

Module Interface Specification for SFWRENG 4G06

Capstone Design Project

Team #18, InfiniView-AI

Anhao Jiao

Kehao Huang

Qianlin Chen

Qi Shu

Xunzhou Ye

January 17, 2024

1 Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

2 Symbols, Abbreviations and Acronyms

See SRS Documentation at [\[give url —SS\]](#)

[\[Also add any additional symbols, abbreviations or acronyms —SS\]](#)

Contents

1	Revision History	i
2	Symbols, Abbreviations and Acronyms	ii
3	Introduction	1
4	Notation	1
5	Module Decomposition	1
6	MIS of RTC Control Module	3
6.1	Module	3
6.2	Uses	3
6.3	Syntax	3
6.3.1	Exported Constants	3
6.3.2	Exported Access Programs	3
6.4	Semantics	3
6.4.1	State Variables	3
6.4.2	Environment Variables	3
6.4.3	Assumptions	3
6.4.4	Access Routine Semantics	3
6.4.5	Local Functions	4
7	MIS of Media Control Module	4
7.1	Module	4
7.2	Uses	5
7.3	Syntax	5
7.3.1	Exported Constants	5
7.3.2	Exported Access Programs	5
7.4	Semantics	5
7.4.1	State Variables	5
7.4.2	Environment Variables	5
7.4.3	Assumptions	5
7.4.4	Access Routine Semantics	5
7.4.5	Local Functions	6
8	MIS of Instructor View Module	6
8.1	Module	6
8.2	Uses	6
8.3	Syntax	6
8.3.1	Exported Constants	6
8.3.2	Exported Access Programs	6

8.4	Semantics	6
8.4.1	State Variables	6
8.4.2	Environment Variables	7
8.4.3	Assumptions	7
8.4.4	Access Routine Semantics	7
8.4.5	Local Functions	7
9	MIS of Practitioner View Module	8
9.1	Module	8
9.2	Uses	8
9.3	Syntax	8
9.3.1	Exported Constants	8
9.3.2	Exported Access Programs	8
9.4	Semantics	9
9.4.1	State Variables	9
9.4.2	Environment Variables	9
9.4.3	Assumptions	9
9.4.4	Access Routine Semantics	9
9.4.5	Local Functions	9
10	MIS of Annotation Configuration Module	9
10.1	Module	9
10.2	Uses	9
10.3	Syntax	10
10.3.1	Exported Constants	10
10.3.2	Exported Access Programs	10
10.4	Semantics	10
10.4.1	State Variables	10
10.4.2	Environment Variables	10
10.4.3	Assumptions	10
10.4.4	Access Routine Semantics	10
10.4.5	Local Functions	11
11	Appendix	13

3 Introduction

The following document details the Module Interface Specifications for [Fill in your project name and description —SS]

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at [provide the url for your repo —SS]

4 Notation

[You should describe your notation. You can use what is below as a starting point. —SS]

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 $(c_1 \Rightarrow r_1 | c_2 \Rightarrow r_2 | \dots | c_n \Rightarrow r_n)$.

The following table summarizes the primitive data types used by SFWRENG 4G06 Capstone Design Project.

Data Type	Notation	Description
character	char	a single symbol or digit
integer	\mathbb{Z}	a number without a fractional component in $(-\infty, \infty)$
natural number	\mathbb{N}	a number without a fractional component in $[1, \infty)$
real	\mathbb{R}	any number in $(-\infty, \infty)$

The specification of SFWRENG 4G06 Capstone Design Project 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 4G06 Capstone Design Project 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	
Behaviour-Hiding	Input Parameters Output Format Output Verification Temperature ODEs Energy Equations Control Module Specification Parameters Module
Software Decision	Sequence Data Structure ODE Solver Plotting

Table 1: Module Hierarchy

6 MIS of RTC Control Module

6.1 Module

RTCControl

6.2 Uses

Web APIs

STUN Server Module

6.3 Syntax

6.3.1 Exported Constants

N/A

6.3.2 Exported Access Programs

Name	In	Out	Exceptions
createPeerConnection	JSON	RTCPeerConnection	-
closeRemoteConnection	RTCPeerConnection	-	-
negotiate	RTCPeerConnection	-	-

6.4 Semantics

6.4.1 State Variables

N/A

6.4.2 Environment Variables

STUN_SERVER_ADDRESS: string — represents the address of the STUN server

SFU_BROADCAST_API: string — represents the API endpoint for SFU broadcast API

SFU_CONSUME_API: string — represents the API endpoint for SFU consume API

6.4.3 Assumptions

SFU server and STUN servers are running in normal conditions.

