

Department of Computer Science and Engineering
The Chinese University of Hong Kong

CSCI3100 Project Final Report

Hole in the Wall

Version 1.0

Print date: 06/05/2022

Group B5

Yeung Ching Fung 1155126967

LAM Tak Hing 1155131289

HO Hong Pan 1155143488

Xiao Qiang 1155143253

GOTTSCHLING, Patrick Rainer 1155173254

TABLE OF CONTENTS	1
1 INTRODUCTION	4
1.1 Project Overview	4
1.2 Objective	4
1.3 Highlights	5
1.3.1 User/ admin login and user registration	5
1.3.2 Hand detection	5
1.3.3 Anti-cheat mechanism	5
1.3.4 Scoring and learderboard	5
1.3.5 User feedbank	6
1.4 Statistics	6
1.4.1 Overall Statistics	6
1.4.1.1 Lines of Code and other	6
1.4.1.2 Overall McCabe's number	6
1.4.2 Frontend function statistics	6
1.4.2.1 Lines of Code etc	6
1.4.2.2 McCabe's number	7
1.4.3 Backend function statistics	7
1.4.3.1 Lines of Code and other	7
1.4.3.2 McCabe's number	7
2 SYSTEM ARCHITECTURAL DESIGN by DFD	8
2.1 System Architecture	8
2.2 DFDs	9
2.2.1 DFD for normal user (player)	9
2.2.2 DFD for admin	9
3 DETAILED DESCRIPTION OF COMPONENTS by UML	10
3.1 Login and Registration	10
3.1.1 Structural Diagram	10

3.1.2 UML Diagrams	10
3.1.2.1 Class Diagram	10
3.1.2.2 Sequence Diagram	11
3.1.3 Functionality	12
3.1.4 Procedures and Functions	12
3.2 Game	13
3.2.1 Structural Diagram	13
3.2.2 UML Diagrams	13
3.2.2.1 Class Diagram	13
3.2.2.2 Sequence Diagram	14
3.2.3 Functionality	15
3.2.4 Procedures and Functions	15
3.3 Wall	15
3.3.1 Structural Diagram	16
3.3.2 UML Diagrams	16
3.3.2.1 Class Diagram	16
3.3.2.2 Sequence Diagram	17
3.3.3 Functionality	18
3.3.4 Procedures and Functions	18
3.4 Timer	18
3.4.1 Structural Diagram	19
3.4.2 UML Diagrams	19
3.4.2.1 Class Diagram	19
3.4.2.2 Sequence Diagram	20
3.4.3 Functionality	20
3.4.4 Procedures and Functions	20
4 USER INTERFACE DESIGN	21
4.1 Login And Registration Page	21

4.2 Home page	21
4.3 Game page	24
4.4 Admin page	27
5 TEST	29
5.1 Test Overview and test plan	29
5.2 Test case assumptions	29
5.3 Result	29
6 Lessons Learned	30
7 Conclusion	30
8 Appendix	30
8.1 Database specification	31
8.1.1 All Database Functions	31
8.1.2 Database Functions in UML Class diagram	32
8.1.3 Database Functions in UML Use Case diagram	33
8.1.4 Database Functions in UML Sequence diagram	34
8.2 Test cases	35
8.2.1 Login and registration page	35
8.2.1.1 Login	35
8.2.1.2 Registration	35
8.2.2 Home page	37
8.2.2.1 Navigation bar	37
8.2.2.2 Starting page	38
8.2.2.3 Your Info	38
8.2.2.4 Modify your Info	38
8.2.2.5 Reset Password	40
8.2.2.6 Feedbacks	41
8.2.3 Game page	41
8.2.4 Admin page	42

8.2.4.1 Navigation bar	42
8.2.4.2 Reset User Password	42
8.2.4.3 Delete user's gameplay	43
8.2.4.3 Delete user's account	43
8.2.5 Page routings	44
8.2.6 Unexpected action	44
8.2 Images used as walls in the game	45

1 INTRODUCTION

1.1 Project Overview

Based on the team members' experiences in utilising Machine Learning models, Web Development, Game Design and Database Management, our group decided to implement a web-based game that utilizes a hand recognition models. The game resembles "Hole in the Wall" (Brain Wall/ Human Tetris), a physical game show contest in which contestants contort their bodies to fit through cut-outs holes in a large styrofoam wall moving toward them.

In our game design, users can only use their hands to play the game. Hands are treated as the contestants, and a virtual game map that represents the styrofoam wall will be revealed in the game interface. Users need to place their hands properly with the suppositional hand gesture, or use their creativity to find a posture that could fit into the hole. Users would be able to compete by the number of stages they have cleared in a limited time frame.

1.2 Objective

The goal of the software is to convert the physical game concept in a digital setting with a state-of-the-art hand recognition model. The development process should incorporate a comparable game design to avoid building a technical intensive but boring product. By achieving compelling game level designs and competitive features, it will not only be a refreshing and exciting game for first-time users, but also a challenging game contest that could attain active players.

The game system contains several technical challenges as it incorporates video processing and complex recognition problems. We aim to minimize the computational complexity and conduct thorough exception handlings to provide a smooth and reliable user experience. The product should be highly accessible like Slither.io, subjected to access of a static camera position for hands tracking.

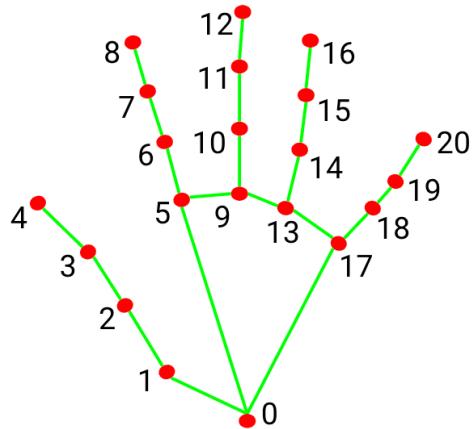
1.3 Highlights

1.3.1 User/ admin login and user registration

In addition to basic features for web applications like user registration, verification and a User and Admin Interface, users are also able to modify their email or username, reset their password or change their icon.

1.3.2 Hand detection

Since the game design is solely based on recognizing the user's hand movements, the client device needs to be equipped with a camera module. A pretrained hand recognition model will access the video to extract the user's hand-palm-fingers locations (21 hand landmarks in coordinates format). The system could then perform a series of computations to decided whether the hand location has satisfied the game stage requirements — the hand should avoid the wall region, but touches all highlighted regions on the fingertips. The former rule is identical to the origin Hole in the Wall game; while the latter rule is newly added to prevent users from passing the majority of stages with hand gestures that have a small projected area e.g. making a fist or placing their hand flat and sideways.



- | | |
|-----------------------|-----------------------|
| 0. WRIST | 11. MIDDLE_FINGER_DIP |
| 1. THUMB_CMC | 12. MIDDLE_FINGER_TIP |
| 2. THUMB_MCP | 13. RING_FINGER_MCP |
| 3. THUMB_IP | 14. RING_FINGER_PIP |
| 4. THUMB_TIP | 15. RING_FINGER_DIP |
| 5. INDEX_FINGER_MCP | 16. RING_FINGER_TIP |
| 6. INDEX_FINGER_PIP | 17. PINKY_MCP |
| 7. INDEX_FINGER_DIP | 18. PINKY_PIP |
| 8. INDEX_FINGER_TIP | 19. PINKY_DIP |
| 9. MIDDLE_FINGER_MCP | 20. PINKY_TIP |
| 10. MIDDLE_FINGER_PIP | |

(the 21 hand landmarks output from the mediapipe hand detection model, image from <https://google.github.io/mediapipe/solutions/hands>)

1.3.3 Anti-cheat mechanism

Moreover, we have also designed a distance detection mechanism to prevent the user from cheating. Since placing hands far away from the camera would result in an even smaller projected area, users could easily avoid the wall regions by doing so. We have came up with a simplistic yet effective algorithm that calculates the distance between the hand landmarks. Once the distance become too close, it implies that the user's hand is too far away from the camera. The game stage could not be passed during this state and the user is prompted to move their hand closer.

1.3.4 Scoring and learderboard

The user is provided with 15 seconds to complete as much stages as possible, while passing each stage contributes 1 score. Users could compete on the learderboard that shows their own score and the highest scores achieved by other users.

1.3.5 User feedbank

Besides the usage of a Machine Learning model in our game, we worked on a feedback system to constantly improve the user game experience. On the feedback page the latest feedback from other users are displayed and new feedback can be written and saved in the database.

1.4 Statistics

1.4.1 Overall Statistics

1.4.1.1 Lines of Code and other

		Lines of Code	Bugs	Vulnerabilities	Code Smells	Security Hotspots	Coverage	Duplications
cscli3100project_2021_22								
models	Backend functions and database schema	466	0	7	21	3	—	0.0%
pages	Static HTML pages	1,167	12	0	2	8	—	1.9%
public	Javascript for dynamic frontend functions and CSS	790	1	0	42	1	—	0.0%
app.js	routing functions	233	0	0	15	3	—	0.0%

1.4.1.2 Overall McCabe's number

Cyclomatic Complexity	268	
models	73	
pages	4	
public	147	
app.js	44	

1.4.2 Frontend function statistics

1.4.2.1 Lines of Code etc

		Lines of Code	Bugs	Vulnerabilities	Code Smells	Security Hotspots	Coverage	Duplications
js		453	1	0	23	1	—	0.0%
game.js		112	0	0	4	0	—	0.0%
game_elements.js		14	0	0	0	0	—	0.0%
lboard.js		91	0	0	6	0	—	0.0%
rule.js		15	0	0	0	0	—	0.0%
sweetalert.js		29	1	0	2	0	—	0.0%
timer.js		51	0	0	0	0	—	0.0%
wall.js		141	0	0	11	1	—	0.0%

1.4.2.2 McCabe's number

Cyclomatic Complexity 92

game.js	20
game_elements.js	0
lboard.js	14
rule.js	2
sweetalert.js	6
timer.js	15
wall.js	35

1.4.3 Backend function statistics

1.4.3.1 Lines of Code and other

	Lines of Code	Bugs	Vulnerabilities	Code Smells	Security Hotspots	Coverage	Duplications
models	466	0	7	21	3	—	0.0%
Admin_functions.js	98	0	0	4	1	—	0.0%
cookiecheck.js	38	0	0	1	0	—	0.0%
feedback.js	9	0	0	0	0	—	0.0%
leader_board.js	7	0	0	0	0	—	0.0%
mail.js	63	0	0	1	2	—	0.0%
System_functions.js	75	0	1	6	0	—	0.0%
testdb.js	18	0	0	3	0	—	0.0%
User.js	10	0	0	0	0	—	0.0%
User_functions.js	148	0	6	6	0	—	0.0%

