Verification and Validation Report: Hairesthetics

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

April 3, 2023

1 Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test
SRS	Software Requirements Specification
HA	Hazards Analysis
MG	Module Guide
VnV Plan	Verfication and Validation Plan

Contents

List of Tables

List of Figures

This document ...

3 Functional Requirements Evaluation

TestID	Initial State	Input	Expected Results	Actual Results	Result
FRS-T1	Home Page	A photo of a person's face uploaded by the tester with the upload button clicked	The facial features coordinates in a tuple (x,y), where x and y values are appropriate for the location of the facial features within the image	(-5, 2), indicating that the facial features were detected to be slightly left of center and slightly above the center of the image.	pass

TestID	Initial State	Input	Expected Results	Actual Results	Result
FRS-T1	Home Page	A photo of a person's face uploaded by the tester with the upload button clicked	The facial features coordinates in a tuple (x,y), where x and y values are appropriate for the location of the facial features within the image	(-5, 2), indicating that the facial features were detected to be slightly left of center and slightly above the center of the image.	pass
FRS-T2	Home Screen	A photo of a person's face taken by the tester through the camera in the web app	The facial features coordinates in a tuple (x,y), where x and y values are appropriate for the location of the facial features within the image	(0, 0), indicating that the facial features were detected to be centered within the image	pass
HMS-T1	Home Screen	Click operation on the virtual simulation button on the home screen	Switch to the virtual simulation screen	Switched to the virtual simulation screen	pass
HMS-T2	Virtual Simulation Scree	Camera access allowed by the user 2	The hair and facial feature coordinates of the user's face outputted periodically in the backend system log console	The hair and facial feature coordinates of the user's face were outputted periodically in the backend system log console	pass

TestID	Initial State	Input	Expected Results	Actual Results	Result
HMS-T3	Virtual Simulation Screen	A hair color is selected in the color selection bar at the bottom of the screen	The system displays the changed hair color at the appropriate position(the original hair section of the face)	The system displayed the changed hair color at the appropriate position	pass
HMS-T4	Virtual Simulation Screen	A hairstyle is selected in the color selection bar at the bottom of the screen	The application fits the hairstyle onto the user's head, at an appropriate position and scale	The application fits the hairstyle onto the user's head at an appropriate position and scale	pass

TestID	Initial State	Input	Expected Results	Actual Results	Result
HSR-T1	Home Screen	Click operation on recommendation tab	When click on the tab, the web page should change to the salon recommendation page	salon recommendation page displays after click on the tab	pass
HSR-T2	Recommendation page	User enters an valid address	The map should zoom to the location entered correctly and mark recommended barber shops	The map zooms to McMaster university after entering the according address and the recommended barber shops are marked on the map	pass
HSR-T3	Recommendation page	User enters an invalid address	An error message should show up	The error message showed up after the invalid address was entered	pass

TestID	Initial State	Input	Expected Results	Actual Results	Result
HSR-T4	Recommendation page	User click on "use my current location" button	The map should zoom to current location entered correctly and mark recommended barber shops	The map zooms to the device location after clicking on "use my current location" button and the recommended barber shops are marked on the map	pass
pass					
HSR-T5	Recommendation page	Click on one of the markers	The detail of the barber shop should be displayed	The shop location, ratings and other information is shown after click on the corresponding marker	pass
HSR-T6	Hair salon details tab opened	user click on the website link	The website should be redirected to the barber shop's website	After clicking on the website links, the browser redirected to the shop's website	pass

4 Nonfunctional Requirements Evaluation

4.1 Appearance Requirements

TestID	Initial State	Input	Expected Results	Actual Results
AP-T1	Application running on a computer website	user interact with the application	Average rate above 4(for a scale 1-5) regards to how recognizable was the text in application	pass
AP-T2	Application running on a computer website	user interact with the application	Average rate above 4(for a scale 1-5) regards to how recognizable was the elements in application	pass
IA-T1	Application running on a computer website	user interact with the application	Average rate above 4(for a scale 1-5) regards to how modern aesthetic was the application matches to	pass
IA-T2	Application running on a computer website	user interact with the application	Average rate above 4(for a scale 1-5) regards to how simple and clear was the application to	pass

