Module Interface Specification for Hairesthetics

Team 18
Charlotte Cheng
Marlon Liu
Senni Tan
Qiushi Xu
Hongwei Niu
Bill Song

January 19, 2023

1 Revision History

Date	**Version**	**Notes**
Jan 17	0	Rev0 MIS

2 Symbols, Abbreviations and Acronyms

symbol	description
ML	Machine Learning
UI	User Interface
AI	Artificial Intelligence
AR	Augumented Reality
App	Application
API	Application programming interface
REST	Representational state transfer
RGB	Red, Green, Blue
macOS	Operating system developed by Apple Inc
MG	Module Guide
MIS	Module Interface Specification

See SRS Documentation at /docs/SRS/SRS.pdf

Contents

List of Tables

List of Figures

3 Introduction

The following document details the Module Interface Specifications for the Hairesthetics application. Hairesthetics is an application that simulates 3D hairstyles.

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at https://github.com/marlon4dashen/Hairesthetics.

4 Notation

The structure of the MIS for modules comes from HoffmanAndStrooper1995, with the addition that template modules have been adapted from GhezziEtAl2003. The mathematical notation comes from Chapter 3 of HoffmanAndStrooper1995. 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 the modules.

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 our modules 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, our modules use 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

This section provides an overview of the module design. Modules are summarized in a hierarchy decomposed by secrets in Table 1. The modules listed below, which are leaves in the hierarchy tree, are the modules that will actually be implemented.

