Module Interface Specification for SFWRENG 4G06A

Team #25, RapidCare
Pranav Kalsi
Gurleen Rahi
Inreet Kaur
Moamen Ahmed

January 17, 2025

1 Revision History

Date	Version	Notes
Jan 14, 2025	1.1	Initial Document

2 Symbols, Abbreviations and Acronyms

See SRS Documentation at SRS document

Symbol	Description
MG	Module Guide
M	Module
MIS	Module Interface Specification
API	Application Programming Interface
MFA	Multi-Factor Authentication

Contents

3 Introduction

The following document details the Module Interface Specifications for the RapidCareapplication.

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at https://github.com/PKALXI/RapidCare/blob/main/docs/Design/SoftArchitecture/MG.pdf.

4 Notation

The structure of the MIS for modules comes from **?**, with the addition that template modules have been adapted from **?**. The mathematical notation comes from Chapter 3 of **?**. 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 SFWRENG 4G06A.

Data Type	**Notation**	**Description**
character	char	a single symbol or digit
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 SFWRENG 4G06A 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, SFWRENG 4G06A 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	None
Behaviour-Hiding	User Authentication Module Administrator View Module Patient View Module Administrator Model Module Patient Model Module Broker Module Administrator Account Management Module Patient Account Management Module
Software Decision	Transcription Module Classification Module Diagnosis Prediction Module Medicine Prediction Module

Table 1: Module Hierarchy

6 MIS of User Authentication Module

6.1 Module

UserAuthentication

6.2 Uses

Firebase Auth

6.3 Syntax

6.3.1 Exported Constants

isAuthenticated: Boolean

6.3.2 Exported Access Programs

Name	In	Out	Exceptions
Auth	-	React.component	RenderError

6.4 Semantics

6.4.1 State Variables

isUserAdmin: Boolean

6.4.2 Environment Variables

N/A

6.4.3 Assumptions

N/A

6.4.4 Access Routine Semantics

Auth():

- transition: Renders the login page on the screen.
- output: N/A
- exception: RenderError — Thrown if the component fails to render.

6.4.5 Local Functions

login():

- transition: Renders adminstrator view page if user is administrator otherwise renders patient view page.
- output: N/A
- exception: InvalidCredentials - Thrown if the user enters invalid credentials.

ResetPassowrd():

- transition: Sends a reset link to the provided email and renders the login page.
- output: N/A
- exception: InvalidInpuError - Thrown if user input a invalid or unregistered email.

7 MIS of Administrator View Module

7.1 Module

Administrator

7.2 Uses

Broker Module
ReactJS

7.3 Syntax

7.3.1 Exported Constants

N/A

7.3.2 Exported Access Programs

Name	In	Out	Exceptions
Administrator	-	React.component	RenderError

7.4 Semantics

7.4.1 State Variables

isAuthenticated: Boolean

7.4.2 Environment Variables

Screen interface
Keyboard
Microphone

7.4.3 Assumptions

User has a functional screen, keyboard, and Microphone.

7.4.4 Access Routine Semantics

Administrator():

- transition: Renders a react component of the administrator view page.
- output: N/A
- exception: RenderError — Thrown if the component fails to render.

7.4.5 Local Functions

handleAdminAccount(id: string, record: FormData, requestType: string):

- transition: Sends an API request to the broker Module to process an add, delete, or update operation in the Administrator Database.
- output: N/A
- exception: InvalidInputError - Thrown if the formData is missing a field or is invalid.

validateInput(InputField: string):

- transition: Renders a success or error message outlining the action performed.
- output: N/A
- exception: InvalidInputError - The input data is incomplete or invalid.

8 MIS of Patient View Module

8.1 Module

Patient

8.2 Uses

Broker Module
ReactJS

8.3 Syntax

8.3.1 Exported Constants

N/A

8.3.2 Exported Access Programs

Name	In	Out	Exceptions
Patient	-	React.component	RenderError

8.4 Semantics

8.4.1 State Variables

isAuthenticated: Boolean

8.4.2 Environment Variables

Screen interface
Keyboard
Microphone

8.4.3 Assumptions

User has a functional screen, keyboard, and microphone.