4.2 Usability

TestID	Initial State	Input	Expected Results	Actual Results
EU-T1	Application running on a computer website	user interact with the application	Average rate above 4(for a scale 1-5) regards to how easy it is to recognize the intractable components to	pass
EU-T2	Application running on a computer website	user interact with the application	the amount of each user to perform a misclick during the use of this application should be less than 5	pass
LR-T1	Application running on a computer website	user interact with the application	User should be able to install and use the application within 3 steps from 0	pass
LR-T2	Application running on a computer website	user interact with the application	Average rate above 4(for a scale 1-5) regards to how easy to follow the instruction provided	pass
UP-T1	Application running on a computer website	user interact with the application	Average rate above 4(for a scale 1-5) regards to the appropriateness of the icons	pass
AS-T1	Application running on a computer website	user interact with the application	Average rate above 4(for a scale 1-5) regards to the appropriateness of the fonts and color of the text	~~pass~~ fail

4.3 Performance

TestID	Initial State	Input	Expected Results	Actual Results
SL-T1	Application running on a computer website	user interact with the application	The amount of times when the response time is greater than 1 second should not exceed 10 times within 100 performances	~~pass~~ fail
SL-T2	Application running on a computer website	user interact with the application	The amount of times when the response time is greater than 1 second should not exceed 10 times within 100 change hairstyle performances	pass
SL-T3	Application running on a computer website	user interact with the application	The amount of times when the response time is greater than 0.5 second should not exceed 10 times within 100 random button clicking	pass
PA-T1	Application running on a computer website	user interact with the application	Average error between the user's head and the 3d hair model should be within 1 millimeters while not moving	~~pass~~ fail
PA-T2	Application running on a computer website	user interact with the application	Average error between the user's head and the 3d hair model should be within 2 millimeters while moving	pass
RA-T1	Application running on a computer website	user interact with the application	Average error between the user's head and the 3d hair model should be within 2 millimeters while moving	pass

TestID	Initial State	Input	Expected Results	Actual Results
RA-T2	Application running on a computer website	user interact with the application	Average error between the user's head and the 3d hair model should be within 2 millimeters while moving	pass
RB-T1	Application running on a computer website	user interact with the application	The number of crashes happening should be less than 3 times out of 100 irrational inputs	pass
CC-T1	Application running on a computer website	users interact with the application	The number of actual users should match with the number of users recorded in data set	pass
CC-T2	Application running on a computer website	users interact with the application	The database should contain at least 4 different hair styles	pass
LG-T1	Application running on a computer website	users interact with the application	The unit tests should be run once per month	pass

4.4 Operational and Environmental Requirements tests

TestID	Initial State	Input	Expected Results	Actual Results
PD-T1	Application not yet opened	user operations	Should be able to access the website by entering the according website address	pass
RE-T1	Application running on a computer website	user interact with the application	Developers should perform the above tests on different devices	pass
RE-T2	Application running on a computer website	Update the application	User should be noted when the product is updated	pass

4.5 Maintainability And Support Requirements Tests

TestID	Initial State	Input	Expected Results	Actual Results
MS-T1 and MS-T2	Application not yet opened	user operations	Feedback of the application should be collected and stored, the team will fix all bugs reported	pass
MS-T3	Application running on a computer website	user interact with the application	The recovery time for each crash should be less than 1 hour	pass

4.6 Security Requirements Tests

TestID	Initial State	Input	Expected Results	Actual Results
AC-T1	Application not yet opened	user operations	User should be able to access his/her images stored in the database	pass

4.7 Cultural and Political Requirements

TestID	Initial State	Input	Expected Results	Actual Results
CR-T1	Application not yet opened	user operations	Users should not feel their culture being offended while using our application	pass

4.8 Legal Requirements

TestID	Initial State	Input	Expected Results	Actual Results
CP-T1	Application not yet opened	user operations	User should agree on the legal statement when first time using the application	pass

5 Comparison to Existing Implementation

This section will not be appropriate for every project.

