Module Interface Specification for X-RayAssist

Team 17, Team MedTech Visionaries
Allison Cook
Ibrahim Issa
Mohaansh Pranjal
Nathaniel Hu
Tushar Aggarwal

April 4, 2024

1 Revision History

Date	Version	Notes
01/11/2024	0.0	Initial Document
01/12/2024	0.1	Started adding MIS for Module sections
01/14/2024	0.2	Added in Table from Module Guide (MG) into Module Decomposition section
01/16/2024	0.3	Made small updates to Table 1 in Module Decomposition section; Continued adding MIS for Module sections
01/17/2024	0.4	Completed the Symbols, Abbreviations and Acronyms and Introduction sections; Finished adding in MIS for Module sections
04/04/2024	0.5	updated MIS for backend ML modules according to changes during implementation

2 Symbols, Abbreviations and Acronyms

This section records the symbols, abbreviations and acronyms information for easy reference for terms used in this document.

For information on most of the symbols, abbreviations and acronyms referenced in this document, see the SRS Documentation at the following link: https://github.com/tusharagg1/chest-x-ray-ai/blob/main/docs/SRS/SRS.pdf.

The information on the rest of the symbols, abbreviations and acronyms referenced in this document are shown in the table below.

symbol	**description**
AI/ML	Artificial Intelligence/Machine Learning
DICOM	Digital Imaging and Communications in Medicine; technical standard for digital storage/transmission of medical images and related information
GUI	Graphical User Interface
JPEG/JPG	Joint Photographic Experts Group; digital image compression standard, image format
M	Module
MG	Module Guide
MVC	Model-View-Controller Software Architecture
NLP	Natural Language Processing
SRS	Software Requirements Specification
X-RayAssist	The Process of Designing and Developing Software; a reference to the software application described in this document

Contents

3 Introduction

The following document details the Module Interface Specifications for the [Your Program Name Here] software application. This software application (sometimes referred to as Software Engineering in this document) performs scans of chest x-ray images, looking for diseases/infections and making predictions. Those scan results and predictions of diseases/infections are then put into natural language radiology reports (or components) and returned.

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at https://github.com/tusharagg1/chest-x-ray-ai/tree/main.

4 Notation

The structure of the MIS for modules comes from Hoffman and Strooper (1995), with the addition that template modules have been adapted from Ghezzi et al. (2003). The mathematical notation comes from Chapter 3 of Hoffman and Strooper (1995). For instance, the symbol := is used for a multiple assignment statement and conditional rules follow the form $(c_1 \Rightarrow r_1|c_2 \Rightarrow r_2|...|c_n \Rightarrow r_n)$.

The following table summarizes the primitive data types used by X-RayAssist.

Data Type	Notation	Description
character	char	a single symbol or digit
string	str	an array of characters
boolean	bool	True or False
integer	$\mathbb{Z}$	a number without a fractional component in (-∞, ∞)
natural number	$\mathbb{N}$	a number without a fractional component in [1, ∞)
real	$\mathbb{R}$	any number in (-∞, ∞)

The specification of X-RayAssist uses some derived data types: sequences, strings, and tuples. Sequences are lists filled with elements of the same data type. Strings are sequences of characters. Tuples contain a list of values, potentially of different types. In addition, X-RayAssist uses functions, which are defined by the data types of their inputs and outputs. Local functions are described by giving their type signature followed by their specification.

5 Module Decomposition

The following table is taken directly from the Module Guide document for this project.

Level 1	**Level 2**
Hardware-Hiding Module	MedInstInter
Behaviour-Hiding Module	ConvertDCM
	ResultsGen
	ChestXrayRead
	ReportGen
	DatabaseOps
	UserAuthMgmt
	AppGUI
	Login
	PerfScan
	ViewResults
	GenHeatmaps
	MLEndpoint
Software Decision Module	AIModel
	Backend
	AppController

Table 1: Module Hierarchy

6 MIS of Medical Institution Interface Module

6.1 Module

MedInstInter

6.2 Uses

N/A

6.3 Syntax

6.3.1 Exported Constants

N/A

6.3.2 Exported Access Programs

Name	In	Out	Exceptions
connectToInst	instID: str, credentials: str	connectionStatus: bool	InvalidCredentialsException, InstNotFoundException

6.4 Semantics

6.4.1 State Variables

- connectedInsts: Set(str) - maintains a set of connected institution IDs.