M1: Controller Module

M2: Facial Recognition Module

M3: Hair Color Module

M4: Hair Style Module

M5: Salon Recommendation Module

M6: ML Model Module

M7: Utility Module

M8: Hair Color View Module

M9: Hair Style View Module

M10: Salon Recommendation Interface Module

M11: Home View Module

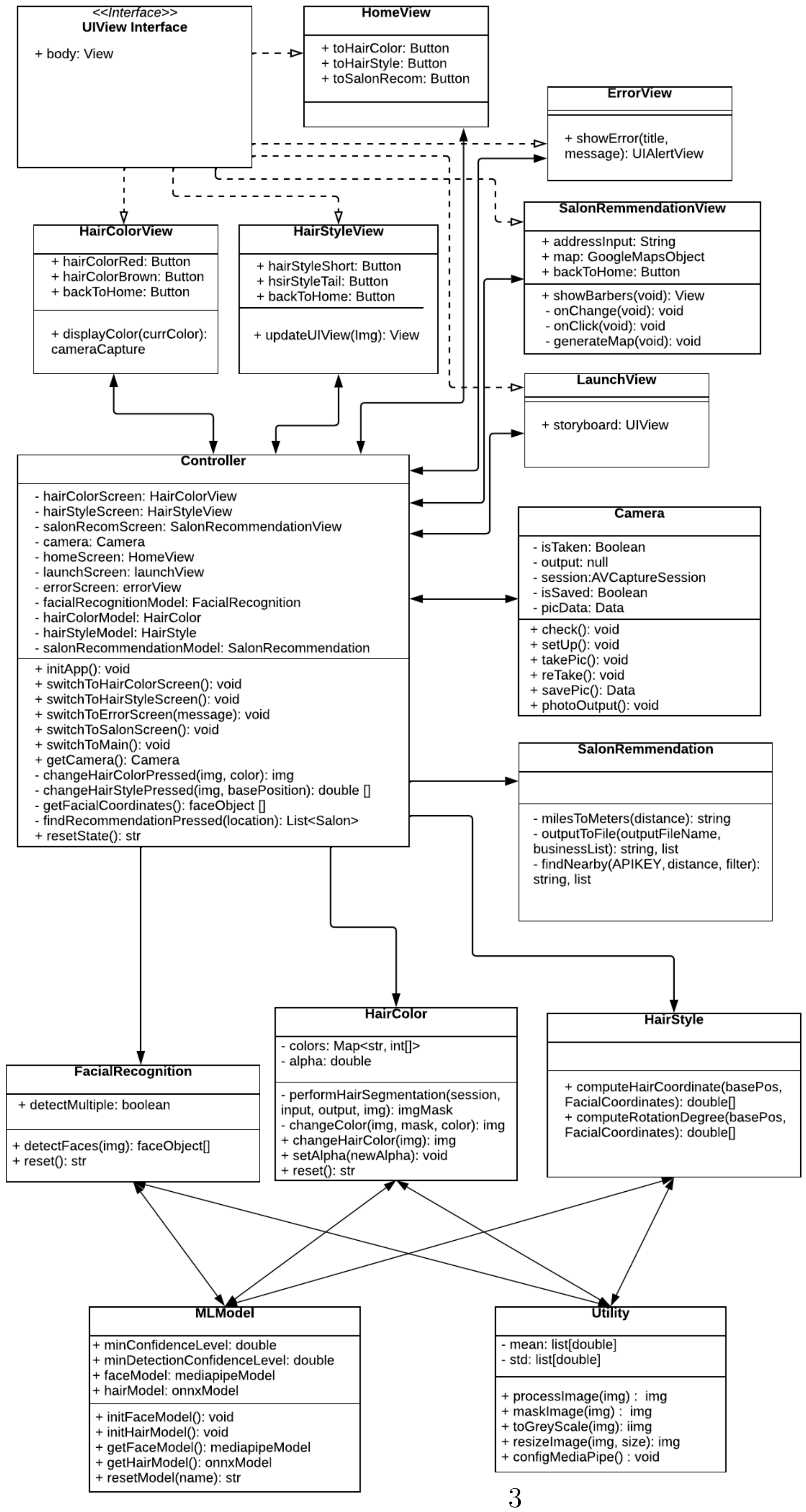
M12: Camera Module

M13: Error View Module

Level 1	**Level 2**
Hardware-Hiding Module	M13
Behaviour-Hiding Module	M1 M2 M3 M4 M5 M8 M9 M10 M11 M13
Software Decision Module	M6 M7

Table 1: Module Hierarchy

5.1 UML Diagram



<<Interface>>
UIView Interface
+ body: View
HomeView
+ toHairColor: Button
+ toHairStyle: Button
+ toSalonRecom: Button
ErrorView
+ showError(title, message): UIAlertView
SalonRemmendationView
+ addressInput: String
+ map: GoogleMapsObject
+ backToHome: Button
+ showBarbers(void): View
- onChange(void): void
- onClick(void): void
- generateMap(void): void
HairColorView
+ hairColorRed: Button
+ hairColorBrown: Button
+ backToHome: Button
+ displayColor(currColor): cameraCapture
HairStyleView
+ hairStyleShort: Button
+ hsirStyleTail: Button
+ backToHome: Button
+ updateUIView(Img): View
LaunchView
+ storyboard: UIView
Controller
- hairColorScreen: HairColorView
- hairStyleScreen: HairStyleView
- salonRecomScreen: SalonRecommendationView
- camera: Camera
- homeScreen: HomeView
- launchScreen: launchView
- errorScreen: errorView
- facialRecognitionModel: FacialRecognition
- hairColorModel: HairColor
- hairStyleModel: HairStyle
- salonRecommendationModel: SalonRecommendation
+ initApp(): void
+ switchToHairColorScreen(): void
+ switchToHairStyleScreen(): void
+ switchToErrorScreen(message): void
+ switchToSalonScreen(): void
+ switchToMain(): void
+ getCamera(): Camera
- changeHairColorPressed(img, color): img
- changeHairStylePressed(img, basePosition): double []
- getFacialCoordinates(): faceObject []
- findRecommendationPressed(location): List<Salon>
+ resetState(): str
Camera
- isTaken: Boolean
- output: null
- session:AVCaptureSession
- isSaved: Boolean
- picData: Data
+ check(): void
+ setUp(): void
+ takePic(): void
+ reTake(): void
+ savePic(): Data
+ photoOutput(): void
SalonRemmendation
- milesToMeters(distance): string
- outputToFile(outputFileName, businessList): string, list
- findNearby(APIKEY, distance, filter): string, list
HairColor
- colors: Map<str, int[]>
- alpha: double
- performHairSegmentation(session, input, output, img): imgMask
- changeColor(img, mask, color): img
+ changeHairColor(img): img
+ setAlpha(newAlpha): void
+ reset(): str
HairStyle
+ computeHairCoordinate(basePos, FacialCoordinates): double[]
+ computeRotationDegree(basePos, FacialCoordinates): double[]
FacialRecognition
+ detectMultiple: boolean
+ detectFaces(img): faceObject[]
+ reset(): str
MLModel
+ minConfidenceLevel: double
+ minDetectionConfidenceLevel: double
+ faceModel: mediapipeModel
+ hairModel: onnxModel
+ initFaceModel(): void
+ initHairModel(): void
+ getFaceModel(): mediapipeModel
+ getHairModel(): onnxModel
+ resetModel(name): str
Utility
- mean: list[double]
- std: list[double]
+ processImage(img) : img
+ maskImage(img) : img
+ toGreyScale(img): iimg
+ resizeImage(img, size): img
+ configMediaPipe() : void