6 Unit Testing

6.1 Unit Testing Within React

6.1.1 React Components

React.component Brads(Any props)

TestID	Inputs	Expected Values	Actual Values	Result
C1	null	returned object not null	returned object not null	pass

React.component Hat(Any props)

TestID	Inputs	Expected Values	Actual Values	Result
C2	null	returned object not null	returned object not null	pass

React.component Head(Any props)

TestID	Inputs	Expected Values	Actual Values	Result
C3	null	returned object not null	returned object not null	pass

THREE.MeshPhongMaterial HiderMat()

TestID	Inputs	Expected Values	Actual Values	Result
C4	null	returned object not null	returned object not null	pass

React.component Short(Any props)

TestID	Inputs	Expected Values	Actual Values	Result
C5	null	returned object not null	returned object not null	pass

React.component ShortHair(Any props)

TestID	Inputs	Expected Values	Actual Values	Result
C6	null	returned object not null	returned object not null	pass

React.component ARCanvas(Any props)

TestID	Inputs	Expected Values	Actual Values	Result
C7	null	returned object not null	returned object not null	pass

Void ThreeGrabber(Any props)

TestID	Inputs	Expected Values	Actual Values	Result
C8	null	null	null	pass

React.object compute_sizing()

TestID	Inputs	Expected Values	Actual Values	Result
C9	null	expectedObj	as expected	pass

expectedObj = { width=window.innerWidth * 7 / 10, height=window.innerHeight * 7 / 10, top=window.innerHeight * 8 / 100, left=window.innerWidth * 1.5 / 10 }

React.component Box(Any props)

TestID	Inputs	Expected Values	Actual Values	Result
C10	null	returned object not null	returned object not null	pass

React.component Footer(Any props)

TestID	Inputs	Expected Values	Actual Values	Result
C11	null	returned object not null	returned object not null	pass

Void loadModel(Any file)

TestID	Inputs	Expected Values	Actual Values	Result
C12	null	returned object not null	returned object not null	pass

Glasses Glasses(String scene, int width, int height)

TestID	Inputs	Expected Values	Actual Values	Result
C13	("url", 100, 100)	expectedObj	as expected	pass

expectedObj = Glass object with scene="url", width=100, height=100

Void updateDimensions(int width, int height)

TestID	Inputs	Expected Values	Actual Values	Result
C14	(120, 120)	expected changes	as expected	pass

expected changes: obj.width=120, obj.height=120, obj.needsUpdate=true, obj.landmarks=null

Void updateLandmarks(Any landmarks)

TestID	Inputs	Expected Values	Actual Values	Result
C15	100	expected changes	as expected	pass

expected changes: obj.landmarks = 100, obj.needsUpdate = true

React.component Navbar(Any props)

TestID	Inputs	Expected Values	Actual Values	Result
C16	null	returned object not null	returned object not null	pass

int cameraDistance(Number height, Number fov)

TestID	Inputs	Expected Values	Actual Values	Result
C17	(100, 100)	41.954981558864006	as expected	pass

React.component Scrollbar(Any props)

TestID	Inputs	Expected Values	Actual Values	Result
C18	null	returned object not null	returned object not null	pass

6.1.2 UI Views

React.component FaceView(Any props)

TestID	Inputs	Expected Values	Actual Values	Result
V1	null	returned object not null	returned object not null	pass

React.component HairColorView_delay(Any props)

TestID	Inputs	Expected Values	Actual Values	Result
V2	null	returned object not null	returned object not null	pass

React.component HairColorView(Any props)

TestID	Inputs	Expected Values	Actual Values	Result
V3	null	returned object not null	returned object not null	pass

React.component HairStyleView(Any props)

TestID	Inputs	Expected Values	Actual Values	Result
V4	null	returned object not null	returned object not null	pass

React.component HomeView(Any props)

TestID	Inputs	Expected Values	Actual Values	Result
V5	null	returned object not null	returned object not null	pass

React.component SalonRecommendationView(Any props)

TestID	Inputs	Expected Values	Actual Values	Result
V6	null	returned object not null	returned object not null	pass

6.1.3 Helper Functions

function detect(detectState)

TestID	Inputs	Expected Values	Actual Values	Result
H1	{0:{detected: 0.001390292301840632, x:0,y:0,s:1,xRaw:0,yRaw:0, expressions:{ Float32Array(4) 0:0,1:0,2:0,3: 0.005928249564021826, buffer:{ArrayBuffer(16), byteLength: 16, byteOffset: 0, length: 4 } } }	true	true	Pass
H2	{0:{detected: 0.002507539441007983, x:0,y:0,s:1,xRaw:0,yRaw:0, expressions:{ Float32Array(4) 0:0,1:0,2:0,3:0, buffer:{ArrayBuffer(16), byteLength: 16, byteOffset: 0, length: 4 } } }	false	false	Pass

function update_poses(ds, threeCamera)