6.4.4 Access Routine Semantics

createPeerConnectionWith(config: JSON):

- transition: N/A

- output: `pc := RTCPeerConnection` — initializes a new `RTCPeerConnection` based on the given configuration.
- exception: N/A

`closeRemoteConneciton(pc: RTCPeerConnection):`

- transition: `pc.signalingState := closed` — closes peer connection and send a signal to the connected peer connection.
- output: N/A
- exception: N/A

`negotiate(pc: RTCPeerConnection):`

- transition:
`pc.localDescription := RTCSessionDescriptionInit`
`pc.remoteDescription := RTCSessionDescriptionInit`
sets the local description of the peer connection to its generated SDP, and set the remote description of the peer connection to its received SDP from `SFU_BROADCAST_API`.
- output: N/A
- exception: N/A

`getRemoteStream(pc: RTCPeerConnection):`

- transition: `pc.event := getRemoteEvent(pc).streams`
- output: N/A
- exception: N/A

6.4.5 Local Functions

`getRemoteEvent(pc: RTCPeerConnection):`

- transition: N/A
- output: `pc.event := RTCTrackEvent`
- exception: N/A

7 MIS of Media Control Module

7.1 Module

`MediaContorl`

7.2 Uses

Web APIs

7.3 Syntax

7.3.1 Exported Constants

N/A

7.3.2 Exported Access Programs

Name	In	Out	Exceptions
setMicEnabled	Boolean	-	-
setCameraEnabled	Boolean	-	-
getStream	-	MediaStream	-

7.4 Semantics

7.4.1 State Variables

isMicEnabled: Boolean

isCameraEnabled: Boolean

7.4.2 Environment Variables

Microphone

Camera

7.4.3 Assumptions

User's devices have a functioning screen, camera and microphone.

7.4.4 Access Routine Semantics

setMicEnabled(isEnabled: Boolean):

- transition: isMicEnabled := isEnabled
- output: N/A
- exception: N/A

setCameraEnabled(isEnabled: Boolean):

- transition: isCameraEnabled := isEnabled
- output: N/A

- exception: N/A

getStream():

- transition: N/A
- output: returns the user media stream based on the state value isCameraEnabled and isMicEnabled
- exception: N/A

7.4.5 Local Functions

N/A

8 MIS of Instructor View Module

8.1 Module

Instructor

8.2 Uses

Media Control Module

RTC Control Module

Annotation Configuration Module

React

Web APIs

8.3 Syntax

8.3.1 Exported Constants

N/A

8.3.2 Exported Access Programs

Name	In	Out	Exceptions
Instructor	-	React.component	-

8.4 Semantics

8.4.1 State Variables

remoteVideoRef: HTMLVideoElement

selfVideoRef: HTMLVideoElement

peerConnection: RTCPeerConnection

8.4.2 Environment Variables

Screen

8.4.3 Assumptions

User's devices have a functioning screen, camera and microphone.

8.4.4 Access Routine Semantics

Instructor():

- transition: N/A
- output: renders a react component of the instructor view page
- exception: N/A

8.4.5 Local Functions

setPeerConnection(pc: RTCPeerConnection):

- transition: peerConnection := pc
- output: N/A
- exception: N/A

getSelfVideo():

- transition:
selfVideoRef.current.video.srcObject:= MediaControl.getStream()
render video stream from the local camera to screen
- output: N/A
- exception: N/A

startRemoteSharing():

- transition: peerConnection.addTrack := MediaControl.getStream()
- output: N/A
- exception: N/A

stopRemoteSharing():

- transition:
`remoteVideoRef.current.video.srcObject = null`
`peerConnection.close:= true`
 stops the remote video on the user's screen and close the RTCPeerConnection
- output: N/A
- exception: N/A

`getRemoteVideo()`:

- transition: get remote video coming from the SFU server and render it on the user's screen.
- output: N/A
- exception: N/A

9 MIS of Practitioner View Module

9.1 Module

Practitioner

9.2 Uses

Media Control Module
 RTC Control Module
 Annotation Configuration Module
 React
 Web APIs

9.3 Syntax

9.3.1 Exported Constants

N/A

9.3.2 Exported Access Programs

Name	In	Out	Exceptions
Practitioner	-	React.component	-

9.4 Semantics

9.4.1 State Variables

remoteVideoRef: HTMLVideoElement

peerConnection: RTCPeerConnection

9.4.2 Environment Variables

Screen

9.4.3 Assumptions

User's devices have a functioning screen.

9.4.4 Access Routine Semantics

N/A

9.4.5 Local Functions

setPeerConnection(pc: RTCPeerConnection):

- transition: peerConnection := pc
- output: N/A
- exception: N/A

getRemoteVideo():

- transition: get remote video coming from the SFU server and render it on the user's screen.
- output: N/A
- exception: N/A

10 MIS of Annotation Configuration Module

10.1 Module

AnnotationConfig

10.2 Uses

RTC Control Module

React

10.3 Syntax

10.3.1 Exported Constants

N/A

10.3.2 Exported Access Programs

Name	In	Out	Exceptions
setIsSkeletonEnabled	Boolean	-	-
setIsCOMEnabled	Boolean	-	-
getIsSkeletonEnable	-	Boolean	-
getIsCOMEnable	-	Boolean	-

10.4 Semantics

10.4.1 State Variables

isSkeletonEnabled: Boolean

isCOMEnabled: Boolean

10.4.2 Environment Variables

N/A

10.4.3 Assumptions

N/A

10.4.4 Access Routine Semantics

setIsSkeletonEnabled(isEnabled: Boolean):

- transition: isSkeletonEnabled := isEnabled
- output: N/A
- exception: N/A

setIsCOMEnabled(isEnabled: Boolean):

- transition: isCOMEnabled := isEnabled
- output: N/A
- exception: N/A

getIsSkeletonEnabled():

- transition: N/A
- output: isSkeletonEnabled
- exception: N/A

getIsCOMEnabled():

- transition: N/A
- output: isCOMEnabled
- exception: N/A

10.4.5 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>.

11 Appendix

[Extra information if required —SS]