Figure 1: UML Diagram

6 MIS of Controller Module

6.1 Module

M1 - Controller
Abstract Data Type Module

6.2 Uses

FacialRecognition (M2)
HairColor (M3)
HairStyle (M4)
SalonRecommendation (M5)
HairColorView (M8)
HairStyleView (M9)
SalonRecommendationView (M10)
HomeView (M11)
ErrorView (M13)
Camera (M12)

6.3 Syntax

6.3.1 Exported Constants

6.3.2 Exported Access Programs

Name	**In**	**Out**	**Exceptions**
initApp	-	-	-
switchToHairColorScreen	-	-	-
switchToHairStyleScreen	-	-	-
switchToErrorScreen	string	-	-
switchToSalonScreen	-	-	-
switchToMain	-	-	-
getCamera	-	Camera	-
resetState			

6.4 Semantics

6.4.1 State Variables

hairColorScreen := HairColorView
hairStyleScreen := HairStyleView
salonRecomScreen := SalonRecommendationView
camera := Camera

homeScreen := HomeView
launchScreen := launchView
errorScreen := errorView
facialRecognitionModel := FacialRecognition
hairColorModel := HairColor
hairStyleModel := HairStyle
salonRecommendationModel := SalonRecommendation
currentView := homeScreen

6.4.2 Environment Variables

6.4.3 Assumptions

6.4.4 Access Routine Semantics

initApp():

- transition:
  switchScreen(launchScreen)
  currentView.display()
- output:
- exception:

switchToHairColorScreen():

- transition: switchScreen(hairColorScreen)
- output:
- exception:

switchToHairStyleScreen():

- transition: switchScreen(hairStyleScreen)
- output:
- exception:

switchToErrorScreen(message):

- transition: switchScreen(errorScreen)
- output:
- exception:

switchToSalonScreen():

- transition: switchScreen(salonScreen)
- output:
- exception:

switchToMain():

- transition: switchScreen(homeScreen)
- output:
- exception:

getCamera():

- transition:
- output: camera
- exception:

resetState():

- transition: currentView.clear()
- output:
- exception:

6.4.5 Local Functions

getFacialCoordinates(img):

- input:
  img - inputImage
- transition:
- output:
  faces := facialRecognitionModel.detectFaces(img)
  return faces - faceObject[]
- exception:

changeHairColorPressed(img, color):

- input:
  img - inputImage
  color - rgb values
- transition:
- output:
  outImg := hairColorModel.changeHairColor(img, color)
  return outImg - image with chose hair color
- exception:

changeHairStylePressed(img, basePosition):

