, the main focus will be to frame a road map for the project with the guidance of the mentor. This period can be used for scoping of requirements and deliverables at the end of this project.

Will be freezing the UI design for the project and will also plan with mentors help which parts of the project will be implemented with gdcm or pynetdicom.

Phase 2 [Coding Period 1]

- **2 weeks(29 May to 17 June)**

1. Will be adding in the menu for Network Setting
2. Creating a model for Dicom Server Settings
3. Save and edit Dicom Server Settings

- **2 weeks (18 June 09 July)**

1. Add Dicom Node Model with full with functionality to add/delete Dicom Node
2. Add functionality to do Dicom Echo to the added Dicom Node with success or failure.
3. Add UI modal for Dicom Query with drop down for added Dicom Nodes.

Phase 3 [GSoC Phase 1 Evaluations]

This period will be used to write a detailed report on the work done in Coding Period 1. All the work done will be uploaded and documentation will be created/uploaded.

Deliverables

- Demo and code for all the UI components and Dicom Echo will be given

Phase 4 [Coding Period 2]

- **2 weeks (14 July - 31 July)**

1. Dicom Server or Storage SCP will be done with received Dicom images/Studies being automatically added to InVesalius. For Windows OS a window service will be created for Storage SCP.

- **2 weeks (01 August - 20 August)**

1. Dicom Query using C-FIND will be done and a data list is populated with the study information which was queried based on search fields. On Clicking the download button a C-MOVE is done and the Dicom study is downloaded and added to InVesalius.
2. Test plan will be written and dicom network testing with other open source PACS or DVTK tool kit will be done.
3. If time permits will create a InVesalius Dicom conformance statement Document.

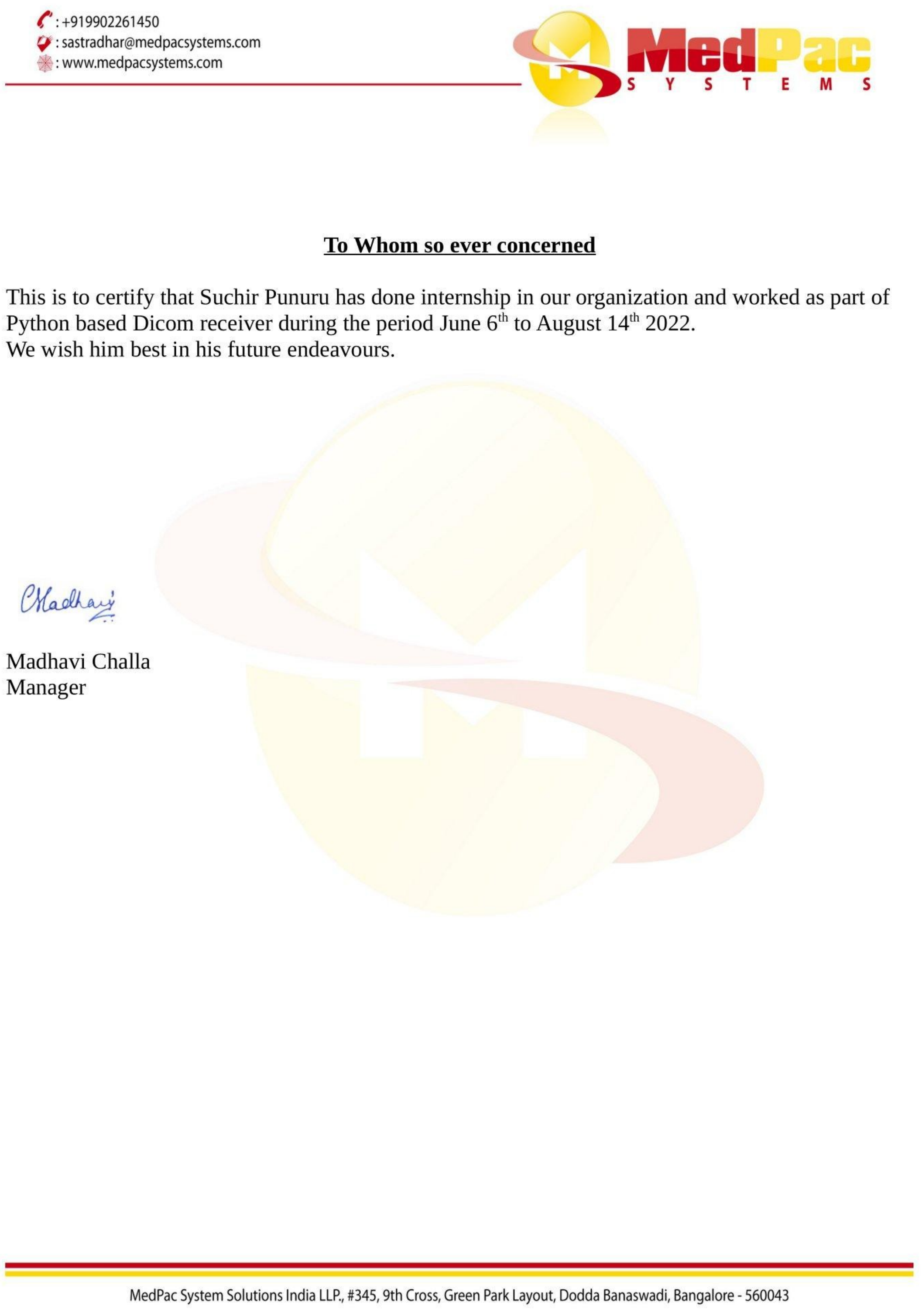
Phase 5 [Phase 2 Evaluations]

All documentation, code, and tests will be uploaded to the project Github page. **All the deliverables promised for GSoC will be provided by this stage.**

Biographical Information

I am Suchir Punuru, an undergraduate student at Mahindra University Hyderabad (India). I have an avid interest in Linux shell scripting and python programming. I can program in C, C++, Python,PHP,Mysql and web-development technologies such as HTML, CSS, and JavaScript. I have done projects using JQUERY and Bootstrap.

I did a summer internship last year in a PACS/Teleradiology company where I had implemented a Dicom Receiver for the PACS using python and pynetdicom.



: +919902261450
: sastradhar@medpacsystems.com
: www.medpacsystems.com

MedPac SYSTEMS

To Whom so ever concerned

This is to certify that Suchir Punuru has done internship in our organization and worked as part of Python based Dicom receiver during the period June 6th to August 14th 2022.
We wish him best in his future endeavours.

[illegible]

Madhavi Challa
Manager

MedPac System Solutions India LLP., #345, 9th Cross, Green Park Layout, Dodda Banaswadi, Bangalore - 560043

Expectations after the end of GSoC 2023

I hope to have a decent knowledge of Dicom network protocol along with working knowledge of Wxwidget and will have hands-on experience real time problem solving.