

Software Requirements Document

for

Pelvic Image Analysis and Geometry Reconstruction

Version <3.0> approved

Prepared by Nicol Barrios, Ralph Belleca, Ted Kim, Silvano Medina,
Robin Mok, Alejandra Olvera, Demetrius Parker, Sabino Ramirez,
Mary Semerdjian, Jason Tejada

Dr. Mathias Brieu (Biomedical)

May 13, 2022

Table of Contents

Table of Contents.....	pg 2
Revision History.....	pg 3
1. Introduction.....	pg 4
1.1. Purpose.....	pg 4
1.2. Intended Audience and Reading Suggestions.....	pg 4
1.3. Product Scope.....	pg 4
1.4. Definitions, Acronyms, and Abbreviations	pg 4
1.5. References.....	pg 4
2. Overall Description.....	pg 5
2.1. System Analysis.....	pg 5
2.2. Product Perspective.....	pg 5
2.3. Product Functions.....	pg 6
2.4. User Classes and Characteristics.....	pg 6
2.5. Operating Environment.....	pg 6
2.6. Design and Implementation Constraints.....	pg 6
2.7. User Documentation.....	pg 6
2.8. Assumptions and Dependencies.....	pg 7
2.9. Apportioning of Requirements.....	pg 7
3. External Interface Requirements.....	pg 8
3.1. User Interfaces.....	pg 8
3.2. Hardware Interfaces.....	pg 8
3.3. Software Interfaces.....	pg 8
3.4. Communications Interfaces.....	pg 8
4. Requirements Specification.....	pg 9
4.1. Functional Requirements.....	pg 9
4.2. External Interface Requirements.....	pg 10
4.3. Logical Database Requirements.....	pg 10
4.4. Design Constraints.....	pg 10
5. Other Nonfunctional Requirements.....	pg 11
5.1. Performance Requirements.....	pg 11
5.2. Safety Requirements.....	pg 11
5.3. Security Requirements.....	pg 11
5.4. Software Quality Attributes.....	pg 11
6. Legal and Ethical Considerations.....	pg 12
Appendix A: Glossary.....	pg 13
Appendix B: Analysis Models.....	pg 14
Appendix C: To Be Determined List.....	pg 15

Revision History

Name	Date	Reason For Changes	Version
Silvano	10/08/2021	Update Document	1.0
Silvano	12/09/2021	Update Document	1.1
Mary	12/09/2021	Update Document	1.1
Silvano	05/12/2022	Revise Document	2.0
Mary	05/13/2022	Finalize Document	3.0

1. Introduction

1.1 Purpose

This document shall:

1. Identify the requirements for Image Analysis and Geometry Reconstruction by giving an in-depth explanation of creating the model of organs, and the purpose of the project.
2. Detail requirements that must be met to effectively create the models for the three organs which are bladder, vagina, and rectum.

1.2 Intended Audience and Reading Suggestions

This document is intended for the project manager, developers, and users of the software.

The suggested reading sections are as follows:

- Project Manager: To help understand the requirements and steer the development of the software
- Developers: To help understand the requirements that need to be met for the implementation of the software
- Users: To help understand basic information pertaining to the software

1.3 Product Scope

The models of the three pelvic organs were created with the Clara Train framework. To create these models, we specified the organs by creating a binary label map using the masking process in 3D slicer. The binary label maps of the organs were then used to train and obtain the models of the organs with the Clara Train framework.

1.4 Definitions, Acronyms, and Abbreviations

Refer to Appendix: A

1.5 References

“Clara Train Framework” nvidia.com, publisher NVIDIA Version 4.1, published 2021

“Install Docker Engine on Ubuntu” docker.com, publisher Docker, publication 2013-2021

“Source Code for monai.losses.dice” monai.io, publisher MONAI, publication 2020-2021

“clara_pt_deepgrow_2d_annotation” nvidia.com, publisher NVIDIA, version 4.1, published March 24, 2022

“Medical Image Segmentation With Digits” nvidia.com, published by Hyungon Ryu and Jack Han

2. Overall Description

2.1 System Analysis

For general use, 3D Slicer and Blender had tough learning curves. The purpose of this program is to simplify general use cases of 3D Slicer and Blender to create organ models for reference use cases. This product is not to be used in a professional environment as it will not produce a peerless accurate model of the organs.