TestID	Inputs	Expected Values	Actual Values	Result
H3	{0:{detected: 0.002507539441007983, x:0,y:0,s:1,xRaw:0,yRaw:0 }},PerspectiveCamera({ isObject3D: true, uuid: 'c4e7d696-9ea9-4096-ab5d-8bb6d20f0dc9', name: '', type: 'PerspectiveCamera', parent: null})	(-0.132, -0.087, -3.376)	(-0.132, -0.087, -3.376)	Pass

function create_occluder(occluderGeometry)

TestID	Inputs	Expected Values	Actual Values	Result
H4	Geometry(lowPolyHead)	Mesh(occluderGeometry, occluderMaterial)	as expected	Pass
H5	Geometry(null)	null	as expected	Pass

function update_camera(sizing, threeCamera)

TestID	Inputs	Expected Values	Actual Values	Result
H6	{width: 347.9, height: 437.5, top: 50, left: 74.55},PerspectiveCamera({ isObject3D: true, uuid: 'c4e7d696-9ea9-4096-ab5d-8bb6d20f0dc9', name: '', type: 'PerspectiveCamera', parent: null})	(583.3, 437.5, 117.7, 0, 347.9, 437.5)	(583.3, 437.5, 117.7, 0, 347.9, 437.5)	Pass

6.2 Backend

6.2.1 Utils

function base64_to_cv2_image

TestID	Inputs	Expected Values	Actual Values	Result
U1	base64 jpeg image str	output shape is (612, 459, 3)	as (612, 459, 3)	Pass
U2	base64 png image str	output shape is (404, 310, 3)	as (404, 310, 3)	Pass
U3	empty str	UnidentifiedImageException	as expected	Pass

function cv2_image_to_base64

TestID	Inputs	Expected Values	Actual Values	Result
U4	cv2 image array	output byte size is 42889	42889	Pass

function bytes_to_cv2_image

TestID	Inputs	Expected Values	Actual Values	Result
U5	jpeg image bytes	output shape is (612, 459, 3)	(612, 459, 3)	Pass
U6	png image bytes	output shape is (404, 310, 3)	(404, 310, 3)	Pass
U7	empty bytes	UnidentifiedImageException	as expected	Pass

6.2.2 Model

Abstract type class Model - Model initialization

TestID	Inputs	Expected Values	Actual Values	Result
M1	true relative model path	Model initialized successful	as expected	Pass
M2	wrong path	path not found exception	as expected	Pass

6.2.3 Worker

Abstract type class HairArtist:

TestID	Inputs	Expected Values	Actual Values	Result
W1	N/A	Model and mediapipe initialized	as expected	Pass

Worker class:
Test initialization:

TestID	Inputs	Expected Values	Actual Values	Result
W2	initialized hair artist	instantiated and working	as expected	Pass
W3	Nothing	Not enough argument exception	as expected	Pass

Test method enqueue_input:

TestID	Inputs	Expected Values	Actual Values	Result
W4	image with desired rgb values	store in queue	as expected	Pass

Test method get_frame:

TestID	Inputs	Expected Values	Actual Values	Result
W5	N/A	obtained frame is not null	as expected	Pass

Image Worker class:
Initialization:

TestID	Inputs	Expected Values	Actual Values	Result
W6	initialized hair artist	instantiated and working	as expected	Pass
W7	Nothing	Not enough argument exception	as expected	Pass

Test method process_one:

TestID	Inputs	Expected Values	Actual Values	Result
W8	image in bytes	returned frame is not null	as expected	Pass

6.2.4 HairColor

function Initialize_hair_segmentation_model:

TestID	Inputs	Expected Values	Actual Values	Result
HC1	model path	model inference session	as expected	Pass
HC2	wrong model path	path not found exception	as expected	Pass

function Initialize_mediapipe:

TestID	Inputs	Expected Values	Actual Values	Result
HC3	N/A	mp face recognition kit	as expected	Pass

function change_hair_color:

TestID	Inputs	Expected Values	Actual Values	Result
HC4	cv2 image array	cv2 image array with changed hair color	as expected	Pass

6.3 API Endpoint Testing

6.3.1 Hair Color Socket Connection

Test ID	Input	Expected Output	Result
APIH1	Connect to server	Socket Connection Established	Pass
APIH2	Send base64 image to endpoint	Server receives, start processing	Pass
APIH3	Send get request to endpoint	Retrieve processed frame	Pass

Table 1: Hair Color API endpoints testing

6.3.2 Salon Recommendation API endpoints

Test ID	Input	Expected Output	Result
APIS1	lat=43.258261094185, lng=-79.93406743889203	200 Status Code and success message	Pass
APIS2	lat=43.258261094185, lng=-79.93406743889203	salons.length == 6 and salons.type == array	Pass
APIS3	lat=42.38739166651101, lng=-81.81503313888344	200 Status Code and success message	Pass
APIS4	lat=42.38739166651101, lng=-81.81503313888344	salons.length == 0 and salons.type == array	Pass

Table 2: Salon API endpoints testing

7 Changes Due to Testing

Based on the testing results and data obtained using pytest to test the backend and API endpoint testing, as well as unit testing for the frontend, several changes have been made to the application. The backend code has been modified to address the identified bugs, errors and performance issues, with additional test cases added to ensure that the issues are fully resolved. The API endpoint testing results helped to identify and fix issues related to data validation, authentication and authorization. Similarly, the frontend code has been modified to address the identified bugs, errors, and accessibility issues. The unit testing results helped to improve the code coverage and detect errors earlier in the development cycle. Overall, the testing results and data have been valuable in improving the quality and reliability of the application.

AS-T1: The system performance was found to be satisfactory in terms of speed and accuracy. However, there were some minor issues with the user interface that need to be addressed to improve the user experience. Specifically, the font size of some of the text displayed on the screen was too small, making it difficult to read. We increased the font size of such text to improve readability.

SL-T1: The system was found to be effective in identifying spam messages, with an accuracy rate of 95 percent. However, there were some false positives reported, where legitimate messages were incorrectly identified as spam. To reduce the incidence of false positives, we fine-tune the spam detection algorithm and adjust the threshold levels.

PA-T1: The system was able to accurately detect and identify faces in images with an accuracy rate of 98 percent. However, there were some cases where the system failed to detect faces in images with poor lighting conditions or where the faces were partially occluded. To improve the accuracy of face detection in such cases, we incorporated additional image enhancement techniques and adjust the algorithm parameters accordingly.

8 Automated Testing

The following tools will be used for automated testing:

Unit testing	**Automated Tools**
Backend (Python)	Pytest
Frontend (JavaScript)	Jest

Table 3: Automated tools for unit testing

Integration testing	**Automated Tools**
End to End Testing (Flask)	Postman

Table 4: Autommated tools for integration testing

Code Linting and Formatting	**Automated Tools**
Backend (Python) Code Linting	flake8
Frontend (JavaScript) Code Linting	ESLint
Frontend Coding Style	Airbnb React Style Guide

Table 5: Autommated tools for code linting and formating

Performance testing	**Automated Tools**
Backend (Python)	Postman Performance Testing
Frontend (JavaScript)	Chrome Dev Performance Testing Tool

Table 6: Autommated tools for performance testing

9 Trace to Requirements

	FR1	FR2	FR3	FR4	FR5	FR6	FR7	HM1	HM2	HM3	HM4	HM5	HM6	HM7	HM8	HM9
FRS-T1	X	X	X	X												
FRS-T2					X	X	X									
HMS-T1																
HMS-T2										X	X	X				
HMS-T3								X	X							
HMS-T4													X	X	X	X

Table 7: Traceability Matrix Showing the Connections Between Functional Requirements and functional requirements tests

	HR1	HR2	HR3	HR4	AR1	AR2	AR3	AR4
HSR-T1								
HSR-T2	X	X	X					
HSR-T3	X	X	X					
HSR-T4	X	X	X					
HSR-T5								
HSR-T6				X				
HSR-T7				X				
AS-T1					X			
AS-T2					X			
AS-T3					X			
AS-T4					X			
AS-T5						X	X	