- input:
  img - inputImage
  basedPosition - camera base position
- transition:
- output:
  coordinates := getFacialCoordinates(img)
  rotationDegrees := hairStyleModel.computeRotationDegree(coordinates)
  return rotationDegrees - double[]
- exception:

switchScreen(view):

```
currentView.reset()
view.display()
currentView := view
```

binding():

```
HairColorView.backToHomeButton.event.pressed(switchScreen(homeScreen))
HairStyleView.backToHomeButton.event.pressed(switchScreen(homeScreen))
SalonRecommendationView.backToHomeButton.event.pressed(switchScreen(homeScreen))
ErrorView.backToHomeButton.event.pressed(switchScreen(homeScreen))
HairColorView.selectedColor.event.pressed(changeHairColorPressed(img, color))
HairStyleView.previousHairStyle.event.pressed(changeHairStylePressed(img, basePosi-
tion))
HairStyleView.nextHairStyle.event.pressed(changeHairStylePressed(img, basePosition))
```

7 MIS of Facial Recognition Module

7.1 Module

M2 - FacialRecognition
Abstract Object Module

7.2 Uses

MLModel (M6)
Utility (M7)

7.3 Syntax

7.3.1 Exported Constants

7.3.2 Exported Access Programs

Name	In	Out	Exceptions
detectFaces	image	list of face objects	InterruptException
detectMultipleFaces	boolean		
reset		string	

7.4 Semantics

7.4.1 State Variables

detectMultiple := true

7.4.2 Environment Variables

7.4.3 Assumptions

7.4.4 Access Routine Semantics

detectFaces(img):

- transition:
- output:
  processedImage = Utility.toGreyScale(img)
  if detectMultiple == true => MLModel.getFaceModel().process(processedImage)
  if detectMultiple == false => MLModel.getFaceModel().process(processedImage, max_faces=1)
  return results - a list of face objects detected in the input image

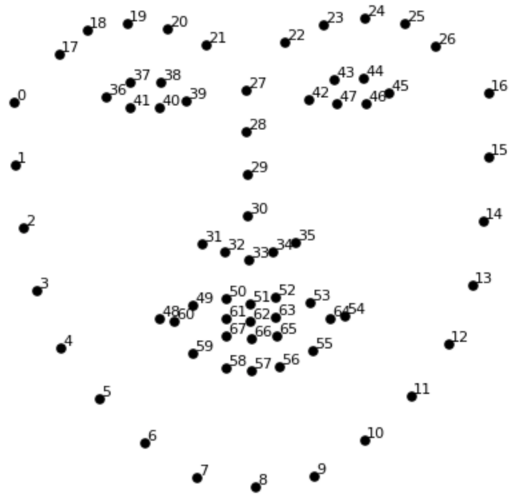


Figure 2: Face Landmarks within a face object

- exception: InterruptException := action terminated by the user

detectMultipleFaces(status):

- transition: detectMultiple := status
- output:
- exception:

reset():

- transition:
- output: message => MLModel.reset(hair)
- exception:

7.4.5 Local Functions

None

8 MIS of Hair Color Module

8.1 Module

M3 - HairColor Module
Abstract Object Module

8.2 Uses

MLModel (M6)
Utility (M7)

8.3 Syntax

8.3.1 Exported Constants

8.3.2 Exported Access Programs

Name	In	Out	Exceptions
changeHairColor	image, string	image	KeyErrorException
setAlpha	double		
reset		string	

8.4 Semantics

8.4.1 State Variables

colors - Map<str, int[]> - a mapping between the name of color and their rgb values
alpha - double - represents the ratio between the original image and the masked image

8.4.2 Environment Variables

8.4.3 Assumptions

8.4.4 Access Routine Semantics

changeHairColor(image, color):

- input:
  image - the copy of an original image
  color - the chosen hair color
- transition: N/A
- output:
  hairModelSession = utility.getHairModel()

mask = performHairSegmentation(hairModelSession, hairModelSession.inputName, hair-ModelSession.output image) - the image where the hair detected by the model is masked.
outputImg = changeColor(image, mask, color)
return outputImg - an image where the hair color of each person is changed to the specified color