6.4.2 Environment Variables

- InstsITSys: Set(str) - the set of external IT systems the application interfaces with to retrieve/exchange information.

6.4.3 Assumptions

- Patient data is stored in the medical institution's database, and the software intends to interface with their server to access that information.

6.4.4 Access Routine Semantics

connectToInst():

- transition:
  - Adds 'instID' to 'connectedInsts' if the provided 'credentials' is valid.

- output:
  - 'connectionStatus' is set to True if the connection is successful, False otherwise.
- exception:
  - Throws 'InvalidCredentialsException' if the provided credentials are invalid.
  - Throws 'InstNotFoundException' if the specified 'instID' does not exist.

6.4.5 Local Functions

N/A

7 MIS of Chest X-Ray Read Module

7.1 Module

ChestXRayRead

7.2 Uses

MLEndpoint

7.3 Syntax

7.3.1 Exported Constants

N/A

7.3.2 Exported Access Programs

Name	In	Out	Exceptions
initapp	None	bucket: firebase bucket	InvalidAuthException
getDCMfiles	patientdir: directory for x-rays, bucket: firebase bucket	dcmfiles: array of DCM file data as byte-like objects	TypeException: invalid file format read

7.4 Semantics

7.4.1 State Variables

N/A

7.4.2 Environment Variables

N/A

7.4.3 Assumptions

Auth key for firbase authentication is stored as a seperate file that is accessed using its path.

7.4.4 Access Routine Semantics

initapp():

- transition:

  - Initiates the firebase bucket object to read chest x-ray files.

- output:
  - A bucket object for the bucket storing the x-ray files for all patients.

- exception:
  - Throws 'InvalidAuthException' if authorization fails.

getDCMfiles():

- transition:
  - Reads DICOM files for a patient from the bucket to eventually pass to the ML model for processing.

- output:
  - A list of x-ray images read from the DICOM files stored as bytes.

- exception:
  - Throws 'TypeException' if it reads a file of the wrong type.

7.4.5 Local Functions

N/A

8 MIS of Results Generation Module

8.1 Module

ResultsGen

8.2 Uses

- AIModel
- GenHeatmaps

8.3 Syntax

8.3.1 Exported Constants

diseases: list of disease names that are checked

8.3.2 Exported Access Programs

Name	In	Out	Exceptions
scanallxrays	procImgs: list of Raw Processed xray Image data	classification: Disease Classification	-

8.4 Semantics

8.4.1 State Variables

N/A

8.4.2 Environment Variables

N/A

8.4.3 Assumptions

- The input list has one or more chest xray images.
- the images are stored as arrays of raw data (pixels in PNG format)

8.4.4 Access Routine Semantics

scanallxrays():

- transition:
  - generates prediction values for chest x-ray images. If more than one exists, the average value is taken for each disease.
- output:
  - 'classification' contains the generated disease prediction value for each disease.
- exception: N/A

8.4.5 Local Functions

- getprediction: generates prediction values for a single image.

9 MIS of Report Generation Module

9.1 Module

ReportGen

9.2 Uses

- AIModel

9.3 Syntax

9.3.1 Exported Constants

N/A

9.3.2 Exported Access Programs

Name	In	Out	Exceptions
generateReport	diagnosis: Disease Diagnosis	report: Radiology Report	-

9.4 Semantics

9.4.1 State Variables

N/A

9.4.2 Environment Variables

N/A

9.4.3 Assumptions

N/A

9.4.4 Access Routine Semantics

generateReport():

- transition: N/A
- output:
  - 'report' contains the generated summary report based on the provided disease diagnosis.
- exception: N/A

9.4.5 Local Functions

N/A

10 MIS of Database Operations Module

10.1 Module

DatabaseOps

10.2 Uses

N/A

10.3 Syntax

10.3.1 Exported Constants

N/A

10.3.2 Exported Access Programs

Name	In	Out	Exceptions
storeReport	report: Radiology Report, patientID: str	success: bool	ReportStorageException
retrieveReport	patientID: str	report: Radiology Report	ReportRetrievalException

10.4 Semantics

10.4.1 State Variables

- 'connectDatabase: bool' indicates whether the module is currently connected to the database.