1.4.3.2 McCabe's number

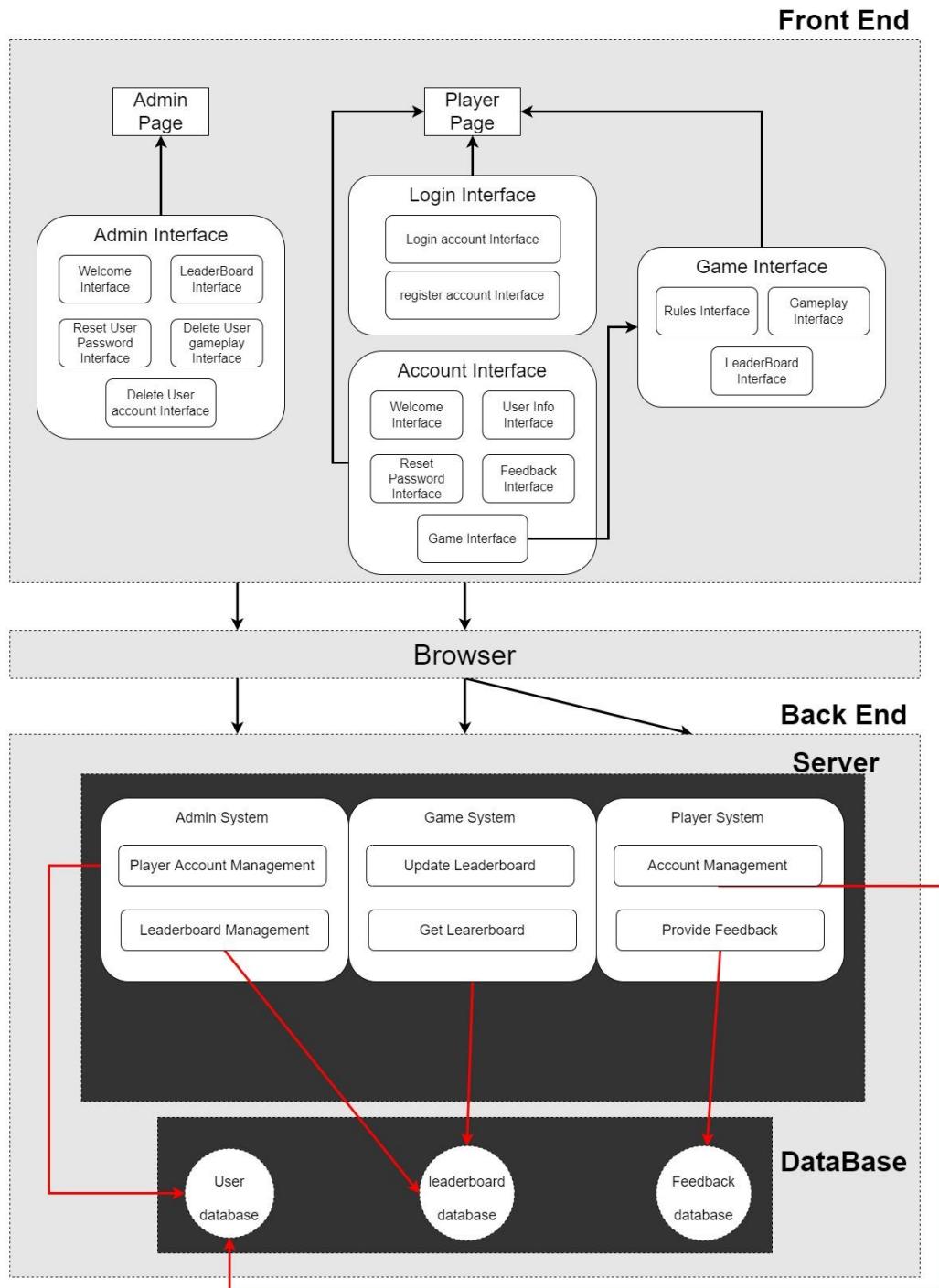
Cyclomatic Complexity 79

Admin_functions.js	17
cookiecheck.js	10
feedback.js	0
leader_board.js	0
mail.js	8
System_functions.js	11
testdb.js	6
User.js	0
User_functions.js	27

(Statistics calculated using SonarCloud.)