- exception: InterruptException - the prediction and masking process is interrupted by the user

setAlpha(newAlpha):

- input: newAlpha - double - input alpha value for update
- transition: alpha := newAlpha - update the alpha value
- output: N/A
- exception: N/A

reset():

- transition:
- output: message => MLModel.reset(hair)
- exception:

8.4.5 Local Functions

performHairSegmentation(session, input, output, image):

- input: session - the onnx inference session that contains the input model
  input - list of integer - the input shape of the image
  output - list of integer - the expected output shape
  image - the copy of an original image
- transition: N/A
- output: mask - hair mask.

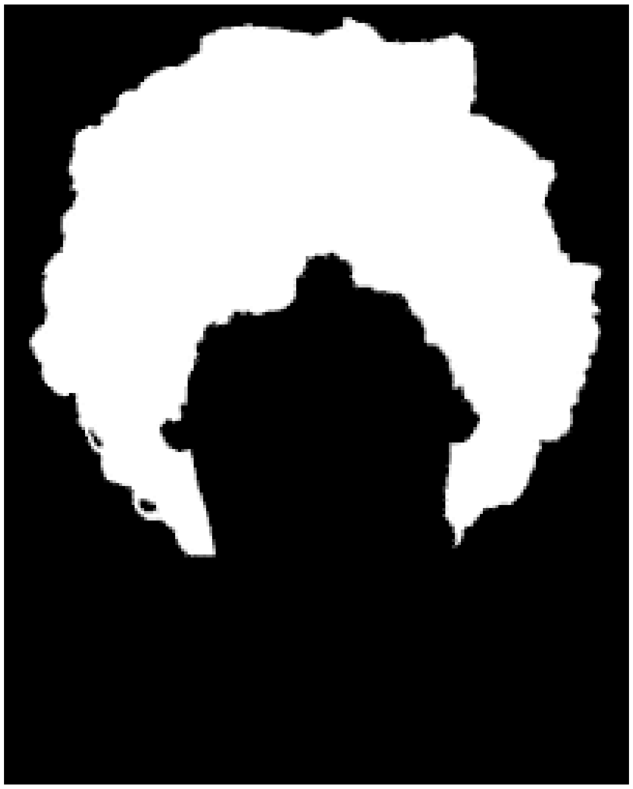


Figure 3: Hair Mask after running the pre-trained hair segmentation model

- exception: KeyErrorException - the specified color is not in the color map

changeColor(img, mask, color):

- input: img - the original image, mask - the masked image generated from hair segmentation, color - color's name as a string
- transition: N/A
- output: an image where the original image is mixed with the masked image.
- exception: KeyErrorException - the specified color is not in the color map

9 MIS of Hair Style Module

9.1 Module

M9 - FacialRecognition
Abstract Object Module

9.2 Uses

MLModel (M6)
Utility (M7)

9.3 Syntax

9.3.1 Exported Constants

9.3.2 Exported Access Programs

Name	In	Out	Exceptions
computeHairCoordinate	list[double], list[double]	list[double]	
computeRotationDegree	list[double], list[double]	list[double]	

9.4 Semantics

9.4.1 State Variables

9.4.2 Environment Variables

9.4.3 Assumptions

9.4.4 Access Routine Semantics

computeHairCoordinate(basePosition, facialCoordinates):

- input: basePosition - the basePosition of the camera setting in a tuple
  facialCoordinates - a list of coordinates of the facial features
- transition: N/A
- output: output the desired position to place the hairstyle centered at a coordinate, computed based on the base position and facial coordinates.
- exception: InterruptException := action terminated by the user

computeRotationDegree(basePosition, facialCoordinates):

- input: basePosition - the basePosition of the camera setting in a tuple
  facialCoordinates - a list of coordinates of the facial features
- transition: N/A
- output: output the desired rotation of the hairstyle when being placed on the user's face, computed based on the base position and facial coordinates.

