

SW Engineering CSC648/848

***Mesekai***

real-time full-body 3D avatar and virtual room web application

Section 04 Team 04

Caelen Wang (Team Lead)

Vasudevan Venugopal (Frontend Lead)

Matthew Madore (Backend Lead)

Jose Miguel Atienza (Scrum Master)

Eugene San Juan (Git Master)

Mohammad Abdelrahman

“Milestone 4”

November 29, 2021

Revision History Table


## Product Summary

Name: *Mesekai*

Unique Features: A real-time 3D avatar web application using webcam for facial expressions, limbs, and fingers animation. Allowing users to select from a list of avatars and worlds.

URL: <https://mesekai.vercel.app/>

### P1 Features:

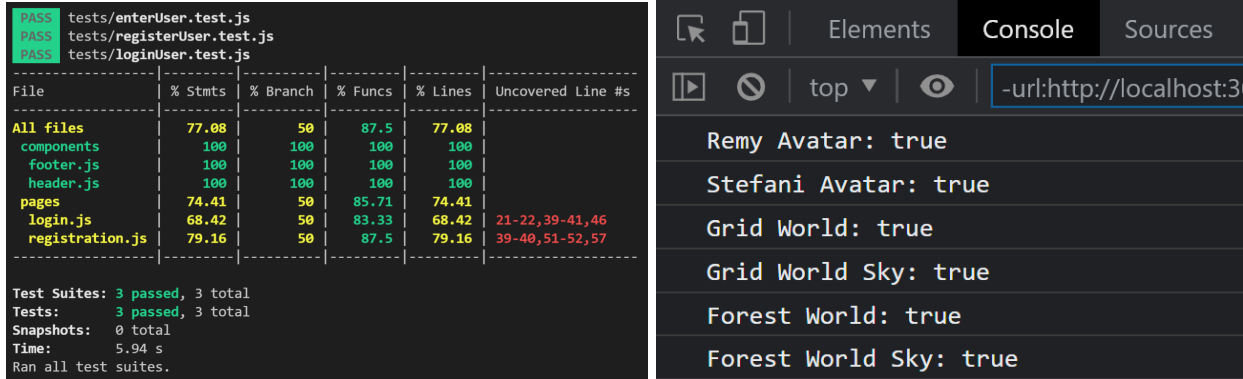
ID	Functional Requirement	Details
1	Account creation	Store User provided information such as username, email, and password in database
2	Log in and log out of account	Log in to account using credentials stored in database provided during account creation or password reset
3	Reset password to account	Link on login page that lets User enter their email address, to which a password reset link will be sent
4	Select Avatar from presets	User chooses from a list of available Avatars from Interface. Choice stored in Account: <ul style="list-style-type: none"><li>● Remy</li><li>● Stefani</li><li>● YBot</li><li>● ...</li></ul>
5	Select World from presets	User chooses from a list of available Worlds from Interface. Choice stored in Account <ul style="list-style-type: none"><li>● Grid</li><li>● Forest</li><li>● Room</li><li>● Castle</li><li>● Space</li></ul>
6	Avatar facial expressions tracking	User's face controls Avatar's face
7	Avatar limbs tracking	User's limbs controls Avatar's limbs
8	Avatar fingers tracking	User's fingers controls Avatar's fingers

## QA Testing

### I. Unit Test

User Stories to be tested:

1. Registration
2. Login
3. Avatar Presets Selection and Instantiation
4. World Presets Selection and Instantiation



The image shows two side-by-side screenshots. The left screenshot is a terminal window displaying Jest CLI output. It shows three tests passing: `tests/enterUser.test.js`, `tests/registerUser.test.js`, and `tests/loginUser.test.js`. Below this is a table of code coverage statistics. The right screenshot is a browser console showing a series of log messages indicating that various avatar and world presets have been successfully loaded or instantiated.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	77.08	50	87.5	77.08	
components	100	100	100	100	
footer.js	100	100	100	100	
header.js	100	100	100	100	
pages	74.41	50	85.71	74.41	
login.js	68.42	50	83.33	68.42	21-22, 39-41, 46
registration.js	79.16	50	87.5	79.16	39-40, 51-52, 57

Test Suites: 3 passed, 3 total  
Tests: 3 passed, 3 total  
Snapshots: 0 total  
Time: 5.94 s  
Ran all test suites.

Browser Console Log:

- Remy Avatar: true
- Stefani Avatar: true
- Grid World: true
- Grid World Sky: true
- Forest World: true
- Forest World Sky: true

Unit tests for login and registration are performed using Jest via `npm test`. The statement coverage for all files tested is 77.08%. Unit tests for Avatar and World loading and instantiation are performed in the browser due to Three.js's reliance on the browser instance. These tests can be accessed at <https://mesekai.vercel.app/unit-test>, and the results can be observed in the browser console. Advanced features such as avatar tracking were not tested due to the complexity of input and output data required.