10.4.2 Environment Variables

N/A

10.4.3 Assumptions

N/A

10.4.4 Access Routine Semantics

storeReport():

- transition:

  - Stores the provided 'report' in the database associated with the specified 'patientID'.

- output:
  - 'success' is set to True if the storing operation is successful, False otherwise.

- exception:
  - Throws 'ReportStorageException' if there is an issue storing the report.

retrieveReport():

- transition:
  - Retrieves the radiology report associated with the specified 'patientID' from the database.

- output:
  - 'report' contains the retrieved radiology report.

- exception:
  - Throws 'ReportRetrievalException' if there is an issue retrieving the report.

10.4.5 Local Functions

N/A

11 MIS of User Authentication/Management Module

11.1 Module

UserAuthMgmt

11.2 Uses

N/A

11.3 Syntax

11.3.1 Exported Constants

N/A

11.3.2 Exported Access Programs

Name	In	Out	Exceptions
authenticateUser	username: str, password: str	status: bool	InvalidCredentialsException, UserNotFoundException
createUserAccount	username: str, password: str	success: bool	UserCreationException
deleteUserAccount	username: str, password: str	success: bool	UserDeletionException
checkAuthentication	username: str	isAuthorized: bool	-

11.4 Semantics

11.4.1 State Variables

N/A

11.4.2 Environment Variables

N/A

11.4.3 Assumptions

N/A

11.4.4 Access Routine Semantics

authenticateUser():

- transition:
  - Verifies the provided 'username' and 'password' for authentication.
- output:
  - 'status' is set to True if authentication is successful, False otherwise.
- exception:
  - Throws 'InvalidCredentialsException' if the provided credentials are invalid.
  - Throws 'UserNotFoundException' if the specified user is not found.

createUserAccount():

- transition:
  - Creates a user account with the provided 'username' and 'password'.
- output:
  - 'success' is set to True if the account creation is successful, False otherwise.
- exception:
  - Throws 'UserCreationException' if there is an issue creating the user account.

deleteUserAccount():

- transition:
  - Deletes the user account associated with the specified 'username'.
- output:
  - 'success' is set to True if the account deletion is successful, False otherwise.
- exception:
  - Throws 'UserDeletionException' if there is an issue deleting the user account.

checkAuthentication():

- transition:
  - Checks whether the specified 'username' is currently authorized.

- output:
  - 'isAuthorized' is set to True if the user is authorized, False otherwise.
- exception: N/A

11.4.5 Local Functions

N/A

12 MIS of App GUI Module

12.1 Module

AppGUI

12.2 Uses

- Login
- PerfScan
- ViewResults

12.3 Syntax

12.3.1 Exported Constants

N/A

12.3.2 Exported Access Programs

Name	In	Out	Exceptions
displayLoginPage	-	-	-
displayScanPage	-	-	-
displayResultsPage	-	-	-

12.4 Semantics

12.4.1 State Variables

N/A

12.4.2 Environment Variables

N/A

12.4.3 Assumptions

N/A

12.4.4 Access Routine Semantics

displayLoginPage():

- transition: Navigates to and displays the login page for the application.
- output: N/A
- exception: N/A

displayScanPage():

- transition: Navigates to and displays the page for inputting an x-ray image for scanning.
- output: N/A
- exception: N/A

displayResultsPage():

- transition: Navigates to and displays the page for viewing scan results and reports.
- output: N/A
- exception: N/A

12.4.5 Local Functions

N/A

13 MIS of Login Module

13.1 Module

Login

13.2 Uses

- UserAuthMgmt

13.3 Syntax

13.3.1 Exported Constants

N/A

13.3.2 Exported Access Programs

Name	In	Out	Exceptions
login	username: str, password: str	loginStatus: bool	InvalidCredentialsException, UserNotFoundException

13.4 Semantics

13.4.1 State Variables

N/A

13.4.2 Environment Variables

N/A

13.4.3 Assumptions

N/A

13.4.4 Access Routine Semantics

login():

- transition:
  - Authenticates the provided 'username' and 'password'.
- output:

  - 'loginStatus' is set to True if login is successful, False otherwise.