The major hurdles met during the projects were:

- Deciding on a method to simplify the usage of the programs and create a semi accurate model at the same time.
- Figuring out where to pull test data for A.I. learning.
- Acquiring third-party servers with sufficient hardware specification and privacy assurances.
- Learning to use the programs to understand the frustrations and skills needed to operate the programs.
- Creating the data for training the A.I.

Solutions to overcome hurdles:

- Testing between boundary points and deep-grow to compare speed, efficiency and accuracy.
- Pulled data from The Cancer Imaging Archive (TCIA) and additional samples provided by Dr. Mathias Brieu.
- Meeting with NVIDIA representatives to discuss privacy use of their servers.
- Conversing with AMAZON WEB SERVICES representatives to discuss the use of their servers.
- Set aside specific amount of time every week dedicated to learning and understanding the programs.
- Learning and understanding the Clara Train Framework.

2.2 Product Perspective

3D Slicer has an plug-in called NVIDIA A.I. Assisted Annotation(AIAA). This application connects 3D Slicer with Nvidia's AIAA servers over HTTP that connect to their Triton Servers which have several GPUs to help with generating highly accurate models. 3D Slicer is used to make binary labels for the Clara Train Framework application. Amazon Web Services (AWS) host the necessary computing hardware environment to implement the Clara Train Framework application used for training the A.I.

2.3 Product Functions

2.3.1 AWS

- Needs to have access to the AWS instance in order to use the Clara Train Framework.

2.3.2 3D Slicer

- Need to create binary label maps to use in Clara Train Framework.

2.3.2.1 Nvidia AIAA

- Needed to auto-annotate and generate the models of organs.

2.3.3 Clara Train Framework

- Needed to train the artificial intelligence on the organs.

2.4 User Classes and Characteristics

This product only has one level of functionality. The features inside the product are accessible to healthcare and medical professionals.

2.5 Operating Environment

The output of the project is an AI model trained on pelvic organ data. The team used Ubuntu, a Linux-based operating system in training and testing the AI model. The model is used as an input inside the Nvidia AIAA plug-in within the 3D Slicer application.

To use the AI model, the operating system has to be able to run 3D Slicer application as well as run the Nvidia AIAA server.

2.6 Design and Implementation Constraints

- Limited amount of MRI scans
- Privacy and Anonymization of MRI scans
- Lack of Nvidia AI models trained on pelvic organs

During the Fall 2021 semester, the first limitation that the team encountered was the limited amount of MRI scans. After obtaining some of them, the next obstacle was to anonymize the MRI scans to ensure user privacy. Lastly, the team encountered problems with Nvidia AIAA due to the lack of AI models trained in pelvic organs.

2.7 User Documentation

The team created documentation regarding the AI Creation and it has 3 sub-parts: Setting up the environment, Clara Train set-up, and AIAA setup.

Setting up required software/environment

1. Make sure to be in an Ubuntu environment.
2. Install [Docker Engine on Ubuntu](#)
 - a. Here's a screenshot of the first few instructions

```
Set up the repository
1. Update the apt package index and install packages to allow apt to use a repository over
HTTPS:

$ sudo apt-get update

$ sudo apt-get install \
ca-certificates \
curl \
gnupg \
lsb-release

2. Add Docker's official GPG key:

$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearm

3. Use the following command to set up the stable repository. To add the nightly or test
repository, add the word nightly or test (or both) after the word stable in the
commands below. Learn about nightly and test channels.

$ echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list
```