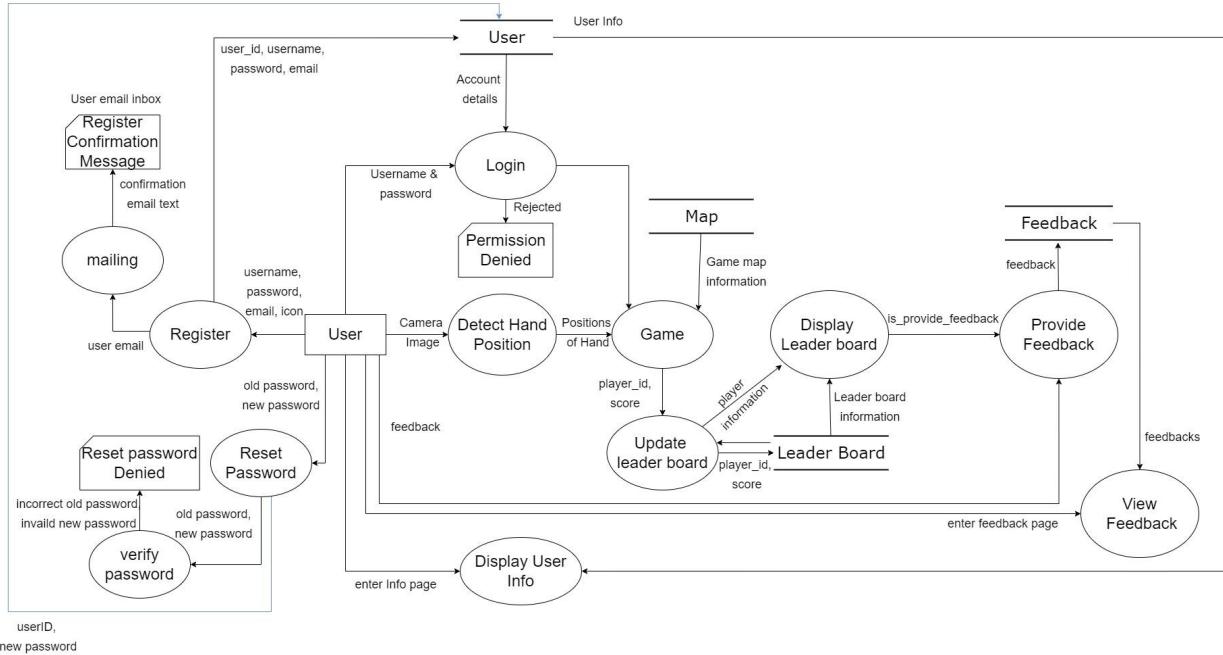
2 SYSTEM ARCHITECTURAL DESIGN by DFD

2.1 System Architecture

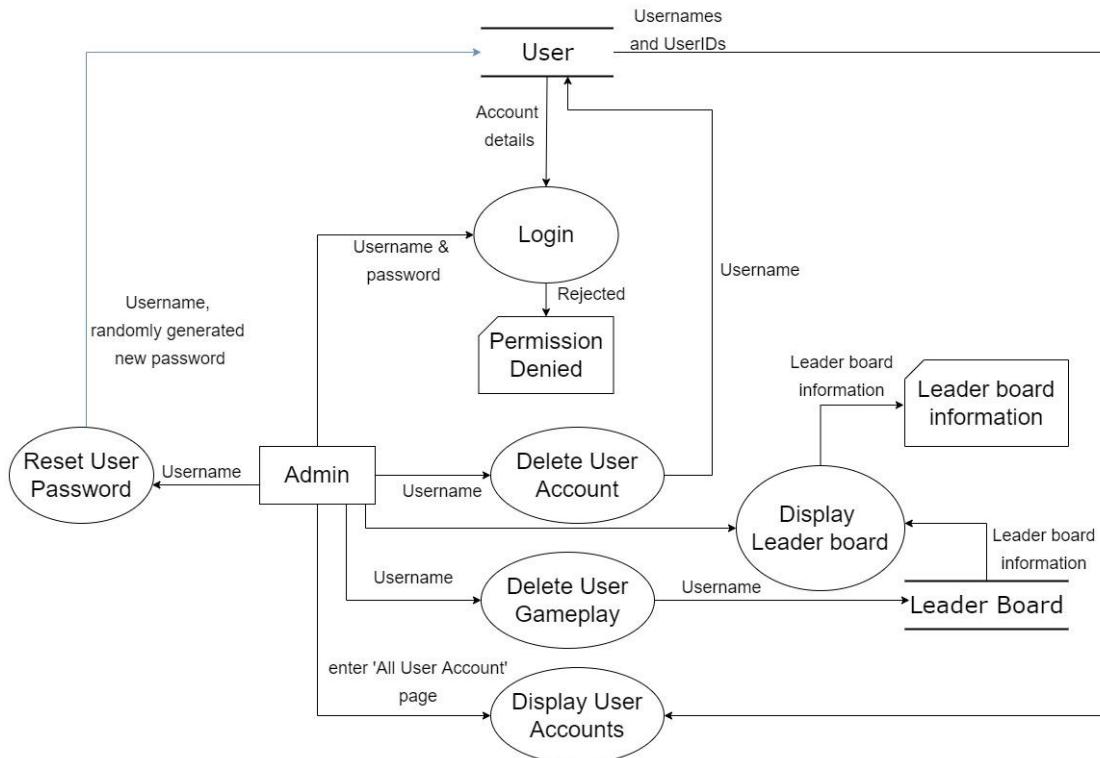


2.2 DFDs

2.2.1 DFD for normal user (player)



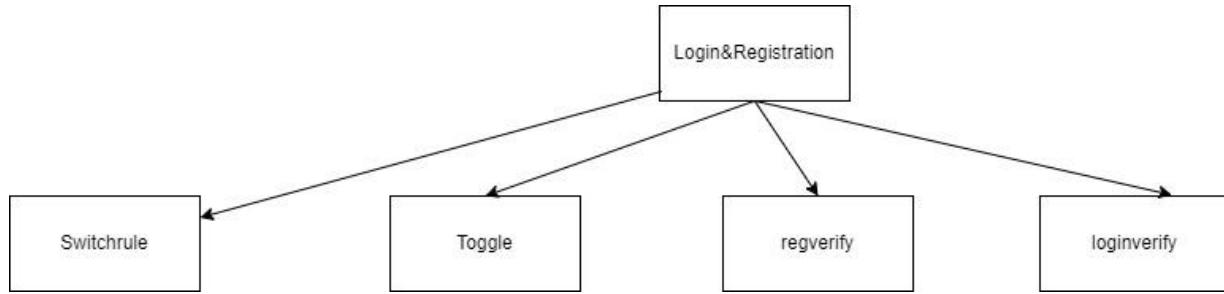
2.2.2 DFD for admin



3 DETAILED DESCRIPTION OF COMPONENTS by UML

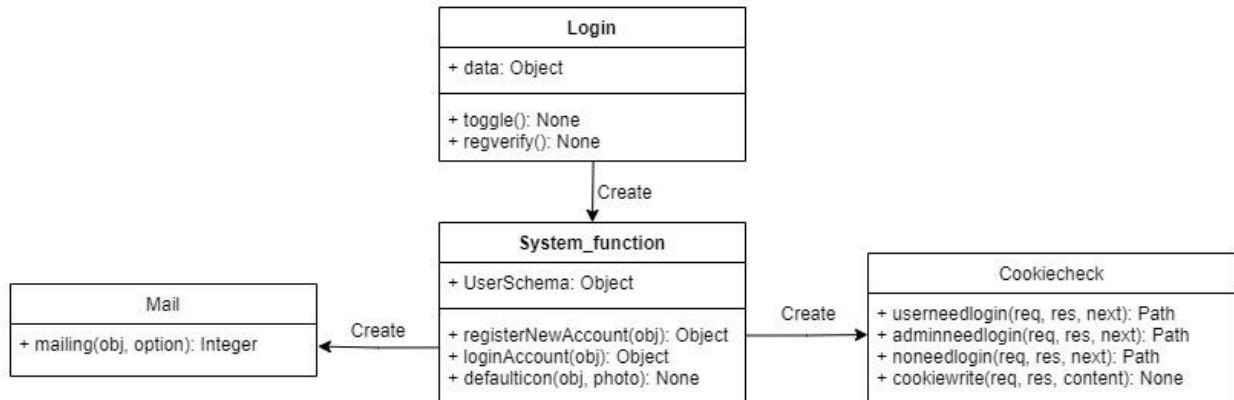
3.1 Login and Registration

3.1.1 Structural Diagram

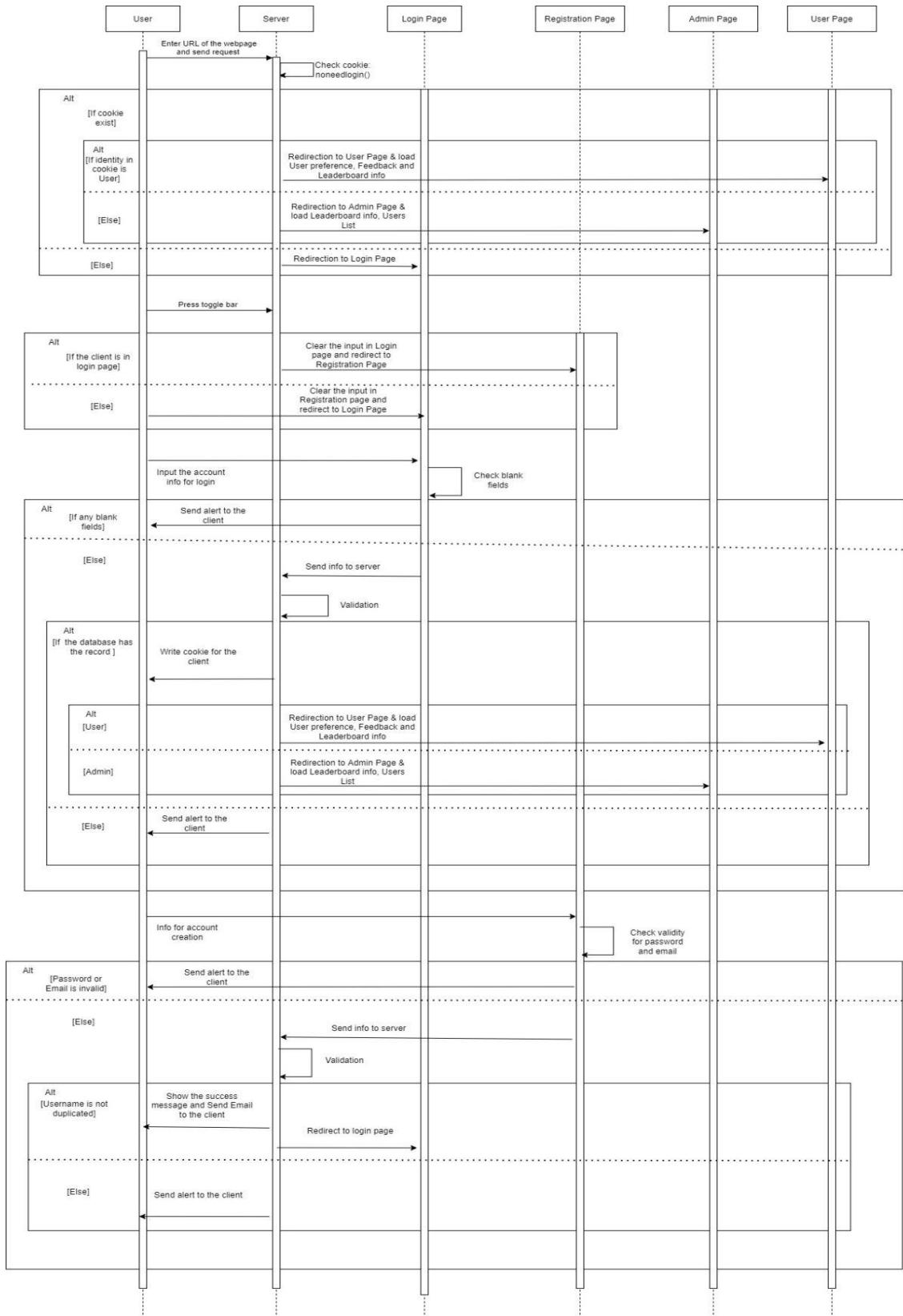


3.1.2 UML Diagrams

3.1.2.1 Class Diagram



3.1.2.2 Sequence Diagram



3.1.3 Functionality

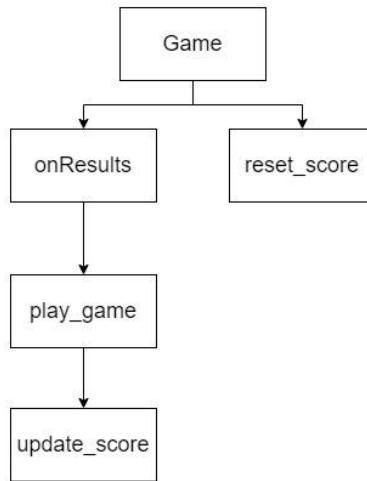
This function is the entry point of the system i.e. handles account creation and user login.

3.1.4 Procedures and Functions

Function	Description
registerNewAccount(obj)	Create a new account.
loginAccount(obj)	Try to login to an account.
defaulticon(obj, photo)	Set the icon of the newly created account to the default icon.
mailing(obj, option)	Send a confirmation email to the email address that the user inputted.
userneedlogin(req, res, next)	Redirect the user to the login and registration page if the user tries to go to the home page without logging in.
adminneedlogin(req, res, next)	Redirect the user to the login and registration page if the user tries to go to the admin page without logging in.
noneedlogin(req, res, next)	Redirect the user to the admin page or the home page if the user has already logged in.
cookiewrite(req, res, content)	Write the username and if the user is admin into the cookie.
toggle()	Direct the data to log in or register an account.
regverify()	Check if the data is in the correct form.

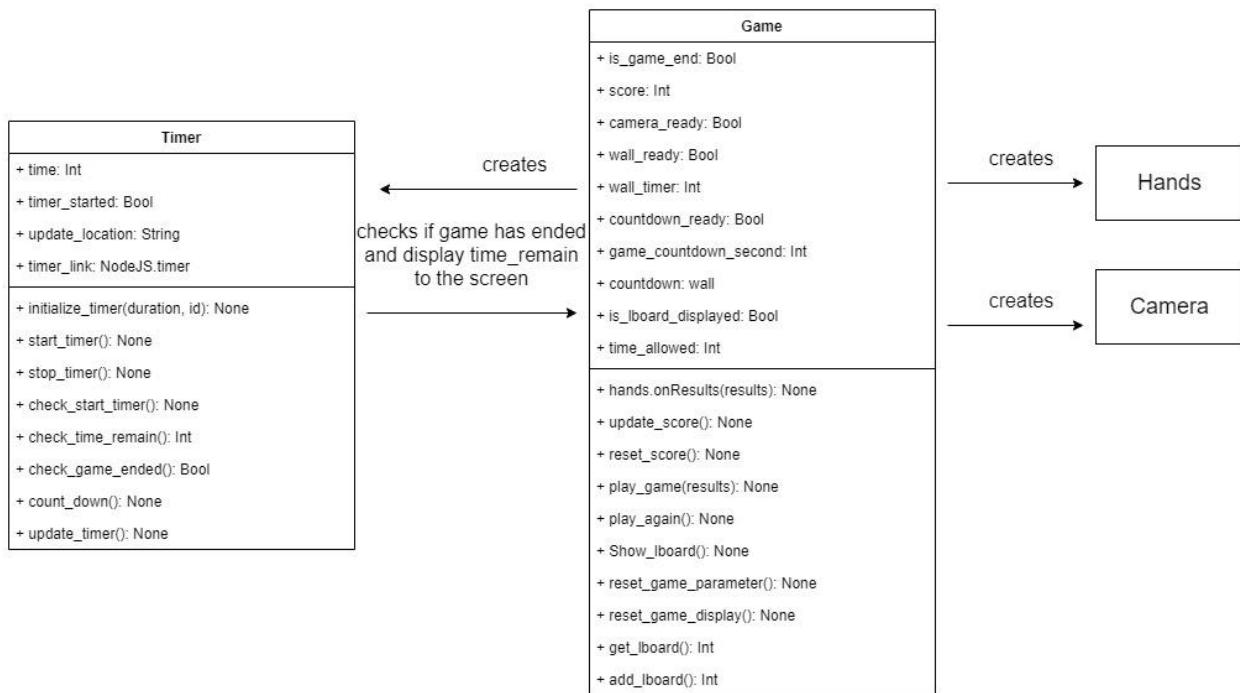
3.2 Game

3.2.1 Structural Diagram

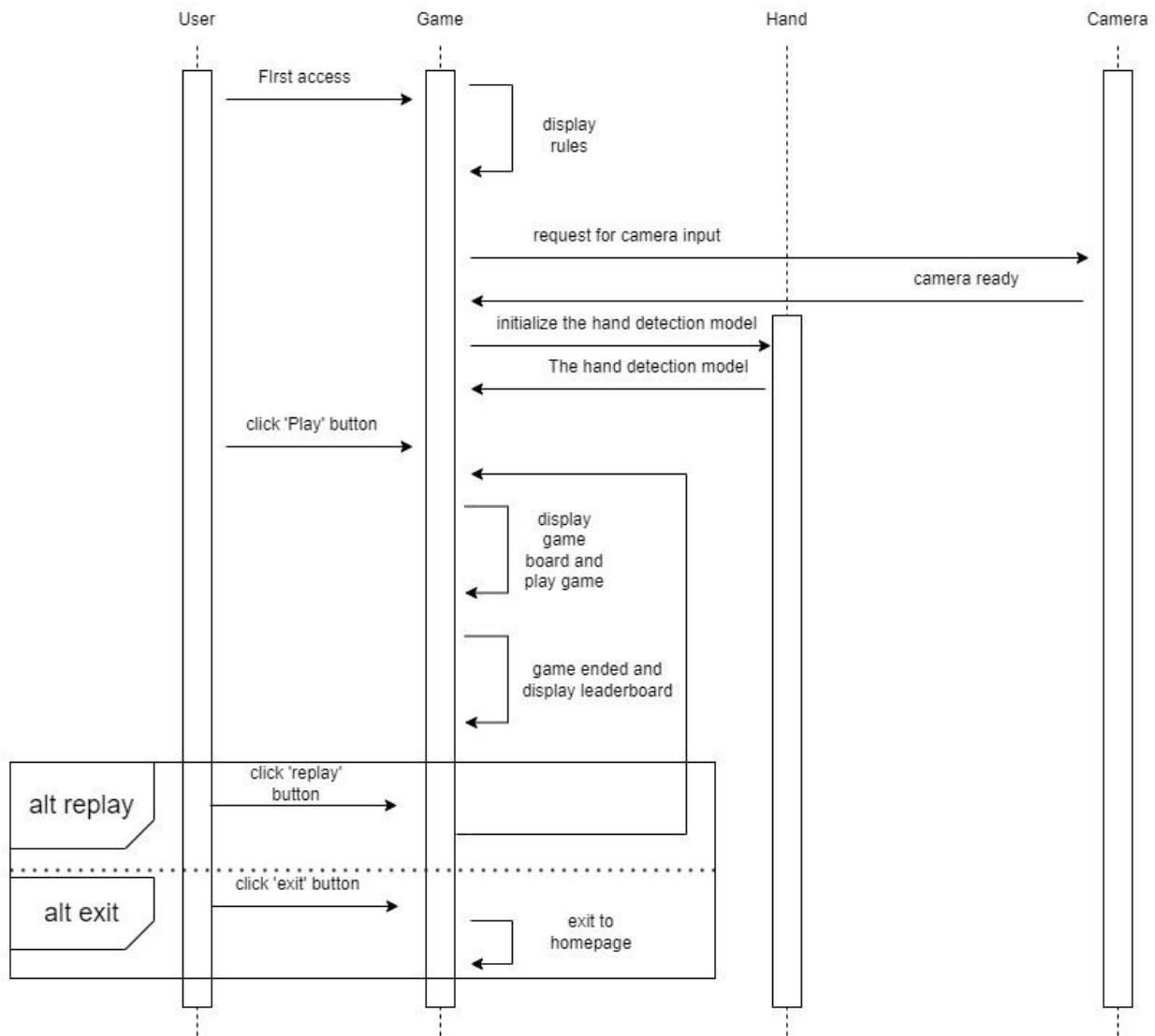


3.2.2 UML Diagrams

3.2.2.1 Class Diagram



3.2.2.2 Sequence Diagram



3.2.3 Functionality

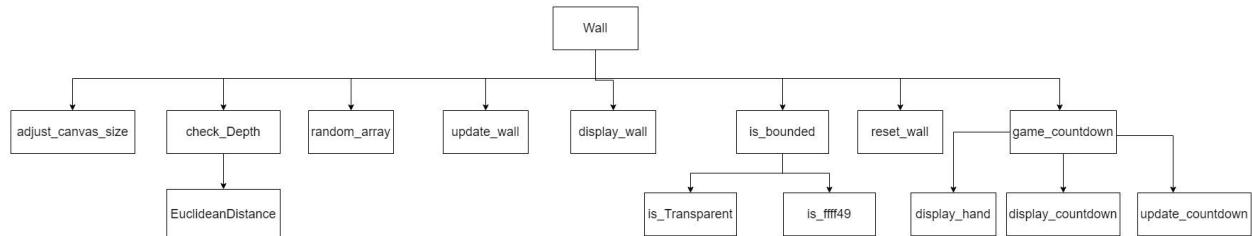
This component handles all game logic determination, including game display, scoring and game end conditions.