## II. Integration Test

All P1 features are tested except for Test Case 3: Reset Password since the feature is still being implemented.

Test Case ID	1	Test Case Description	registration functionality			
Created By	Vasu	Reviewed By	Caelen	Version	4	
QA Tester's Log						
Tester's Name	Mohammad	Date Tested	November 22, 2021	Test Case (Pass/Fail/Not Executed)	Pass	
S #	Prerequisites:		S #	Test Data Requirement		
1	Access to a supported browser		1	username = "test1"		
2			2	email = "test1@test.com"		
3			3	password = "test1"		
4			4			
Test Conditions						
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended		
1	Go to <a href="https://mesekai.vercel.app/">https://mesekai.vercel.app/</a>	User should be directed to landing page	The landing page is shown	Pass		
2	Click on login tap	User should be redirected to login page	User is taken to the login page	Pass		
3	Click on Register Here!	User should be redirected to registration page	User is taken to the registration page	Pass		
4	Enter username, email and password to register (see test data table)	User should be registered in the database and redirect to landing page	A message of success pops up and then user is taken to homepage	Pass		
5	Enter a not unique username	User shouldn't be able to register in the database	A message of failure pops up and then user stay at registration page	Pass		

Test Case ID	2	Test Case Description	login functionality and authentication				
Created By	Vasu	Reviewed By	Caelen	Version	4		
QA Tester's Log							
Tester's Name	Vasu	Date Tested	November 22, 2022	Test Case (Pass/Fail/Not Executed)	Fail		
S #	Prerequisites:		S #	Test Data Requirement			
1	Access to a supported browser		1	username = "test3"			
2			2	password = "test3"			
3			3				
4			4				
Test Conditions							
Step #	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended		
1	Go to <a href="https://mesekai.vercel.app/">https://mesekai.vercel.app/</a>	User should be directed to landing page	The landing page is shown		Pass		
2	Click on login tab	User should be redirected to login page	User is taken to the login page		Pass		
3	Try to login with a random username and password	User should see a message "Login failure"	The message "Login failure" pops up after pressing the submit button		Pass		
4	Enter username and password that has been registered before (see test data table)	User should be authenticated and taken to landing page	User is taken to the landing page after seeing the message "Login success"		Pass		
5	Click on logout	User should be logout and the session should end	No session seems to be up, so the login tab still exists		Fail		

Test Case ID		4	Test Case Description	avatar loading and switching based on button click				
Created By	Caelen	Reviewed By	Jose	Version		4		
QA Tester's Log								
Tester's Name		Caelen	Date Tested		November 29, 2021	Test Case (Pass/Fail/Not Executed)		Pass
S #		Prerequisites:		S #		Test Data Requirement		
1		browser		1		cursor click		
Test Conditions								
Step #	Step Details		Expected Results		Actual Results		Pass / Fail / Not executed / Suspended	
1	wait for avatar to load		male avatar loads		male avatar loads		Pass	
2	click on "Stefani" button		avatar changes to female		avatar changes to female		Pass	
3	click on "Remy" button		avatar changes to male		avatar changes to male		Pass	

Test Case ID	5	Test Case Description	world loading and switching based on button click						
Created By	Jose	Reviewed By	Caelen	Version			4		
QA Tester's Log									
Tester's Name	Jose	Date Tested	November 29, 2021	Test Case (Pass/Fail/Not Executed)			Pass		
S #	Prerequisites:		S #	Test Data Requirement					
1	browser		1	cursor click					
Test Conditions									
Step #	Step Details	Expected Results	Actual Results			Pass / Fail / Not executed / Suspended			
1	wait for world to load	grid world loads	grid world loads			Pass			
2	click on "Forest" button	world changes to forest	world changes to forest			Pass			
3	click on "Grid" button	world changes to grid	world changes to grid			Pass			