8.4.4 Access Routine Semantics

Patient():

- transition: Renders a react component of the patient view page.

- output: N/A
- exception: RenderError — Thrown if the component fails to render.

8.4.5 Local Functions

handlePatientAccount(id: string, record: FormData, requestType: string):

- transition: Sends an API request to the API Module to process an add, delete, or update operation in the Patient Database.
- output: N/A
- exception: InvalidInputError - Thrown if the formData is missing a field or is invalid.

validateInput(InputField: string):

- transition: Renders a success/error message outlining the action performed.
- output: N/A
- exception: InvalidInputError - The input data is incomplete or invalid.

9 MIS of Broker Module

9.1 Module

Broker

9.2 Uses

- Transcription Module
- Classification Module
- Diagnosis Prediction Module
- Medicine Prediction Module
- Administrator Account Management Module
- Patient Account Management Module

9.3 Syntax

9.3.1 Exported Constants

N/A

9.3.2 Exported Access Programs

Name	In	Out	Exceptions
`getToken`	-	Token : String	InvalidTokenError
`transcribeText`	request : FormData	transcribedText : String	FailedResponseError
`classifyText`	request : FormData	classifiedText : map	FailedResponseError
`predictDiagnosis`	request : FormData	applicableDiagnosis : String	FailedResponseError
`predictedMedicine`	request : FormData	applicableMedicine : String	FailedResponseError

9.4 Semantics

9.4.1 State Variables

- secretKey : String

9.4.2 Environment Variables

N/A

9.4.3 Assumptions

- Requires a stable database connection.
- All end points in distributed systems are up.

9.4.4 Access Routine Semantics

getToken()-¿ String:

- **Transition:** N/A
- **Output:** Issues an access token with encoded with the `secretKey`
- **Exception:** InvalidTokenError : Invalid or expired authorization code.

transcribeText(request : FormData)-¿ String:

- **Transition:** N/A
- **Output:** Check if request header has valid token with `authorize`, then return live transcription of audio bytes in the request.body.
- **Exception:** FailedResponseError: The corresponding service/module has returned an error.

classifyText(request : FormData) -¿ map:

- **Transition:** N/A
- **Output:** Check if request header has valid token with `authorize`, then classify the request.text into the fields given in request.chart which represents the medical chart data in FormData form. Return map of text classified.
- **Exception:** FailedResponseError: The corresponding service/module has returned an error.

predictDiagnosis(request : FormData) -¿ String:

- **Transition:** N/A
- **Output:** Check if request header has valid token with `authorize`, then provide stream of diagnosis suggestions based on request.chart which represents the medical chart data in FormData form.
- **Exception:** FailedResponseError: The corresponding service/module has returned an error.

predictMedicine(request : FormData)-¿ String:

- **Transition:** N/A
- **Output:** Check if request header has valid token with `authorize`, then provide stream of medicine suggestions based on request.chart which represents the medical chart data in FormData form.
- **Exception:** FailedResponseError: The corresponding service/module has returned an error.

9.4.5 Local Functions

authorize(header : String) -¿ boolean:

- **Transition:** N/A
- **Output:** This is a header function to make sure the request is authorized, here this function returns whether the request is authorized.
- **Exception:** Invalid client credentials.

10 MIS of Transcription Module

10.1 Module

TransMod

10.2 Uses

This module does not use any other modules. It uses audio data as the input.

10.3 Syntax

10.3.1 Exported Constants

N/A

10.3.2 Exported Access Programs

Name	In	Out	Exceptions
trans_mod	audio data stream	written transcribed text	UnclearAudio

10.4 Semantics

10.4.1 State Variables

N/A

10.4.2 Environment Variables

- Recording Tool: It needs access to the recording tool so that the conversation between the patient and healthcare professional is recorded.