3.2.4 Procedures and Functions

Functions	Description
onResults()	a function that handles each frame captured by the camera, which is also where the main game logic occurs
play_game()	a function that handles the game logic, includes: <ol style="list-style-type: none">1. game state check (is_game_started, is_game_ended)2. score check3. update countdown and wall images
update_score()	a function that updates the score after player passed a wall
reset_score()	a function that resets the score to 0, called when the player replay the game.
play_again()	This function is called after the replay button is clicked, which will reset all game settings and restart the game.
Show_lboard()	This function is called when the game has ended (15 seconds passed), which will then display the leaderboard.
reset_game_parameter()	this function resets all game parameters when called.
reset_game_display()	This function resets the display of the game, i.e. displays the game canvas again.
get_lboard()	This function adds the obtained leaderboard information to the leaderboard display.
add_lboard()	This function fetches and returns the leaderboard information from the database.

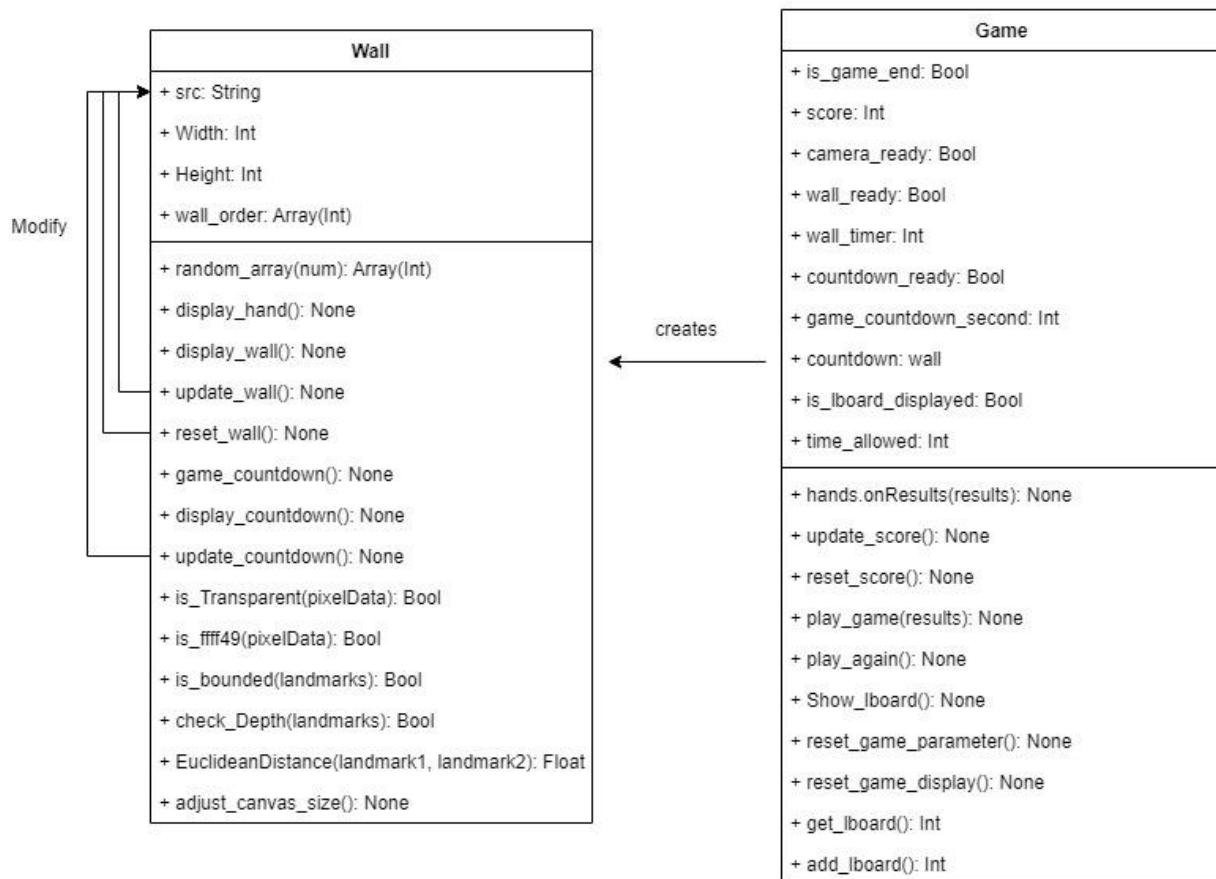
3.3 Wall

3.3.1 Structural Diagram

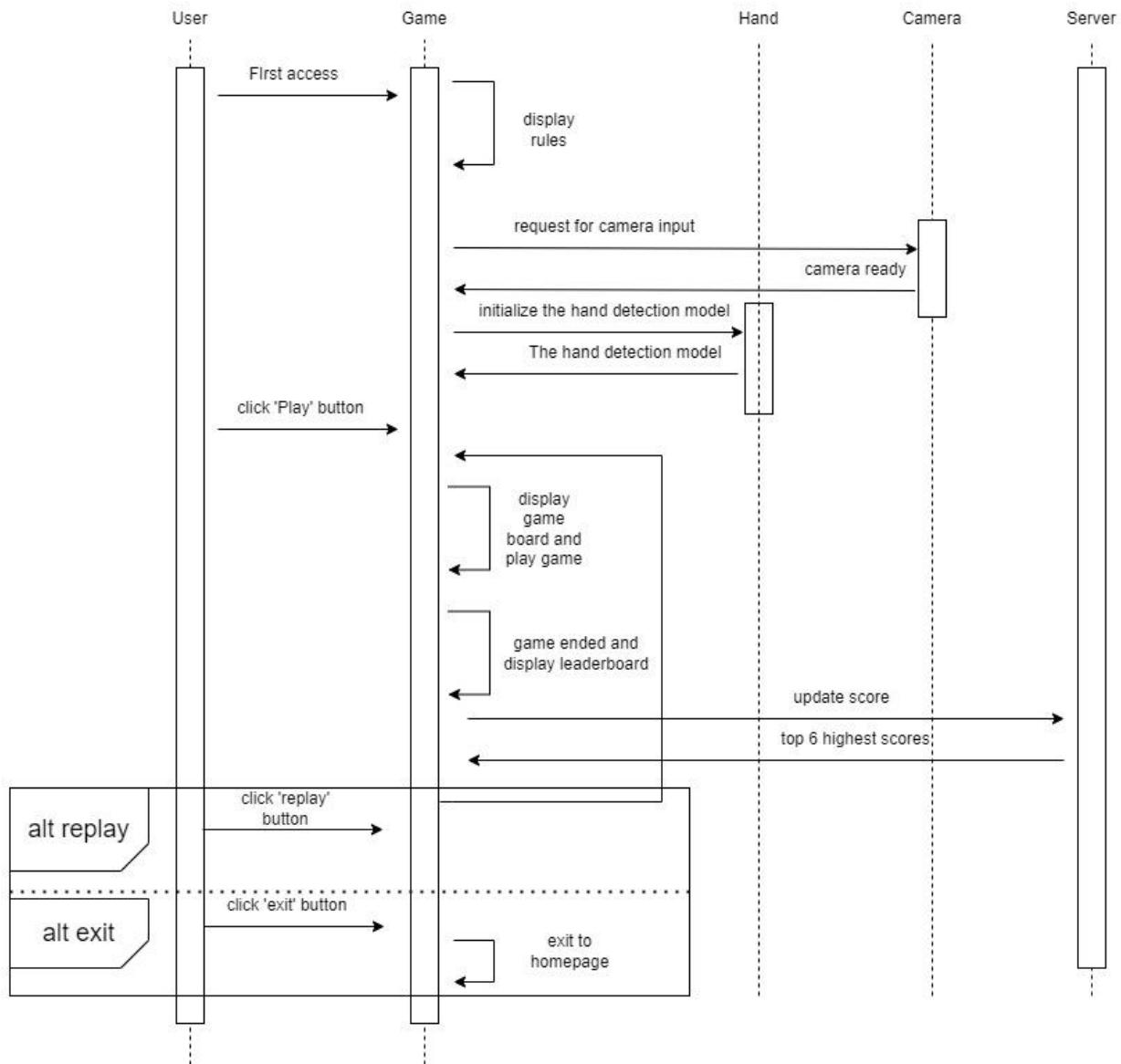


3.3.2 UML Diagrams

3.3.2.1 Class Diagram



3.3.2.2 Sequence Diagram



3.3.3 Functionality

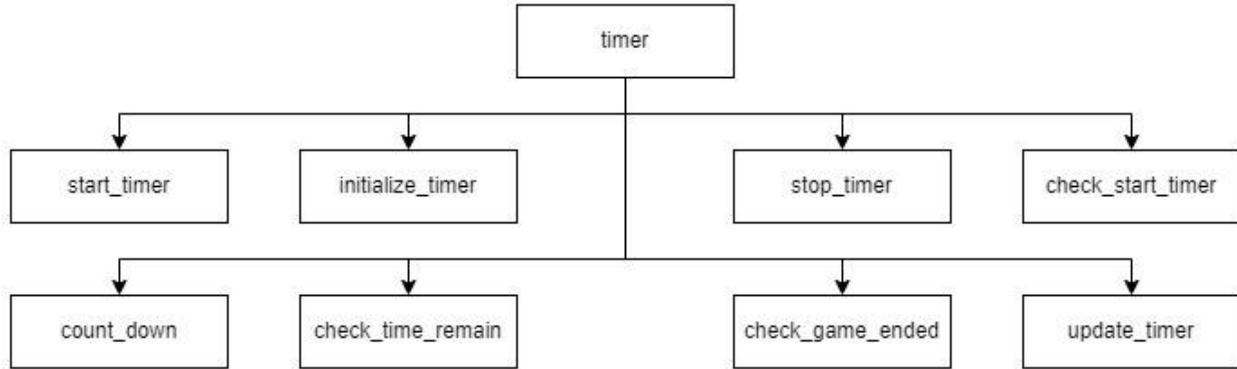
This component handles all actions related to the image displayed in the game canvas.