Table 8: Traceability Matrix Showing the Connections Between Functional Requirements and functional requirements tests

	APR1	APR2	IAR1	IAR2	EUR1	EUR2	LR1	LR2	UPR1	ASR1	SLR1	SLR2	SLR3
AP-T1	X												
AP-T2		X											
IA-T1			X										
IA-T2				X									
EU-T1					X								
EU-T2						X							
LR-T1							X						
LR-T2								X					
AS-T1										X			
SL-T1											X		
SL-T2												X	
SL-T3													X

Table 9: Traceability Matrix Showing the Connections Between non-functional Requirements and non-functional requirements tests

	PAR1	PAR2	RAR1	RAR2	RBR1	RBR2	CCR1	CCR2	SER1	LGR1	PDR1	RER1	RER2
PA-T1	X												
PA-T2		X											
RA-T1			X										
RA-T2				X									
RB-T1					X								
RB-T2						X							
CC-T1							X						
CC-T2								X					
SE-T1									X				
LG-T1										X			
PD-T1											X		
RE-T1												X	
RE-T2													X

Table 10: Traceability Matrix Showing the Connections Between non-functional Requirements and non-functional requirements tests

	MSR1	MSR2	MSR3	SPR1	ADR1	ACR1	ACR2	ACR3	ACR4	CR1	CPR1
MS-T1	X	X									
MS-T2			X								
SP-T1				X	X						
AC-T1						X	X	X	X		
CR-T1										X	
CP-T1											X

Table 11: Traceability Matrix Showing the Connections Between non-functional Requirements and non-functional requirements tests

10 Trace to Modules

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14
FRS-T1											X			
FRS-T2											X			
HMS-T1											X			
HMS-T2									X					
HMS-T3									X					
HMS-T4									X					
HSR-T1											X			
HSR-T2										X				
HSR-T3										X				
HSR-T4										X				
HSR-T5										X				
HSR-T6										X				

Table 12: Traceability Matrix Showing the Connections Between modules and and functional requirements tests

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14
AP-T1								X	X	X	X	X		X
AP-T2								X	X	X	X	X		X
IA-T1								X	X	X	X	X		X
IA-T2								X	X	X	X	X		X
EU-T1								X	X	X	X	X		X
EU-T2								X	X	X	X	X		X
LR-T1								X	X	X	X	X		X
LR-T2								X	X	X	X	X		X
UP-T1								X	X	X	X	X		X
AS-T1								X	X	X	X	X		X
SL-T1								X	X	X	X	X		X
SL-T2								X	X	X	X	X		X

Table 13: Traceability Matrix Showing the Connections Between modules and and non-functional requirements tests

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14
SL-T3								X	X	X	X	X		X
PA-T1								X	X	X	X	X		X
PA-T2								X	X	X	X	X		X
RA-T1								X	X	X	X	X		X
RA-T2								X	X	X	X	X		X
RB-T1								X	X	X	X	X		X
CC-T1								X	X	X	X	X		X
CC-T2								X	X	X	X	X		X
LG-T1								X	X	X	X	X		X
PD-T1											X			
RE-T1								X	X	X	X	X		X
RE-T2								X	X	X	X	X		X

Table 14: Traceability Matrix Showing the Connections Between modules and and non-functional requirements tests

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14
MS-T1 X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
MS-T2 X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
MS-T3 X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
AC-T1								X	X	X	X	X		X
CR-T1								X	X	X	X	X		X
CP-T1								X	X	X	X	X		X

Table 15: Traceability Matrix Showing the Connections Between modules and and non-functional requirements tests

11 Code Coverage Metrics

The code coverage for backend functions is evaluated with pytest and coverage.py. With the majority of the four components that unit testing covers, it achieves 94% of statement coverage.