9.4.5 Local Functions

10 MIS of Salon Recommendation Module

10.1 Module

M5: Salon Recommendation Module
Abstract Object Module

10.2 Uses

10.3 Syntax

10.3.1 Exported Constants

10.3.2 Exported Access Programs

Name	**In**	**Out**	**Exceptions**
milesToMeter	int	int	-
outputToFile	string, list	file	-
findNearby	string, list	string, list	IndexOutOfRange

10.4 Semantics

10.4.1 State Variables

10.4.2 Environment Variables

10.4.3 Assumptions

10.4.4 Access Routine Semantics

milesToMeter(distance):

- input: The input to the function would be the distance between the two points in miles.
- transition:
- output: output the distance between two locations from miles to meters. used to give the user more precise measurements, or to conform to international standards.
- exception: InvalidValueException - The value is negative or irational numbers

outputToFile(outputFileName, businessList):

- input: The input to the function would be the self defined output filename and the generated list after sorting the salon choices.
- transition:

- output: output the file that stores the customized hair salon information after sorting.

findNearby(APIKEY, distance, filter):

- input: The input to the function would be the google API KEY, the distance user want to search for, and the filter information used to filter hair salons.
- transition:
- output: output the sorted business list
- exception: IndexOutOfRange: the value of distance is invalid

10.4.5 Local Functions

11 MIS of ML Model Module

11.1 Module

M6 - MLModel
Abstract Object Module

11.2 Uses

Mediapipe (External Module)
Onnx (External Module)

11.3 Syntax

11.3.1 Exported Constants

11.3.2 Exported Access Programs

Name	In	Out	Exceptions
initFaceModel			InterruptException
initHairModel			
getFaceModel		mediapipeModel	
getHairModel		onnxModel	
resetModel	string	string	

11.4 Semantics

11.4.1 State Variables

minConfidenceLevel := 0.5
minDetectionConfidence := 0.5
modelFilePath := filePath (path to the pre-trained model)
faceModel := null
hairModel := null

11.4.2 Environment Variables

11.4.3 Assumptions

11.4.4 Access Routine Semantics

initFaceModel():

- transition: faceModel := mediapipe.FaceMesh(minConfidenceLevel, minDetectionConfidence)

- output:
- exception:

initHairModel():

- transition: hairModel := onnxruntime.InferenceSession(modelFilePath)

- output:
- exception:

getFaceModel():

- transition:
- output: faceModel
- exception:

getHairModel():

- transition:
- output: hairModel
- exception:

resetModel(name):

- transition:
  if name == "face" then initFaceModel()
  else if name == "hair", then initHairModel()
- output:
- exception:

11.4.5 Local Functions

12 MIS of Utility Module

12.1 Module

M7 - Utility Module
Library

12.2 Uses

OpenCV (External Module) Numpy (External Module)

12.3 Syntax

12.3.1 Exported Constants

12.3.2 Exported Access Programs

Name	**In**	**Out**	**Exceptions**
processImage	image, list[int]	tensor	illegalArgumentException
maskImage	image, image	image	illegalArgumentException
toGreyScale	image	image	
resizeImage	image, list[int]	image	illegalArgumentException

12.4 Semantics

12.4.1 State Variables

mean - list[double] - the mean values of trained images, used to normalize the images
std - list[double] - the standard deviation values of trained images, used to normalize the images

12.4.2 Environment Variables

12.4.3 Assumptions

12.4.4 Access Routine Semantics

processImage(image, input_size):

- input: image - the original input image in the form of 3-dimensional array, input_size - a tuple represents the input size the ML model requires
- transition: N/A

- output:
  OpenCV.convertColor(image, BGR2RGB) - convert the image to RGB format
  resizeImage(image, input_size) - convert image to input size
  image = (image / 255 - mean) / std - normalize the image
  Numpy.expandDimension(image, axis=0) - expand one dimension to a tensor
  output a image tensor ready for process with the model
- exception: illegalArgumentException - illegal input size for resizing

maskImage(original_img, mask):