3.3.4 Procedures and Functions

Functions	Description
random_array(num)	returns a random shuffled array of wall indices.
display_hand()	draws the detected hand onto the game canvas
display_wall()	draws the wall image onto the game canvas
update_wall()	update the wall image, called after player successfully passed a wall
reset_wall()	reset the wall image and the wall index array, called after player clicked replay
game_countdown()	Countdown and display the countdown images when the game has just started
display_countdown()	Display the countdown image onto the game canvas when the game is counting down before start
update_countdown()	updates the countdown image during countdown called once a second update game_countdown_second == 0
is_Transparent(pixelData)	check if the input pixel on the wall image is transparent
is_ffff49(pixelData)	check if the input pixel on the wall image is of RGB value #ffff49
is_bounded(landmarks)	check if the input hand landmarks are all bounded by the wall image
check_Depth(landmarks)	check if the player's hand is too far away from the camera. (as an anti-cheat measure)
EuclideanDistance(landmark1, landmark2)	computes the 3D euclidean distance between 2 hand landmarks
adjust_canvas_size()	adjusts the wall canvas size dynamically

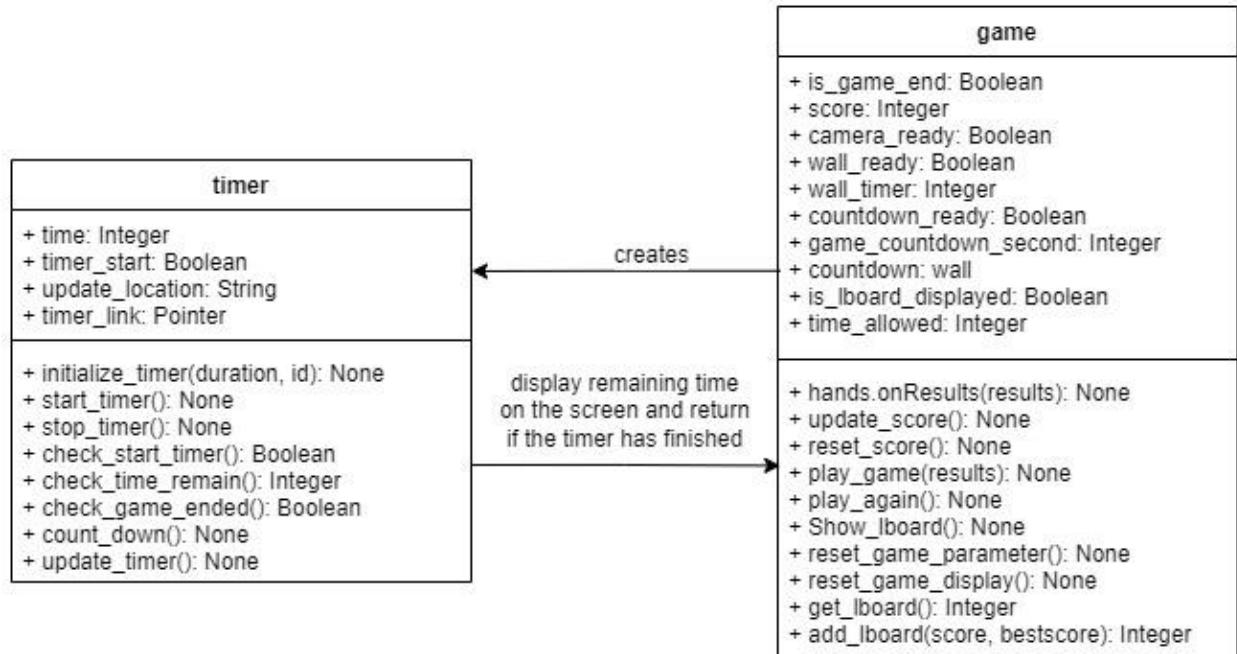
3.4 Timer

3.4.1 Structural Diagram

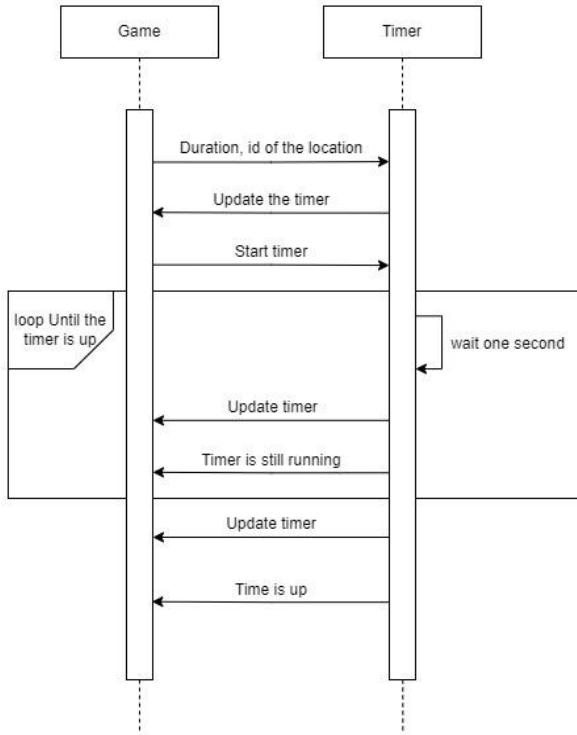


3.4.2 UML Diagrams

3.4.2.1 Class Diagram



3.3.2.2 Sequence Diagram



3.4.3 Functionality

This component handles the countdown of the timer of the initialized duration and displays the remaining time in the initialized location. It can also stop the timer, return the remaining time, and the state of the timer.

3.4.4 Procedures and Functions

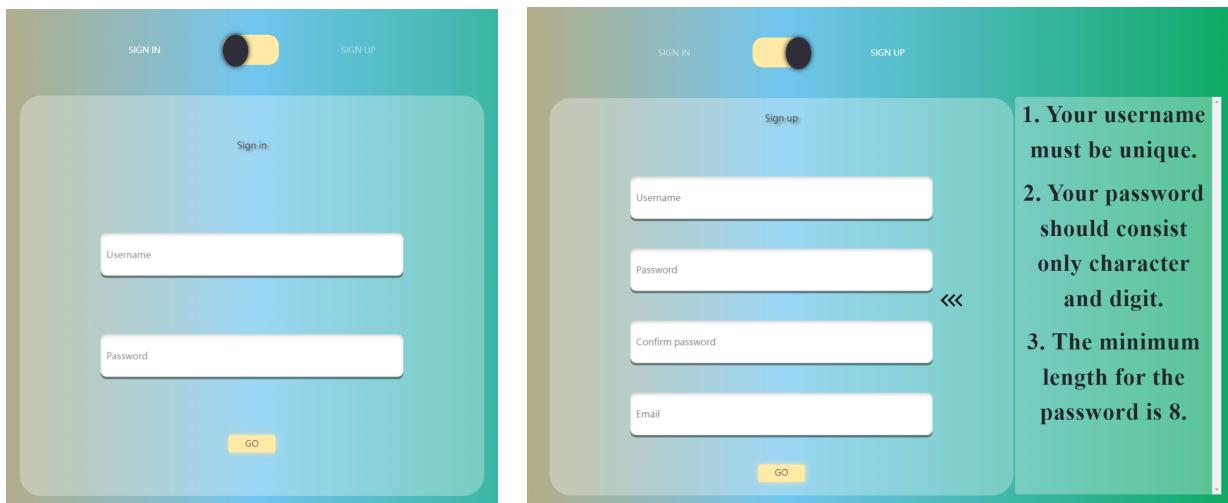
Function	Description
initialize_timer(duration, id)	Initialize the timer with the duration and the location of the timer.
start_timer()	Start the countdown of the timer.
stop_timer()	Stop the countdown of the timer.
check_start_timer()	Return if the timer has started or not.
check_time_remain()	Return the remaining time.
check_game_ended()	Return if the countdown has ended or not.
count_down()	Countdowning the timer.
update_timer()	Update the timer in the location.

4 USER INTERFACE DESIGN

The UI components are implemented mainly with plain CSS, and Bootstrap to allow rapid development. Although certain design references were made from CodePen and template websites, all implementation processes were original and completely built from scratch.

4.1 Login And Registration Page

Once an anonymous user (with no identity session on the server) enters the website, the sign-in page will be displayed.



For the sign-in page, a background picture will be loaded first. Then, a toggle bar and a gray form with two shaded input fields for the username and password, which will be floating on the background picture.

When there is no input, the placeholder in each field will notify the user what is required for the field. If the user tries to press the 'go' button with any blank fields on the page, an alert 'Please fill this field' will pop out. If all fields are filled and button is pressed, the server will check the identity. If the account information input is wrong, the input will be eliminated. Then, the placeholder will be changed into 'the username or password is wrong!' And the border of the input field will turn into red for notification. Once the user input the correct account information, he will be redirected to the user home page.

'Sign In' text is on the left side of the toggle bar and the 'Sign Up' text is on the right side. The text will be highlighted to indicate which form the user is filling in. By pressing the ball in the toggle bar, the user can switch between the signin and signup function. Animation is added in the transition. There will be 0.7s for the ball to move from one side to another side and the highlighted text will be changed. Concurrently, the form will smoothly flip along the y-axis and disappear to provide another form to the user.

If user switch to the signup form, the specification for the creating username and password will be displayed on the board at the right of the form. There are small black arrows for showing or hiding the specification. Similarly, the placeholders will be shown on the input fields to notify the required input for the field. If any blank field is found after pressing the 'go' button on the form, the border of the input field will turn red and the placeholder will change into 'Can't be empty!!!'

Also, input checking will proceed. Once an invalid input is found, an alert box with the message corresponding to the error input type will pop out. Then, the input on the field will be removed and the

border for that field will turn red. After the user updates the invalid input and presses the button , the 'red' effect will disappear. If all inputs fulfill the requirements, the success alert will pop out and the user will be redirected to the signin page.

4.2 Home page

After users have successfully logged in, they will be directed to the home page. The home page contains a navigation bar to access different sections of the website. The main page shows a short introduction to our services with a quick start button and news feed for the newly added features.

Wall Hole

Your Info Modify your Info Reset Password Feedbacks Start Game Log-out *

Welcome user to
Hole in the Wall

Start Game

What's new

- The game is now online!
- User can now leave their feedback of the gema at the feedback page.
- User can now change the icon.

If the user wants to view the profile of their account, they can click on the button "Your info" at the navigation bar at the top of the home page. After the user clicks on the button, the user's icon, username, and email will be shown with the buttons "Update info" and "Reset password" under the email.

Wall Hole

Your Info Modify your Info Reset Password Feedbacks Start Game Log-out *

Your Info

Your Information

Username: JapaneseHentai
Email: p80246@gmail.com

Update Info Reset Password

If the user wants to modify the profile of their account, they can click on the button "Modify your info" at the navigation bar at the top of the home page or the button "Update info" under "Your Info". After the user clicks on the button, there will be two input boxes for the user to input their new username and email address, a button for the user to choose their new icon, and an "Update" button at the end of the form for the user to upload their new profile. After the user has clicked the "Update" button, multiple checks will be performed. First, the form will check if at least one of the fields of the form has been filled. Then, the form will check if the newly inputted username and the email address are different from the one that the user has. Finally, the form will check if the email address is in the correct form. If all the checks for the

inputs have passed, the updated profile will be sent to the server and update the user profile. If the check fails, a warning message will be displayed to the user and the upload will be stopped.

The screenshot shows a user interface for modifying account information. At the top, there is a navigation bar with links: Your Info, Modify your Info, Reset Password, Feedbacks, Start Game (highlighted in orange), and Log-out. Below the navigation bar is a section titled "Modify your Information". It contains three input fields: "Username" (empty), "Email address" (empty), and "New Icon". Under "New Icon", there is a button labeled "選擇檔案" (Select File) and a note "未選擇任何檔案" (No file selected). At the bottom of the form is a blue "Update" button.

If the user wants to modify the password of their account, they can click on the button "Reset Password" at the navigation bar at the top of the home page or the button "Reset Password" under "Your Info". After the user clicks on the button, there will be three input boxes for the user to input their old password, new password, and the new password again, and an "Update" button at the end of the form for the user to upload their new password. After the user has clicked the "Update" button, multiple checks will be performed. First, the form will check if all input boxes have been filled. Next, the form will check if the inputted old password is the same as the one that the user has. Then, the form will check if the new password is different from the old password. After that, the form will check if the two new passwords are the same. Finally, the form will check if the new password is in the correct form. If all the checks for the inputs have passed, the new password will be sent to the server, update the user password, and the user will need to log in to their account again. If the check fails, a wrong message will be displayed to the user and the upload will be stopped.