<b>Test Case ID</b>	6	<b>Test Case Description</b>	avatar tracks user face			
<b>Created By</b>	Caelen	<b>Reviewed By</b>	Jose	<b>Version</b>	4	
<b>QA Tester's Log</b>						
<b>Tester's Name</b>	Caelen	<b>Date Tested</b>	November 29, 2021	<b>Test Case (Pass/Fail/Not Executed)</b>	Pass	
<b>S #</b>	<b>Prerequisites:</b>		<b>S #</b>	<b>Test Data Requirement</b>		
1	web browser		1	webcam video feed		
2	webcam		2	user face in video feed		
3			3			
4			4			
<b>Test Condition s</b>						
<b>Step #</b>	<b>Step Details</b>	<b>Expected Results</b>	<b>Actual Results</b>	<b>Pass / Fail / Not executed / Suspended</b>		
1	rotate head	avatar head turns in same direction	avatar head turns in same direction	Pass		
2	smile/frown	avatar smiles/frowns	avatar smiles/frowns	Pass		
3	open/close mouth	avatar opens/closes mouth	avatar opens/closes mouth	Pass		
4	blink/wink	avatar blinks/winks	avatar blinks/winks	Pass		
5	raise/lower eyebrows	avatar raises/lowers eyebrows	avatar raises/lowers eyebrows	Pass		
6	twitch nose	avatar twitches nose	avatar twitches nose	Pass		

Test Case ID	7	Test Case Description	avatar tracks user limbs and movement						
Created By	Caelen	Reviewed By	Jose	Version			4		
QA Tester's Log									
Tester's Name	Caelen	Date Tested	November 29, 2021	Test Case (Pass/Fail/Not Executed)			Fail		
S #	Prerequisites:		S #	Test Data Requirement					
1	web browser		1	webcam video feed					
2	webcam		2	user arms in video feed					
3			3	user legs in video feed					
4			4						
Test Conditions									
Step #	Step Details	Expected Results	Actual Results			Pass / Fail / Not executed / Suspended			
1	rotate body	avatar body turns in same direction	avatar head turns in same direction			Pass			
2	wave arms	avatar waves arms	avatar waves arms			Pass			
3	move legs	avatar moves legs	avatar moves legs			Pass			
4	walk side to side	avatar walks side to side	avatar walks side to side			Pass			
5	jump up and down	avatar jumps up and down	avatar stays still			Fail			
6	walk further and closer	avatar walks further and closer	avatar stays still			Fail			



<b>Test Case ID</b>	8	<b>Test Case Description</b>	avatar tracks user fingers			
<b>Created By</b>	Caelen	<b>Reviewed By</b>	Jose	<b>Version</b>	4	
<b>QA Tester's Log</b>						
<b>Tester's Name</b>	Caelen	<b>Date Tested</b>	November 29, 2021	<b>Test Case (Pass/Fail/Not Executed)</b>	Fail	
<b>S #</b>	<b>Prerequisites:</b>		<b>S #</b>	<b>Test Data Requirement</b>		
1	web browser		1	webcam video feed		
2	webcam		2	user hands in video feed		
3			3			
4			4			
<b>Test Condition s</b>						
<b>Step #</b>	<b>Step Details</b>	<b>Expected Results</b>	<b>Actual Results</b>		<b>Pass / Fail / Not executed / Suspended</b>	
1	rotate hand	avatar hand turns in same direction	avatar hand stays still		Fail	
2	open/close fists	avatar opens/closes fists	avatar opens/closes fists		Pass	
3	make OK sign	avatar makes OK sign	avatar hand stays still		Fail	
4	make thumbs up	avatar makes thumbs up	avatar hand stays still		Fail	

## Code Review

We use camelCase for function and variable names, and kebab-case for filenames. We use 4-space indentation. We also import modules in the order of vanilla, third party, and our own code. We were very good with keeping a consistent coding style for our work because when we pull each other's work, we mimic and follow what the previous commit style is. This continued on the further that we made it into our project, and it became the standard on how we want our coding style.

### Example Code Review:

1. Caelen submits testing branch PR: login register avatar world unit tests #8
2. Vasu reviews PR
3. Vasu gives clarifying feedback on tests performed
4. Caelen merges testing branch PR to master

## Self-Check

### M1 Non-Functional Requirements:

#### A. Performance

- Loading time should be reasonable on all desktop browsers
  - DONE
- FPS should be 20+ on all desktop browsers
  - DONE
- Multiplayer should be in real-time with minimal lag
  - ISSUE - removed multiplayer from final product

#### B. Scalability

- Database should scale horizontally as more users are using the application
  - ON TRACK
- Database should scale vertically as more users join the same World
  - ISSUE - removed multiplayer from final product

#### C. Recoverability

- All database operations are logged first before they are performed to ensure recovery upon system failure
  - ON TRACK
- Users can recover accounts by resetting password
  - ON TRACK

#### D. Security

- Password encryption to secure user private information in case of security breach
  - DONE

#### E. Additional Requirements

- All 3D models and assets should be aesthetically pleasing
  - ON TRACK
- Avoid the Uncanny Valley
  - ON TRACK