10.4.3 Assumptions

- It is assumed that the input audio data is stable and will be provided in streams.
- It is assumed that the medical terminologies used in the conversation is accurate.
- There is no background noise.

10.4.4 Access Routine Semantics

trans_mod(AudioData: Stream)():

- transition:
  - Input audio data is provided in continuous streams in real-time.
  - The streams are sent to the speech-recognition engine for transcription.
- output: Written text transcribed from the audio stream.
- exception: UnclearAudio - If there is any issue with the audio, for example: low volume or words not clear.

10.4.5 Local Functions

N/A

11 MIS of Classification Module

11.1 Module

ClassifyMod

11.2 Uses

It uses transcribed text from Transcription Module.

11.3 Syntax

11.3.1 Exported Constants

N/A

11.3.2 Exported Access Programs

Name	In	Out	Exceptions
ClassifyMod	Transcribed Text	Classified Text	ClassificationError

11.4 Semantics

11.4.1 State Variables

N/A

11.4.2 Environment Variables

N/A

11.4.3 Assumptions

- It is assumed that the text is transcribed accurately by the Transcription Module for correct classification.

11.4.4 Access Routine Semantics

ClassifyMod(transcribed_text: text):

- transition:
  - Uses the classification model or rules to categorize the text or extract relevant information.
  - Structures the output as classified text containing extracted information.

- output: Classified text generated for report generation.
- exception: ClassificationError - If the module fails to classify text accurately or encounters an error.

11.4.5 Local Functions

N/A

12 MIS of Diagnosis Prediction Module

12.1 Module

DiagnosisPred

12.2 Uses

- Report Generation Module
- Preprocessing Module
- Tensorflow
- Scikit-Learn
- Flask
- Flask-CORS

12.3 Syntax

12.3.1 Exported Constants

The exported constants for this module would a prediction of the possible diagnosis.

12.3.2 Exported Access Programs

Name	In	Out	Exceptions
diagnosePatient	request:FormData	Prediction of possible Diagnosis : String	InputDimError

12.4 Semantics

12.4.1 State Variables

- model: Tensorflow model

12.4.2 Environment Variables

N/A

12.4.3 Assumptions

- Patients are not making up symptoms and all input features are accurate.

12.4.4 Access Routine Semantics

diagnosePatient(request : FormData):

- transition: N/A
- output: Returns the predicted diagnosis for the patient based on preprocessed input given by `preProcessData`. FormData is expected to contain the key-value format of (past medical history:String, symptoms:String, user characteristics including age:int and weightLint, and physician confirmed diagnosis:String)
- exception: InputDimError - The expected request body items were not received or were in the wrong format.

12.4.5 Local Functions

preProcessData(pastHistory: String, symptoms: String, user -¿ {age, weight}):

- transition: N/A
- output: Preprocess the text using TF-ID and normalize the continuous inputs and return the preprocessed data.
- exception: InputDimError - The expected arguments were not received or were in the wrong format.

13 MIS of Medicine Prediction Module

13.1 Module

MedPred

13.2 Uses

- Report Generation Module
- Preprocessing Module
- Tensorflow
- Flask
- Flask-CORS
- Scikit-Learn

13.3 Syntax

13.3.1 Exported Constants

The exported constants for this module would be a prediction of the possible medicine to treat the diagnosis.

13.3.2 Exported Access Programs

Name	In	Out	Exceptions
medicatePatient	request : FormData	Prediction of possible Medicine : String	InputDimError

13.4 Semantics

13.4.1 State Variables

- model: Tensorflow model

13.4.2 Environment Variables

N/A

13.4.3 Assumptions

- All input features are accurate.

13.4.4 Access Routine Semantics

medicatePatient(request : FormData):

- transition: N/A
- output: preprocess the request body (FormData) which contains the has the key-value format of (past medical history:String, symptoms:String, user characteristics including age:int and weightLint, and physician confirmed diagnosis:String)
- exception: InputDimError - The expected request body items were not received or were in the wrong format.