The screenshot shows a user interface for resetting a password. At the top, there is a navigation bar with links: Your Info, Modify your Info, Reset Password, Feedbacks, Start Game (highlighted in orange), and Log-out. Below the navigation bar is a section titled "Reset Password". It contains three input fields: "Old password" (empty), "New password" (empty), and "New password again" (empty). Below the "New password" field is a note: "Password should consist only character and digit with minimum length of 8.". Below the "New password again" field is a note: "Must be the same as above". At the bottom of the form is a blue "Update" button.

If the user wants to view the feedback from other users, they can click on the button "Feedbacks" at the navigation bar at the top of the home page. After the user clicks on the button, there will be an input box for the user to input their feedback and a "Submit" button after the input box. Then, there will be a list of feedback from users with their username and icon.

Feedbacks

Leave your own feedback:

(C)

Submit

Username: JapaneseHentai
 Feedback: Only 1% of people can score higher than 6 in this insane game!!!

Username: Patrick
 Feedback: So much fun! Cannot imagine it is for free!

If the user wants to leave their own feedback, they can input their feedback at the input box at the top of the page under "Feedbacks". After the user has inputted their feedback, they can click on the submit button. If the input box is filled, the feedback will be sent to the server and the user feedback will be displayed at the top of the list of feedback, else a wronging message will pop up and alert the user that the input box is not filled.

If the user wants to play the game, they can click on the button "Start Game" at the navigation bar at the top of the home page or the button "Start Game" under the short introduction on the first page. After the user clicks on the button, the website will bring the user to the game page.

If the user wants to go back to the first page, they can click on the button "Wall Hole" at the navigation bar at the top of the home page. After the user clicks on the button, the website will bring the user to the first page.

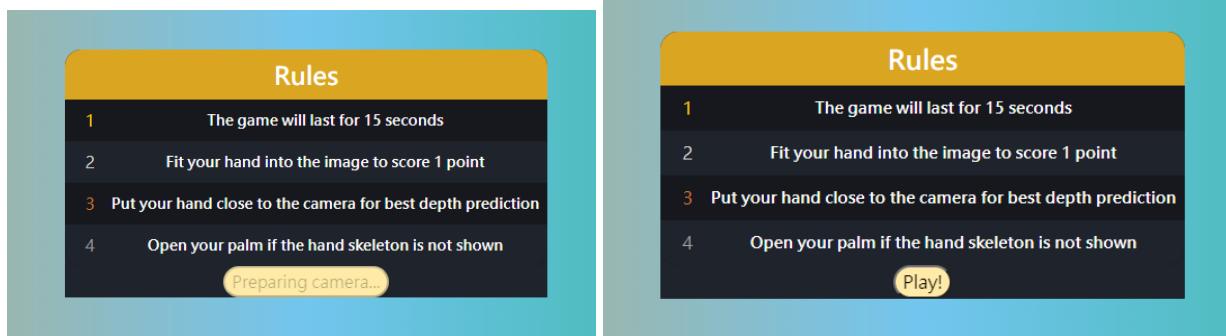
If the user wants to log out of the account, they can click on the button "Log-out" at the navigation bar at the top of the home page. After the user clicks on the button, the website will log out of the user account, save the last page that the user is viewing in order to display the last page after the user logs in next time and bring the user back to the login and registration page.

If the user wants to change the theme of the page, they can click on the last button at the navigation bar at the top of the home page. After the user clicks on the button, the website will save the preference of the theme and display the same after they log in the next time.



4.3 Game page

The Game page consists of 3 separate components, which are displayed at different stages of the game. Initially, when the user first enters the page, rules of the game will be displayed, and the program will start requesting camera access.

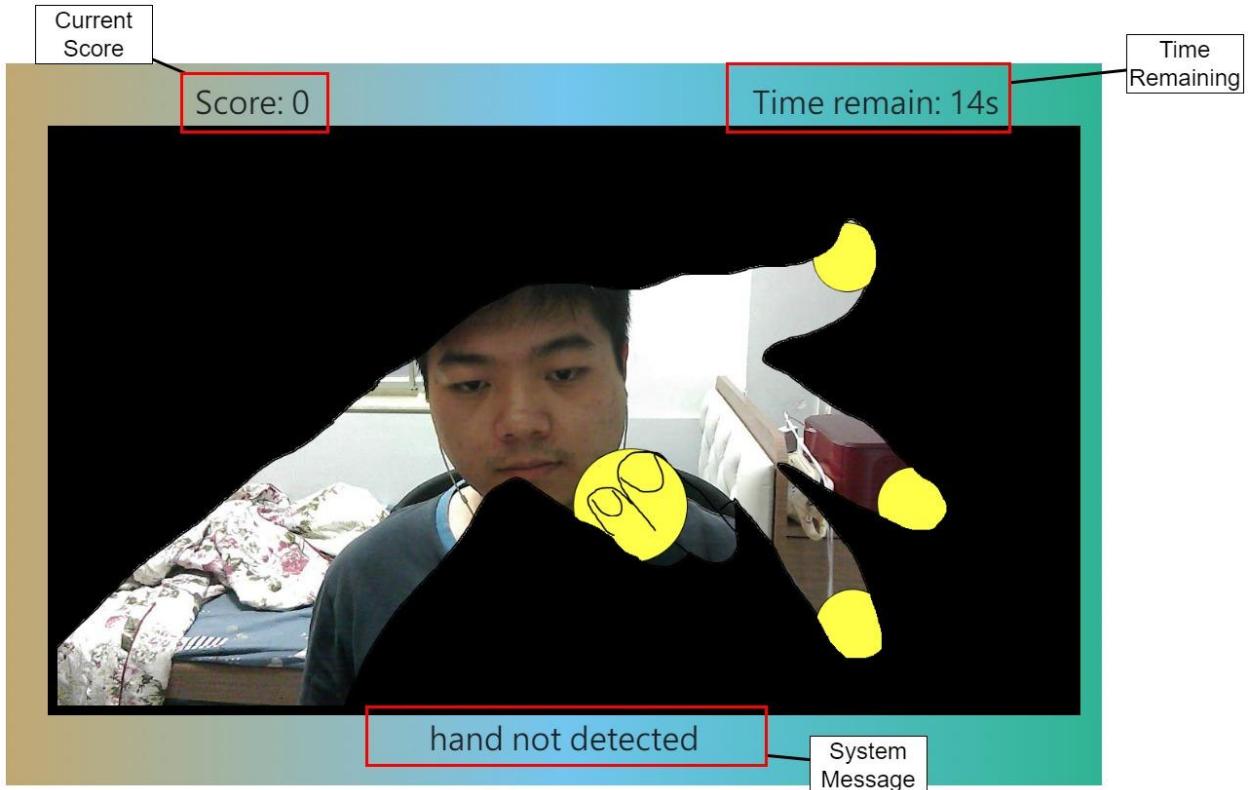


Once the camera input is ready, the 'play' button will be set to available and the user can click the button to start the game.

Upon game start, the timer will start the countdown and the user can see the game canvas located at the center of the screen, displaying the camera input, the detected hand skeletons and the wall. On top of the game canvas are the game information (score, time_remaining), and below the game canvas is the information of the status of the detected hand.



If the calculated depth of the hand is too far away from the camera, a message 'your hand is too far away' will be displayed, and if no hands are detected, a message 'no hands detected' will be displayed.'



After the countdown is over, the game is ended and the leaderboard page will be displayed, in the leaderboard page, top 6 scores in the leaderboard will be displayed alongside with the user's score from the game play. There are also 2 buttons, a 'replay' button and an 'exit' button. Upon clicking the replay button, the game component will be displayed again and the game will start over. If the 'exit' button is pressed, the user will exit the game and be redirected to the home page.

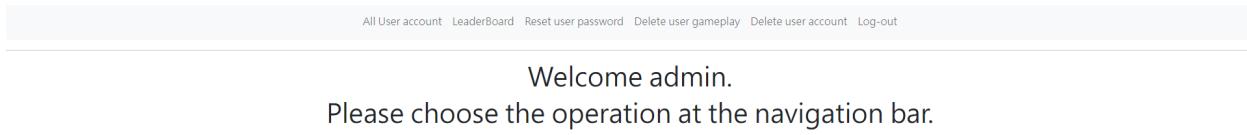
Leader Board		
1	Peter	4
2	gamedemo	3
3	Brian	3
4	Jacky	3
5	JapaneseHentai	3
6	Thompson	2

Your final score: 0

(Play Again) (Exit)

4.4 Admin page

After the admin has successfully logged in, they will reach the admin page. The admin page will contain a navigation bar to reach the different pages of the website, a short welcome of the admin.



If the admin wants to view the username of all accounts, they can click on the button "All User account" at the navigation bar at the top of the admin page. After the admin clicks on the button, the username and user ID of all the accounts will be displayed in a table.

Username	User ID
brian1234	62350cc97293107aa7d1ed02e
1234	6255014a8934d092a0112a
jacky	a/5b/8071/8813/e13/a1/999
gamedemo	625629f110101f1f6a/8a7
JapaneseHentai	6256280164a1104118a0000
Thompson	62562a3a030d81f716639bd
Brian	62562a50390j81171/6630a
Jacky	a/5b/8071/8813/e13/a1/999
John	a/5b/8071/8813/e13/a1/999
Peter	a/5b/8071/8813/e13/a1/999
bernard	62562dc9ctc04330ct1226b
testing1	62552a8076304c14521a08
Patrick	a/5b/8071/8813/e13/a1/999

If the admin wants to view the leaderboard for all accounts, they can click on the button "LeaderBoard" at the navigation bar at the top of the admin page. After the admin clicks on the button, the username and score for all accounts will be displayed in a table.

The screenshot shows a table titled "LeaderBoard" with two columns: "Username" and "Score". The data is as follows:

Username	Score
Peter	4
gamedemo	3
Brian	3
Jacky	3
JapaneseHentai	3
Thompson	2
jacky	2
brian1234	0

If the admin wants to reset the password of an account, they can click on the button "Reset user password" at the navigation bar at the top of the admin page. After the admin clicks on the button, there will be an input box for the admin to input the username of the account and a "Submit" button after the input box for the admin to submit the username. After the admin has filled in the username, he or she should click the "Submit" button. If the input box is filled, the username will be sent to the server and reset the password of the account, else an alert will be applied to alert the admin that the input box is not filled.

The screenshot shows a form titled "Reset User Password" with two fields: "Username:" and a "Submit" button.

If the admin wants to delete the gameplay score of an account, they can click on the button "Delete user gameplay" at the navigation bar at the top of the admin page. After the admin clicks on the button, there will be an input box for the admin to input the username of the account and a "Submit" button after the input box for the admin to submit the username. After the admin has filled in the username, he or she should click the "Submit" button. If the input box is filled, the username will be sent to the server and reset the password of the account, else an alert will be applied to alert the admin that the input box is not filled.

The screenshot shows a form titled "Delete user's gameplay" with two fields: "Username:" and a "Submit" button.

If the admin wants to delete an account, they can click on the button "Delete user account" at the navigation bar at the top of the admin page. After the admin clicks on the button, there will be an input box for the admin to input the username of the account and a "Submit" button after the input box for the admin to submit the username. After the admin has filled in the username, he or she should click the "Submit" button. If the input box is filled, the username will be sent to the server and reset the password of the account, else an alert will be applied to alert the admin that the input box is not filled.

If the admin wants to log out of the account, they can click on the button "Log-out" at the navigation bar at the top of the admin page. After the admin clicks on the button, the website will log out of the admin account and bring the admin back to the login and registration page.

5 TEST

5.1 Test Overview and test plan

To ensure the reliability of our website, black-box testing is held to check if the website is acting as the specification stated. The testing will be separated based on pages on the website, which are the login and registration page, the home page, the admin page, and the game page. White box testing is also held to ensure the robustness of our website.

Detailed test cases could be found in Appendix 8.2.