- input: original_img - the original image in the form of 3-dimensional array, mask - the masked image in the form of 3-dimensional array with same dimension as original
- transition: N/A
- output:
  OpenCV.bitwise_or(original_img, original_img, mask) - apply masking to the original image with the given mask.
  output a masked image.
- exception: illegalArgumentException - original image has different size from the masked image.

toGreyScale(image):

- input: image - the input image to be converted to grey scale
- transition: N/A
- output:
  OpenCV.convertColor(image, BGR2GRAY) - convert the input image to an grey scale image
  output the greyscaled image
- exception:

resizeImage(image, shape):

- input: image - the input image
  shape - a tuple represents the width / height to be reshaped into.
- transition: N/A
- output:
  Numpy.reshape(image, shape) - reshape the image
  output an reshaped image
- exception: illegalArgumentException - illegal input size for resizing

12.4.5 Local Functions

None

13 MIS of Hair Color View Module

13.1 Module

M8 - HairColorView
Abstract Object Module

13.2 Uses

None

13.3 Syntax

13.3.1 Exported Constants

None

13.3.2 Exported Access Programs

Name	In	Out	Exceptions
buttonActionRed		updateCameraView	-
buttonActionBrown		updateCameraView	-
buttonActionBackToHome		backToHomePage	-

13.4 Semantics

13.4.1 State Variables

None

13.4.2 Environment Variables

Screen, Camera, Buttons

13.4.3 Assumptions

None

13.4.4 Access Routine Semantics

buttonActionRed():

- transition: None
- output: the camera view with user's hair color changed by calling Local function updateHairColor(red).

- exception: None

buttonActionBrown():

- transition: None
- output: the camera view with user's hair color changed by calling Local function updateHairColor(Brown).
- exception: None

buttonActionBackToHome():

- transition: None
- output: Go back to home page interface.
- exception: None

13.4.5 Local Functions

updateHairColor(color):

- transition: None
- output: the camera view with user's hair color changed.
- exception: None

14 MIS of Hair Style View Module

14.1 Module

M9 - HairStyleView
Abstract Object Module

14.2 Uses

HairStyleModule

14.3 Syntax

14.3.1 Exported Constants

None

14.3.2 Exported Access Programs

Name	In	Out	Exceptions
buttonActionShort		updateCameraView	-
buttonActionTail		updateCameraView	-
upDateCoordinate		updateCoorinates and angle	-
buttonActionBackToHome		backToHomePage	-

14.4 Semantics

14.4.1 State Variables

Double[][]: coordinates(Represent the coordinates to put the hair model) Double: A]angle(Represent the angle that the hair model needs to turn)

14.4.2 Environment Variables

Screen, Camera, Buttons

14.4.3 Assumptions

None

14.4.4 Access Routine Semantics

updateCoordinate():

- transition: coordinates = HairStyleModule.getCoordinates()
- output: None
- exception: None

updateAngle():

- transition: coordinates = HairStyleModule.getAngle()
- output: None
- exception: None

buttonActionShort():

- transition: None
- output: the camera view with user's hair style changed by calling Local function upDateHairColor(Short).
- exception: None

buttonActionTail():

- transition: None
- output: the camera view with user's hair style changed by calling Local function upDateHairColor(Tail).
- exception: None

buttonActionBackToHome():

- transition: None
- output: Go back to the home page interface.
- exception: None

14.4.5 Local Functions

updateHairStyle(style, coordinates, angle):

- transition: None
- output: the camera view with user's hair style changed.
- exception: None

15 MIS of Salon Recommendation View Module

15.1 Module

M10 - SalonRecommendationView Abstract Object Module

15.2 Uses

None

15.3 Syntax

15.3.1 Exported Constants

None

15.3.2 Exported Access Programs

Name	**In**	**Out**	**Exceptions**
showBarbers	void	View	

15.4 Semantics

15.4.1 State Variables

addressInput: String
map: GoogleMapsObject
backToHome: UINavigationButton

15.4.2 Environment Variables

Screen

15.4.3 Assumptions

The map object is successfully generated from GoogleMap API before the showBarbers function is called.