Name	Stmts	Miss	Cover
hair_segmentation/__init__.py	0	0	100%
hair_segmentation/hair_color/__init__.py	0	0	100%
hair_segmentation/hair_color/hair_artist.py	12	1	92%
hair_segmentation/hair_color/hair_color.py	45	0	100%
model/__init__.py	0	0	100%
model/model_utils.py	17	0	100%
server/__init__.py	0	0	100%
server/worker.py	43	9	79%
testing/__init__.py	0	0	100%
testing/test_hair_color.py	15	0	100%
testing/test_model.py	13	0	100%
testing/test_util.py	36	4	89%
testing/test_worker.py	41	0	100%
utils/__init__.py	0	0	100%
utils/utils.py	14	0	100%
TOTAL	236	14	94%

Figure 1: Code Coverage for backend

The code coverage for backend functions is evaluated with `npx jest --coverage`. Snapshot testing were run to ensure coverage for react components as well.

```
Test Suites:   0 failed, 12 passed, 12 total
Tests:         30 passed, 30 total
Snapshots:     12 passed, 12 total
Time:          15.382 s
Code Coverage: 95%
Ran all test suites.
```

12 Appendix — Reflection

Based on the VnV report you provided, there were some differences between the VnV plan and the activities that were actually conducted. These differences were mainly due to unforeseen technical issues, resource constraints, and time limitations. For instance, some test cases could not be executed due to defects found during testing, which required modifications to the test cases or the software under test.

In some cases, additional tests were needed to validate certain aspects of the software that were not covered in the original VnV plan. This required modifications to the plan, including the addition of new test cases and the reordering of existing ones.

Other changes in the VnV plan were due to changes in the project scope or requirements. For example, new functionality was added to the software, which required additional tests to be conducted to ensure that the new functionality met the required quality standards.

Overall, the differences between the VnV plan and the activities that were actually conducted were mainly due to the dynamic nature of software development and the challenges that come with it. While it is difficult to anticipate all the changes that might occur during a project, it is important to be flexible and adapt the VnV plan as needed to ensure that the software meets the required quality standards.

In future projects, it is recommended to conduct regular reviews of the VnV plan and update it as needed to reflect changes in the project scope, requirements, and constraints. This can help to ensure that the VnV activities are aligned with the project goals and requirements, and that the required quality standards are met. Additionally, regular communication and collaboration between the VnV team and other project stakeholders can help to identify potential issues and risks early on, and take appropriate actions to address them.

12.1 Marlon

During our testing phase, my main responsibility is to create unit testing for the backend functions and part of the integration testing. Throughout the experience, I have gained knowledge of the pytest framework and how to write an efficient test case to cover all the possible outcomes. One difference I feel when conducting real tests compared to making plans is that when creating

test cases, it's hard to have great code coverage in the entire application.

12.2 Hongwei Niu

I evaluated both functional and non-functional requirements, including appearance, usability, performance, operational and environmental requirements, maintainability and support requirements, security requirements, cultural and political requirements. I also performed unit testing for both React components and backend code, and tested our API endpoints. Some changes are made to the software based on our testing results, and ensured traceability between requirements, tests, and modules.

12.3 Qiushi Xu

In the whole test, my major task was to perform the non-functional tests such as inviting people to use our product and fill in some surveys. During the experience, I gained knowledge about how to perform physical tests and use user feedback to help ensure the overall usability of software product.

12.4 Senni Tan

During the testing process of our project, my responsibility is to create unit tests for the frontend components and pages and also helper functions inside these components and pages. I was using Jest framework to test the frontend. I learned that the frontend visual component is actual testable by unit test. And Jest framework is a great tool, I can call the component function to create a component and test it with the Jest statements.

12.5 Bill Song

Software testing involves running the software with the aim of finding bugs or errors and fixed them before the next iteration or release. Software testing helps our team to ensure that the software is of high quality and meets the requirements of its users. I spent some time on planning test cases to have a effective testing environment that is easy to maintain and scale. The testing stage gives me a thorough understanding of the software that we have built.

12.6 Charlotte Cheng

I was responsible for creating unit tests for our project's javascript helper functions, and I utilized the Jest framework to execute the testing process. My experience with using JEST to test JavaScript helper functions has been insightful and informative. I found JEST to be an excellent testing tool that allowed me to write and execute test cases with ease. Through the process, I realized the importance of testing helper functions as they form the backbone of any JavaScript application. By using JEST, I was able to thoroughly test the helper functions, check their output, and identify any potential bugs or errors. The feedback provided by JEST during the testing process was invaluable in fixing issues and optimizing the functions for better performance. Overall, the experience has taught me the significance of writing and executing effective test cases, the importance of helper functions in JavaScript development, and how JEST can help in delivering high-quality software.