- b.
- 3. After installing Docker Engine, follow these post-installation instructions:
 - a. <https://docs.docker.com/engine/install/linux-postinstall/>
 - b. Follow steps 1-4. Also run commands to start Docker on boot.
 - c. This should be the last command you need from these instructions:

```
Configure Docker to start on boot

Most current Linux distributions (RHEL, CentOS, Fedora, Debian, Ubuntu 16.04 and higher) use systemd to
manage which services start when the system boots. On Debian and Ubuntu, the Docker service is configured to
start on boot by default. To automatically start Docker and Containerd on boot for other distros, use the
commands below:

$ sudo systemctl enable docker.service
$ sudo systemctl enable containerd.service
```

d.

Once docker is installed, and you can run hello-world container without sudo, begin the installation of Clara-train-sdk container: <https://docs.nvidia.com/clara/clara-train-sdk/pt/installation.html>

The rest of the documentation can be found here: [link](#)

2.8 Assumptions and Dependencies

It is assumed that developers will have the appropriate system and machine requirements. Developers that do not have a local GPU must use a virtual GPU through AWS.

2.9 Apportioning of Requirements

The future goal/requirement of this project is to build a user-friendly web application for medical professionals that incorporates some features of 3D Slicer and the AI model created.

3. External Interface Requirements

3.1 User Interfaces

No user interfaces.

3.2 Hardware Interfaces

No hardware interfaces.

3.3 Software Interfaces

3.3.1 3D Slicer, version number 4.11.20210226

- <https://download.slicer.org/>

3.3.2 Blender, version number 2.93.5

- <https://www.blender.org/download/>

3.3.3 Nvidia AIAA

- <https://github.com/NVIDIA/ai-assisted-annotation-client>

3.3.4 Nvidia Clara Train SDK v4.1

- <https://docs.nvidia.com/clara/clara-train-sdk/index.html>

3.3.5 AWS Deep Learning Base AMI Instance: P3.2xlarge

- <https://aws.amazon.com/marketplace/pp/prodview-v14n7ldaufpg4>

3.3.6 Monai Framework v0.7

- <https://github.com/Project-MONAI>

3.4 Communications Interfaces

3.4.1 Collaborative tools

We used zoom to conduct our meeting through voice calls and used discord to communicate with each other. Project work and project planning was done through Google Drive, Google slides and Google docs.

4. Requirements Specification

4.1 Functional Requirements

4.1.1 Input

4.1.1.1 Input data shall be prepared prior to training.

4.1.1.2 The system shall provide a python script to handle the logic.

4.1.1.3 The system shall also provide a file system architecture that is to be used for

4.1.2 Configuration

4.1.2.1 The system shall be pre-configured to default specifications.

4.1.2.2 The user shall have the option to add or remove components from within the

4.1.2.3 To achieve customization, the user must change the configuration to point to

4.1.2.4 The user shall then have the option to use a custom training script.

4.1.3 Performance analysis

4.1.3.1 The system shall gather metrics including images about the network's

4.1.3.2 This information shall be accessible from within the MMAR home folder.

4.1.3.3 Other software can be used to more deeply assess the data.

4.1.3.4 The system shall provide scripts to help speed up the process of inference and

4.1.1 Application Requirements

4.1.1.1 The application shall run on Windows, Mac, or Linux

4.1.1.2 The application should provide an interface to access functions of web application

4.1.1.2.1 The application will provide an MRI scan upload function

4.1.1.2.2 The application will provide a Select Region of Interest function

4.1.1.2.3 The application will provide a Model Render function

4.1.1.2.4 The application will provide an Organ Model download function

4.1.2 MRI Scan Upload Function

4.1.2.1 Allows user to upload their scans of the human body to create a general use model render of organ(s).

4.1.3 Select Region of Interest Function

4.1.4 Model Render Function

4.1.5 Organ Model Download Function

4.2 External Interface Requirements

Does not apply.