15.4.4 Access Routine Semantics

showBarbers():

- transition: None
- output: out := View
- exception: None

15.4.5 Local Functions

onChange(event):

- transition: addressInput := event.text
- output: out := None
- exception: None

onClick():

- transition: None
- output: out := navigate back to home page
- exception: None

generateMap():

- transition: map := new GoogleMapsObject from Google Map API
- output: out := None
- exception: None

16 MIS of Home View Module

16.1 Module

M11 - HomeView Module
UIView Module

16.2 Uses

Camera (M13)
SwiftUI (External Module)
RealityKit (External Module)
ARKit (External Module)

16.3 Syntax

16.3.1 Exported Constants

16.3.2 Exported Access Programs

16.4 Semantics

16.4.1 State Variables

body - View - View of the home interface, a swift object. currentMode - String - A string that describes current mode
toHairColor - UIButton - Navigate to Haircolor view
toHairStyle - UIButton - Navigate to Hairstyle view
toSalonRecom - UIButton - Navigate to Salon Recommendation view

16.4.2 Environment Variables

16.4.3 Assumptions

16.4.4 Access Routine Semantics

16.4.5 Local Functions

17 MIS of Error View Module

17.1 Module

M12 - ErrorView Abstract Object Module

17.2 Uses

None

17.3 Syntax

17.3.1 Exported Constants

None

17.3.2 Exported Access Programs

Name	**In**	**Out**	**Exceptions**
showError	String, String	UIAlertView	-

17.4 Semantics

17.4.1 State Variables

None

17.4.2 Environment Variables

Screen

17.4.3 Assumptions

None

17.4.4 Access Routine Semantics

showError(title, message):

- transition: None
- output: the UIAlertView with the input title and the input message
- exception: None

17.4.5 Local Functions

None

18 MIS of Camera Model Module

18.1 Module

M13 - CameraModel
Abstract Object Module

18.2 Uses

AVCapture (External Module)

18.3 Syntax

18.3.1 Exported Constants

18.3.2 Exported Access Programs

Name	In	Out	Exceptions
Check			AVCaptureRequestDenial
setUp			AVCaptureConfigureError
takePic			
reTake			
savePic		Data	UIImageWriteToSavedPhotosAlbum
photoOutput			

18.4 Semantics

18.4.1 State Variables

AVCapturePhotoCaptureDelegate := {
isTaken := false
output := null
session := AVCaptureSession()
alert := false
isSaved := false
picData - Data - stores camera session image data
}

18.4.2 Environment Variables

18.4.3 Assumptions

18.4.4 Access Routine Semantics

Check():

- transition: AVCaptureDevice.authorizationStatus(for: .video) ? setUp() : AVCaptureDevice.requestAccess(for: .video)
- output:
- exception: AVCaptureRequestDenial

setUp():

- transition: AVCaptureDevice := AVCaptureDevice.default(.builtInWideAngleCamera, for: .video)
- output:
- exception: AVCaptureConfigureError

takePic():

- transition:
  isTaken := !self.isTaken
  output.capturePhoto := (with: AVCapturePhotoSettings(), delegate: self)
- output:
- exception:

reTake():

- transition:
  isTaken := !self.isTaken
  isSaved := false
  self.output.capturePhoto := (with: AVCapturePhotoSettings(), delegate: self)
- output:
- exception:

savePic():

- transition:
  isSaved := true

- output: picData
- exception: UIImageWriteToSavedPhotosAlbumException

photoOutput():

- transition:
  picData := AVCapturePhotoOutput.photo.fileDataRepresentation()
- output:
- exception:

18.4.5 Local Functions

19 Appendix