- exception:
  - Throws 'InvalidCredentialsException' if the provided credentials are invalid.
  - Throws 'UserNotFoundException' if the specified user is not found.

13.4.5 Local Functions

N/A

14 MIS of Perform Scan Module

14.1 Module

PerfScan

14.2 Uses

N/A

14.3 Syntax

14.3.1 Exported Constants

N/A

14.3.2 Exported Access Programs

Name	In	Out	Exceptions
initiateScan	patient: selected patient	-	-

14.4 Semantics

14.4.1 State Variables

N/A

14.4.2 Environment Variables

- patientID: $\mathbb{Z}$

14.4.3 Assumptions

N/A

14.4.4 Access Routine Semantics

initiateScan():

- transition:
  - Receives the patient ID from the user to initiate the scanning process for that patient.
- output: N/A
- exception: N/A

14.4.5 Local Functions

N/A

15 MIS of View Results Module

15.1 Module

ViewResults

15.2 Uses

- PerfScan

15.3 Syntax

15.3.1 Exported Constants

N/A

15.3.2 Exported Access Programs

Name	In	Out	Exceptions
displayReport	report: Radiology Report	-	-

15.4 Semantics

15.4.1 State Variables

- 'loading: bool' indicates if the backend has received the scan information to be displayed

15.4.2 Environment Variables

N/A

15.4.3 Assumptions

N/A

15.4.4 Access Routine Semantics

displayReport():

- transition:
  - Displays the generated radiology report on the GUI.
- output: N/A
- exception: N/A

15.4.5 Local Functions

displayLoading():

- transition:
  - Sets the display page to the loading screen, until loading is set to false
- output: N/A
- exception: N/A

16 MIS of AI Model Module

16.1 Module

AIModel

16.2 Uses

- MLEndpoint

16.3 Syntax

16.3.1 Exported Constants

N/A

16.3.2 Exported Access Programs

Name	In	Out	Exceptions
getresponse	imgs: DICOM images	response: to send as JSON to be displayed	-

16.4 Semantics

16.4.1 State Variables

N/A

16.4.2 Environment Variables

N/A

16.4.3 Assumptions

N/A

16.4.4 Access Routine Semantics

getresponse():

- transition:
  - uses the getresults, heatmapgen and reportgen modules to get predictions, heatmaps and report.

- output:
  - 'resp' contains the diagnostic results wrapped as a JSON object.
- exception: N/A

16.4.5 Local Functions

- getrawimgs(): uses convertDCM module to convert DICOM image data to raw image pixel arrays.

17 MIS of Convert DCM Model Module

17.1 Module

convertDCM

17.2 Uses

- AIModel

17.3 Syntax

17.3.1 Exported Constants

N/A

17.3.2 Exported Access Programs

Name	In	Out	Exceptions
getimgdata	DICOM file data	raw image (PNG) data	InvalidFormatException
getxraypngs	list of raw image (PNG) data	list of encoded xray images	InvalidFormatException

17.4 Semantics

17.4.1 State Variables

N/A

17.4.2 Environment Variables

N/A

17.4.3 Assumptions

N/A

17.4.4 Access Routine Semantics

getimgdata():

- transition:
  - converts DICOM image data (bytes) to image array (PNG pixels).

- output:
  - raw image data as array of pixels.
- exception:
  - Throws 'InvalidFormatException' if the provided xray image is in an invalid format.

getxraypngs():

- transition:
  - stores list of raw image data (pixels) as a list of encoded byte array of pixels in PNG format.
- output:
  - list of images as encoded array of bytes (PNG format).
- exception:
  - Throws 'InvalidFormatException' if the provided images are in an invalid format.

17.4.5 Local Functions

- linstretchimg(): applies linear stretch in order to enhance the quality while converting DICOM files to a PNG format.
- getencodedimg(): used to convert a single image as raw pixel data to encoded array of bytes in PNG format.