13.4.5 Local Functions

preProcessData(diagnosis: String, pastHistory: String, symptoms: String, user -¿ {age, weight}):

- transition: N/A
- output: Preprocess the text using TF-ID and normalize the continuous inputs, finally the diagnosis will be encoded using a LabelEncoder then return the preprocessed data.
- exception: InputDimError - The expected arguments were not received or were in the wrong format.

14 MIS of Administrator Account Management Module

14.1 Module

AdministratorAccountManagement

14.2 Uses

Broker Module

14.3 Syntax

14.3.1 Exported Constants

N/A

14.3.2 Exported Access Programs

Name	In	Out	Exceptions
addHealthcareProfessional	id:string, value: FormData	-	InvalidInputError
deleteHealthcareProfessional	id:string	-	IdNotFound
updateHealthcareProfessional	id:string, value: FormData	-	InvalidInputError

14.4 Semantics

14.4.1 State Variables

dbConnection : Database connection point.

14.4.2 Environment Variables

N/A

14.4.3 Assumptions

Database containing the account information for healthcare professionals exists.

14.4.4 Access Routine Semantics

addHealthcareProfessional(id:string, record: FormData):

- transition: Adds a new document with the provided details to the database.
- output: N/A
- exception: InvalidInputError - The input data is incomplete or invalid or a duplicate document already exists.

deleteHealthcareProfessional(id:string):

- transition: Deletes the corresponding document from the database.
- output: N/A
- exception: IdNotFound - Provided id is invalid or does not exist.

updateHealthcareProfessional(id:string, record: FormData):

- transition: Update document with the provided details in the database.
- output: N/A
- exception: InvalidInputError - The input data is incomplete or invalid.

14.4.5 Local Functions

N/A

15 MIS of Patient Account Management Module

15.1 Module

PatientAccountManagement

15.2 Uses

Broker Module

15.3 Syntax

15.3.1 Exported Constants

N/A

15.3.2 Exported Access Programs

Name	In	Out	Exceptions
createPatientRecord	id:string, value: FormData	-	InvalidInputError
deletePatientRecord	id:string	-	IdNotFound
updatePatientRecord	id:string, value: FormData	-	InvalidInputError

15.4 Semantics

15.4.1 State Variables

dbConnection : Database connection point.

15.4.2 Environment Variables

N/A

15.4.3 Assumptions

Database containing the account information for the patients exists.

15.4.4 Access Routine Semantics

createPatientRecord(id:string, record: FormData):

- transition: Adds a new document with the provided details to the database.

- output: N/A
- exception: InvalidInputError - The input data is incomplete or invalid or a duplicate document already exists.

deletePatientRecord(id:string):

- transition: Deletes the corresponding document from the database.
- output: N/A
- exception: IdNotFound - Provided id is invalid or does not exist.

updatePatientRecord(id:string, record: FormData):

- transition: Update document with the provided details in the database.
- output: N/A
- exception: InvalidInputError - The input data is incomplete or invalid.

15.4.5 Local Functions

N/A

16 Appendix

[Extra information if required —SS]

Appendix — Reflection

[Not required for CAS 741 projects —SS]

The information in this section will be used to evaluate the team members on the graduate attribute of Problem Analysis and Design.

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?

2. What pain points did you experience during this deliverable, and how did you resolve them?

3. Which of your design decisions stemmed from speaking to your client(s) or a proxy (e.g. your peers, stakeholders, potential users)? For those that were not, why, and where did they come from?

4. While creating the design doc, what parts of your other documents (e.g. requirements, hazard analysis, etc), it any, needed to be changed, and why?

5. What are the limitations of your solution? Put another way, given unlimited resources, what could you do to make the project better? (LO_ProbSolutions)

6. Give a brief overview of other design solutions you considered. What are the benefits and tradeoffs of those other designs compared with the chosen design? From all the potential options, why did you select the documented design? (LO_Explores)