4.3 Logical Database Requirements

Does not apply.

4.4 Design Constraints

4.4.1 Hardware

4.4.1.1 Nvidia Clara Train utilizes GPU Acceleration.

4.4.1.2 Without meeting minimum requirements, the system will not work.

4.4.1.3 Using multiple GPUs will achieve faster processing.

4.4.1.4 Not all pre-trained models have the same requirements.

4.4.2 Software

4.4.2.1 The system was made within an Ubuntu environment.

4.4.2.2 User must also be in Ubuntu environment to train data.

4.4.2.3 For using the AIA server with an image viewer, the user shall have more

4.4.2.4 Containerization software such as Docker shall be used extensively.

4.4.2.5 Services like Amazon Web Services offer alternatives to physical hardware by which users shall complete training tasks from any hardware setup.

4.4.3 Web application user requirements

4.4.1.1 Functioning computer

4.4.1.2 Access to internet

4.4.1.3 Usable MRI scan

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 The application should take no more than X seconds to load 3D Slicer's tools

5.1.2 The maximum number of simultaneous users should be around X

5.1.3 Information type handled is DICOM MRI files

5.2 Safety Requirements

To maintain privacy, the user is highly recommended not to share organ models and other data with other users.

5.3 Security Requirements

Private data provided by anonymous users are discarded after the user is finished with their use. Any third-party resources have been checked to maintain privacy. The server hosts, programmers, and program owners do not have access to any private data that users may want to use in this web application.

5.4 Software Quality Attributes

The website is fully functional on all browsers on PC computers.

6. Legal and Ethical Considerations

The legal and ethical issues involved in this project are not limited to but include the following:

6.1 Use Cases

6.1.1 The users are not to use models for professional use

6.2 Legal use of body scans

6.3 User Intentions

6.3.1 Bad Intentions

6.3.1.1 Misleading the public

6.3.1.2 Inflammatory remarks

6.3.1.3 Propaganda

6.3.1.4 Weaponize data to target certain groups of people

6.3.1.5 Invasion of privacy of other beings

6.3.2 Good Intentions

6.3.2.1 Education

6.3.2.2 General Reference

6.4 Allow usage of our service to all skill levels.

6.5 Accommodate all realistic form factors used to access site.

6.6 Data privacy

Appendix A: Glossary

3D Slicer - a free, open source and multi-platform software package widely used for medical, biomedical, and related imaging research.

AIAA - Artificial Intelligence Assisted Annotation

AWS - Amazon Web Services

Binary Label Map - store all segments in a single 3D array.

Convolutional Neural Network - a class of deep neural networks used in deep learning and machine learning.

Decoder - combines info from the bottom of the “U” with high resolution.

Deep Grow - a pretrained model from NVIDIA that Applies segmentation based on foreground and background clicks by the user.

Dice Loss Function - calculated the similarity between two images.

Encoder - uses convolution and max pooling layers to downsample the image.

GANs - Generative Adversarial Networks

Loss Function - a method of evaluating how well a specific algorithm models the given data.

Manual Thresholding - Start from scratch and use traditional, manually used techniques to get the best possible 3D shape and output of the female pelvis and its specified key organs.

Masking - the process of blanking out a segment or area in a volumetric image to show only a selected organ.

MMAR - Medical Model ARchive

MONAI - Medical Open Network for AI

MRI - Magnetic Resonance Imaging of the abdomen.

NVIDIA - the inventor of the GPU, which creates interactive graphics on laptops, workstations, mobile devices, notebooks, and PCs, and more.

SDD - Software Design Document

Segmentation - a method in which a digital image is broken down into various subgroups called image segmentations which helps in reducing the complexity of the image to make further processing or analysis of the image simpler.

U-Net - an encoder-decoder network that can return accurate and detailed segmentations.

...

Appendix B: Analysis Models

Appendix C: To Be Determined List