18 MIS of Generate Heatmaps Module

18.1 Module

GenHeatmaps

18.2 Uses

- AIModel

18.3 Syntax

18.3.1 Exported Constants

N/A

18.3.2 Exported Access Programs

Name	In	Out	Exceptions
genheatmappatient	List of Raw image data	heatmap image data	-

18.4 Semantics

18.4.1 State Variables

N/A

18.4.2 Environment Variables

N/A

18.4.3 Assumptions

N/A

18.4.4 Access Routine Semantics

genheatmappatient():

- transition:
  - generates heatmaps for each disease and x-ray image for the patient
- output:

  - list of heatmap images stored as encoded array of bytes (PNG format).

- exception: N/A

18.4.5 Local Functions

- genheatmap(img): used to generate heatmaps for all diseases for a single xray image.

19 MIS of ML Endpoint Module

19.1 Module

MLEndpoint

19.2 Uses

N/A

19.3 Syntax

19.3.1 Exported Constants

N/A

19.3.2 Exported Access Programs

Name	In	Out	Exceptions
processfileendpoint	Directory name containing x-rays of the patient	response containing diagnosis, heatmaps and x-ray images to be displayed	InvalidRequestException

19.4 Semantics

19.4.1 State Variables

N/A

19.4.2 Environment Variables

N/A

19.4.3 Assumptions

N/A

19.4.4 Access Routine Semantics

processfileendpoint():

- transition:

  - uses ChestXRayRead to read xray images and AIModel to generate response containing diagnosis data

- output:
  - response containing prediction values, heatmaps, summary report and x-ray image data for that patient

- exception:
  - throws 'InvalidRequestException' if the request has an invalid body or the xray files are not found.

19.4.5 Local Functions

- mockresponse(): used for testing purposes to return a mock response to be sent to front end

20 MIS of Backend Module

20.1 Module

Backend

20.2 Uses

- UserAuthMgmt
- MedInstInter
- DatabaseOps

20.3 Syntax

20.3.1 Exported Constants

N/A

20.3.2 Exported Access Programs

Name	In	Out	Exceptions
connectDatabase	credentials: str	connectionStatus: bool	InvalidCredentialsException
disconnectDatabase		success: bool	-

20.4 Semantics

20.4.1 State Variables

20.4.2 Environment Variables

20.4.3 Assumptions

20.4.4 Access Routine Semantics

connectDatabase():

- transition: N/A
- output:
  - 'connectionStatus' is set to True if the connection is successful, False otherwise.
- exception:

  - Throws 'InvalidCredentialsException' if the provided credentials are invalid.

disconnectDatabase():

- transition: N/A
- output:
  - 'success' is set to True if the disconnection is successful, False otherwise.
- exception: N/A

20.4.5 Local Functions

N/A

21 MIS of App Controller Module

21.1 Module

AppController

21.2 Uses

- AIModel
- NLPModel
- AppGUI
- Backend

21.3 Syntax

21.3.1 Exported Constants

N/A

21.3.2 Exported Access Programs

Name	In	Out	Exceptions
accessBackend	-	-	-
accessGUI	-	-	-
accessAI	-	-	-
accessNLP	-	-	-

21.4 Semantics

21.4.1 State Variables

N/A

21.4.2 Environment Variables

N/A

21.4.3 Assumptions

N/A

21.4.4 Access Routine Semantics

accessBackend():

- transition:
  - Controller accesses the backend server.
- output: N/A
- exception: N/A

accessGUI():

- transition:
  - Controller accesses the application GUI.
- output: N/A
- exception: N/A

accessAI():

- transition:
  - Controller accesses the AI Model.
- output: N/A
- exception: N/A

accessNLP():

- transition:
  - Controller acceses the NLP Model.
- output: N/A
- exception: N/A

21.4.5 Local Functions

N/A

References

Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2003.

Daniel M. Hoffman and Paul A. Strooper. *Software Design, Automated Testing, and Maintenance: A Practical Approach*. International Thomson Computer Press, New York, NY, USA, 1995. URL http://citeseer.ist.psu.edu/428727.html.

22 Appendix