5.2 Test case assumptions

Non-existing username: “spec”, “1234testing”

Existing user account: {username: testing, password: a1234567, email:12@abc.com, score: 1}

Existing admin account: {username: 1234, password: 1234 }

For testing the home page and game page, it is using the user account of “testing”

5.3 Result

The website only fails for test cases in 8.2.4.2 and passes all other test cases.

Note that for test case 1 in 8.2.3.3 game page, if the user clicks on the disabled 'play' button slightly earlier than the camera being ready, the user will go straightly into the game session immediately after the camera is indicated as ready, without clicking on the enabled 'play' button, but the camera is still ensured to be ready so this test case is passed.

6 Lessons Learned

Throughout this project, our team has gained experience in using various tools for web application development, including standard html/css/js, their working principle inside a browser, using mongo database. We have used git extensively for version control, which is vital for any software development projects.

We have also practiced the spiral development process model, where we consider every two week as a cycle to publish incremental prototypes and revise our designs. For instance, one of the milestone results is that we switched the game play from following the original game 'Hole in the Wall', where there will be a countdown for the player to fit their hand into the image within a few seconds, gaining them 1 point. To a new gameplay, the player needs to fit as many walls (images) as possible within 15 seconds, where 1 point is gained as soon as their hand is fitted into the image. Such that the game could establish competitions.

Through working together on the same project, we also get to understand why and how certain software engineering practices are important, such as commenting our code and files and writing tests. By having proper comments, it would be much easier for the other developers to understand the code, and ensure that mistakes like accidental removal of files will not happen as they would not misinterpret the purpose of the code. Finally, if we had written tests even before coding, we could eliminate errors within each module, and reduce the time consumed by identifying and locating the bugs.

Overall, it was a great opportunity to put the theories from the course into practice. We have gained a better understanding of the theories themselves, as well as the importance of acting in accordance with the software engineering principles.

7 Conclusion

The team has successfully reproduced the Hole in the Wall game in a compact, efficient package. With the wide variations of game stages and rigorous anti-cheat measures, the game is always fresh and challenging. Moreover, the implementation of user/ admin features allowed the game plays to be recorded and managed by the server-side, thus enabling the players to compete with each other in top scores.

Despite developing the software itself, we had also managed to host the project on a cloud service; which not only showed the efficiency of the program, but also proved how it's easily scalable. Considering that the software is compatible for both web and mobile environments, we believe the project has the potential to attain a huge range of players. All things considered, the project is a definite success in realizing an exciting game idea into practice.

8 Appendix

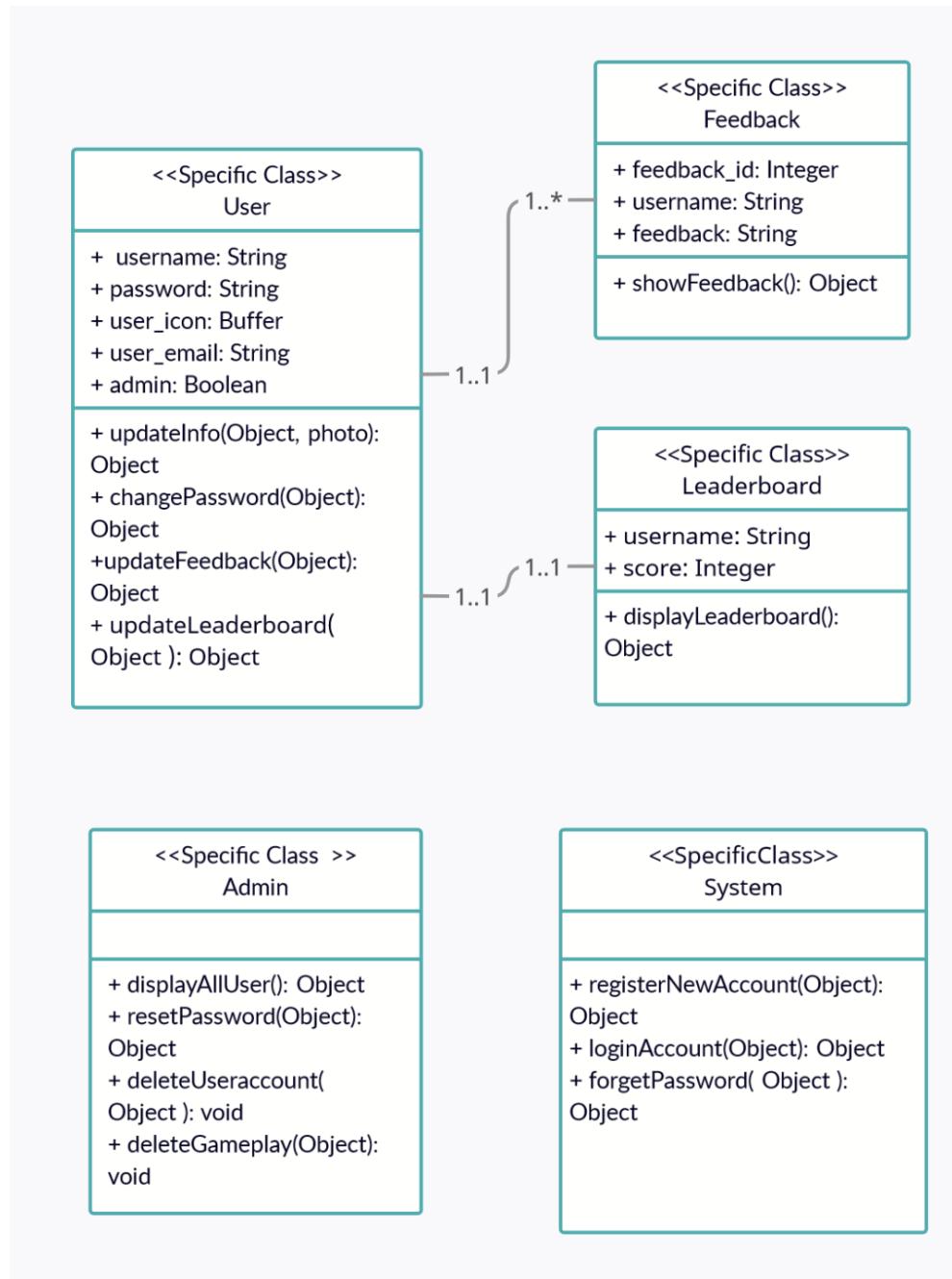
8.1 Database specification

8.1.1 All Database Functions

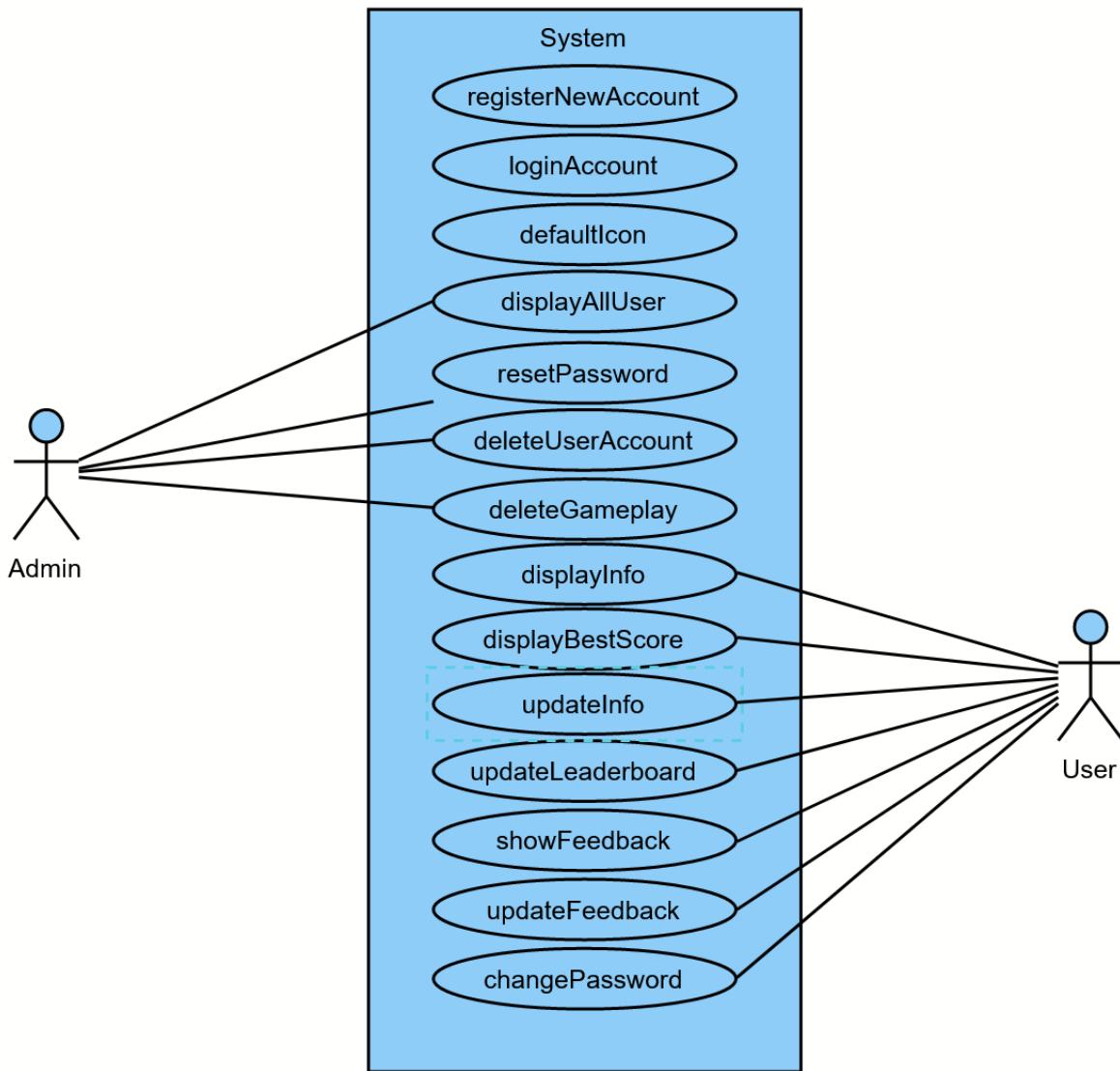
Function	Description
registerNewAccount(Obj)	Registers a new user account in the database
loginAccount(Obj)	Verifies login data and logs user in
defaultIcon(Obj)	Sets the default icon for an account
displayAllUser(Obj)	Returns all Users from the database
resetPassword(Obj)	Resets the password of the specified user
deleteUserAccount(Obj)	Deletes the specified user from the database
deleteGameplay(Obj)	Deletes the gameplay record of the specified user in the leaderboard
displayInfo(Obj)	Returns the user data from the database
displayBestScore(Obj)	Returns the user's best score
updateInfo(Obj)	modifies the user data from the specified information
updateLeaderboard(Obj)	adds the new score to the leaderboard, if it is better than the best previous score
showFeedback(Obj)	Returns the 5 latest feedbacks from the database
updateFeedback(Obj)	Adds the feedback to the database

changePassword(Obj)	Changes the user's password
---------------------	-----------------------------

8.1.2 Database Functions in UML Class diagram

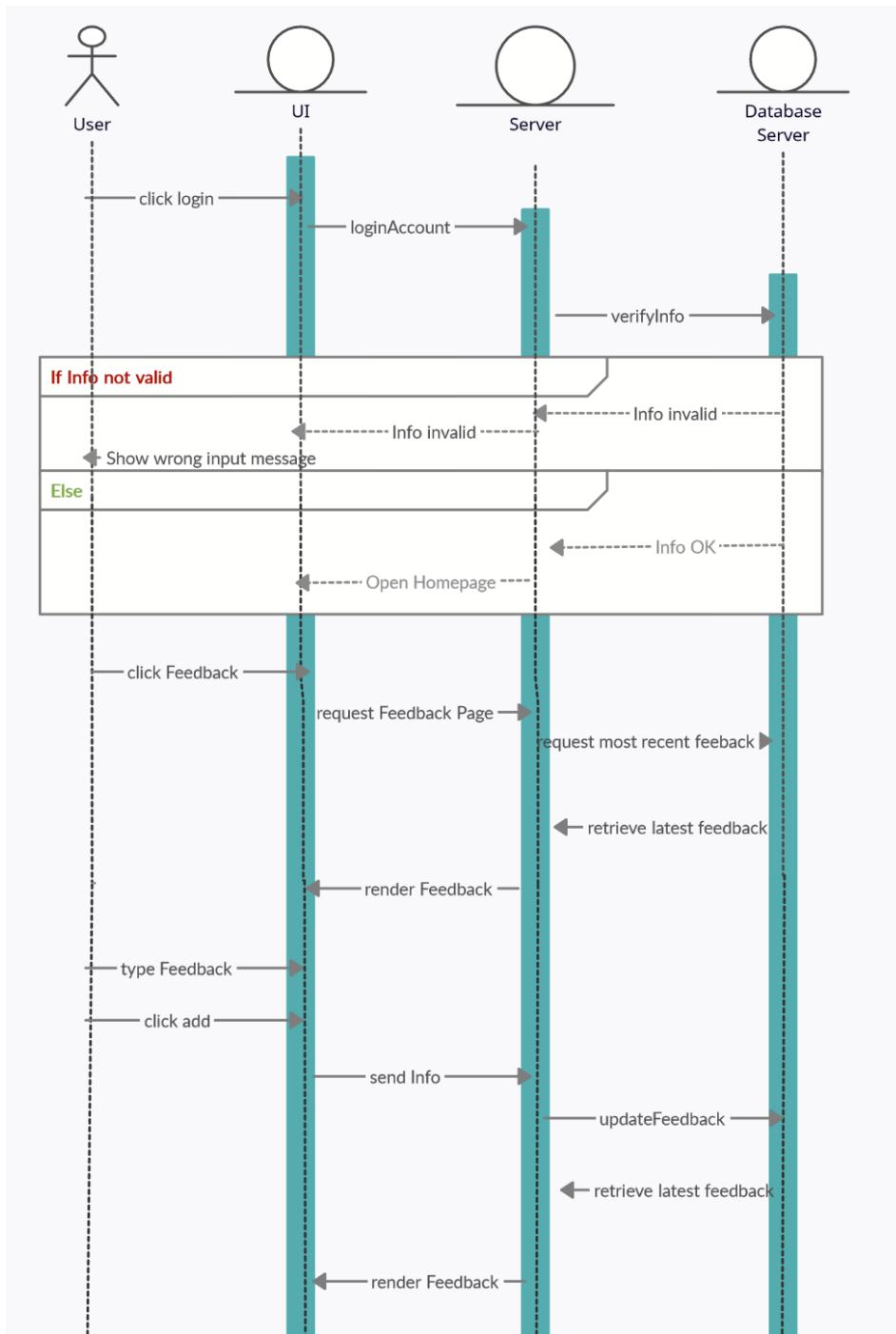


8.1.3 Database Functions in UML Use Case diagram



8.1.4 Database Functions in UML Sequence diagram

Case: Login, access the feedback page and add a feedback



8.2 Test cases

8.2.1 Login and registration page

8.2.1.1 Login

Test set	Purpose	Inputs	Expected Output	Pass/Fail Criteria
1	To check if users can get into an account without any input.	isAuthenticated = false Username = "" Password = ""	Raise an error	The test fails if the user can log into another account.
2	To check if users can get into an account without providing a password.	isAuthenticated = false Username = "testing" Password = ""	Raise an error	The test fails if the user can log into another account.
3	To check if users can get into an account without providing a username.	isAuthenticated = false Username = "" Password = "xxxx0000"	Raise an error	The test fails if the user can log into another account.
4	To check if users can log in with a valid username and password pair.	isAuthenticated = false Username = "testing" Password = "xxxx0000"	Log into the user "testing" account and display the homepage	The test fails if the user cannot log in or log into another account.
5	To check if users can log in with an invalid username and password pair.	isAuthenticated = false Username = "testing" Password = "!xxx0000"	Raise an error	The test fails if the user can log into the "testing" account.
6	To check if the admin can log in with a valid username and password pair.	isAuthenticated = false Username = "1234" Password = "1234"	Log into the admin account and display the admin page	The test fails if the admin cannot log in or log into another account.
7	To check if an admin can log in with an invalid username and password pair.	isAuthenticated = false Username = "1234" Password = "!234"	Raise an error	The test fails if the user can log into the admin account.

8.2.1.2 Registration

Test set	Purpose	Inputs	Expected Output	Pass/Fail Criteria
1	To check if users can create an account without any input.	isAuthenticated = false Username = "" Password = "" Confirm password = "" Email = ""	Raise an error	The test fails if the user can create an account.

2	To check if users can create an account without a username.	isAuthenticated = false Username = “” Password = “a1234567” Confirm password = “a1234567” Email = “12@abc.com”	Raise an error	The test fails if the user can create an account.
3	To check if users can create an account without a password.	isAuthenticated = false Username = “spec” Password = “” Confirm password = “a1234567” Email = “12@abc.com”	Raise an error	The test fails if the user can create an account.
4	To check if users can create an account without a confirmed password.	isAuthenticated = false Username = “spec” Password = “a1234567” Confirm password = “” Email = “12@abc.com”	Raise an error	The test fails if the user can create an account.
5	To check if users can create an account without an Email.	isAuthenticated = false Username = “spec” Password = “a1234567” Confirm password = “a1234567” Email = “”	Raise an error	The test fails if the user can create an account.
6	To check if users can create an account with an existing username.	isAuthenticated = false Username = “testing” Password = “a1234567” Confirm password = “a1234567” Email = ”12@abc.com”	Raise an error	The test fails if the user can create an account.
7	To check if users can create an account with an invalid password.	isAuthenticated = false Username = “spec” Password = “12345678” Confirm password = “12345678” Email = ”12@abc.com”	Raise an error	The test fails if the user can create an account.
8	To check if users can create an account with a different password and confirm password.	isAuthenticated = false Username = “spec” Password = “a1234567” Confirm password = “b1234567” Email = ”12@abc.com”	Raise an error	The test fails if the user can create an account.
9	To check if users can create an account with	isAuthenticated = false Username = “spec”	Raise an error	The test fails if the user can create an

	an invalid email.	Password = "a1234567" Confirm password = "a1234567" Email = "12#abc.com"		account.
10	To check if users can create an account with correct input.	isAuthenticated = false Username = "spec" Password = "a1234567" Confirm password = "a1234567" Email = "12@abc.com"	Create an account with username "spec", password, "a1234567" , and email, "12@abc.com"	The test fails if the user fails to create an account.

8.2.2 Home page

8.2.2.1 Navigation bar

Test set	Purpose	Inputs	Expected Output	Pass/Fail Criteria
1	To check if the website could display user information.	Click on "Your Info"	Display the "Your Info" page with correct information.	The test fails if it displays the wrong page or information.
2	To check if the website could display the form for modifying user information.	Click on "Modify your Info"	Display the "Modify your Information" page.	The test fails if it displays the wrong page.
3	To check if the website could display the form for resetting passwords.	Click on "Reset Password"	Display the "Reset password" page.	The test fails if it displays the wrong page.
4	To check if the website could display the feedback from others.	Click on "Feedbacks"	Display the "Feedbacks" page.	The test fails if it displays the wrong page or information.
5	To check if the website could reach the game page.	Click on "Start Game"	Display the game page.	The test fails if it displays the wrong page.
6	To check if the website could log out the user account.	Click on "Log-out"	Log out the account and display the login and registration page.	The test fails if it does not log out of the account or go back to the login and registration page.
7	To check if the website changes to dark mode.	mode = light Click on the sun icon	The website changes to dark	The test fails if it does not change the

			mode and stays on the page.	mode or display another page.
8	To check if the website changes to light mode.	mode = dark Click on the sun icon	The website changes to light mode and stays on the page.	The test fails if it does not change the mode or display another page.
9	To check if the website could display the starting page.	Click on “Wall Hole”	Display the starting page.“	The test fails if it displays the wrong page.

8.2.2.2 Starting page

Test set	Purpose	Inputs	Expected Output	Pass/Fail Criteria
1	To check if the website could reach the game page.	Click on “Start Game”	Display the game page.	The test fails if it displays the wrong page.

8.2.2.3 Your Info

Test set	Purpose	Inputs	Expected Output	Pass/Fail Criteria
1	To check if the user information is correctly displayed	(login)		The test fails if it displays the wrong user information.
2	To check if the website could display the form for modifying user information.	Click on “Update Info”	Display the “Modify your Information” page.	The test fails if it displays the wrong page.
3	To check if the website could display the form for resetting passwords.	Click on “Reset Password”	Display the “Reset password” page.	The test fails if it displays the wrong page.

8.2.2.4 Modify your Info

Test set	Purpose	Inputs	Expected Output	Pass/Fail Criteria
1	To check if users can modify user information without any input.	Username = “” Email address= “” New Icon = “”	Raise an error	The test fails if any user information is changed.
2	To check if users can	Username = “1234”	Raise an error	The test fails if any

	modify the username with an existing username.	Email address= “” New Icon = “”		user information is changed.
3	To check if users can modify the username with the same username.	Username = “testing” Email address= “” New Icon = “”	Raise an error	The test fails if any user information is changed.
4	To check if users can modify the username without other input.	Username = “testing1” Email address= “” New Icon = “”	The username is changed to “testing1” and displays the login and registration page.	The test fails if it raises an error or does not change the user information.
5	To check if users can modify email with an invalid email.	Username = “” Email address= “12#abc.com” New Icon = “”	Raise an error	The test fails if any user information is changed.
6	To check if users can modify email with the same email.	Username = “” Email address= “12@abc.com” New Icon = “”	Raise an error	The test fails if any user information is changed.
7	To check if users can modify email without other input.	Username = “” Email address = “123@abc.com” New Icon = “”	The email address is changed to “123@abc.com” and displays the login and registration page.	The test fails if it raises an error or does not change the user information.
8	To check if users can upload an icon without other input.	Username = “” Email address = “” New Icon = image	The Icon of the user is changed to the uploaded image and displays the login and registration page.	The test fails if it raises an error or does not change the user information.
9	To check if users can upload an icon, modify username and email address.	Username = “testing2” Email address = “1234@abc.com” New Icon = image	The information of the user is changed as input and displays the login and registration page.	The test fails if it raises an error or does not change the user information.

8.2.2.5 Reset Password

Test set	Purpose	Inputs	Expected Output	Pass/Fail Criteria
1	To check if users can modify their password without any input.	Old password = "" New password = "" New password again = ""	Raise an error	The test fails if the password is changed.
2	To check if users can modify their password without a correct old password.	Old password = "c1234567" New password = "b1234567" New password again = "b1234567"	Raise an error	The test fails if the password is changed.
3	To check if users can modify their password with the same old and new password.	Old password = "a1234567" New password = "a1234567" New password again = "a1234567"	Raise an error	The test fails if the password is changed.
4	To check if users can modify their password with an invalid new password.	Old password = "a1234567" New password = "12345678" New password again = "12345678"	Raise an error	The test fails if the password is changed.
5	To check if users can modify their password with a different new password and new password again.	Old password = "a1234567" New password = "b1234567" New password again = "c1234567"	Raise an error	The test fails if the password is changed.
6	To check if users can modify their password without an old password.	Old password = "" New password = "b1234567" New password again = "b1234567"	Raise an error	The test fails if the password is changed.
7	To check if users can modify their password without a new password.	Old password = "a1234567" New password = "" New password again = "b1234567"	Raise an error	The test fails if the password is changed.
8	To check if users can modify their password	Old password = "a1234567"	Raise an error	The test fails if the password is changed.

	without a new password again.	New password = "b1234567" New password again = ""		
9	To check if users can modify their password with the correct old, new password, and new password again.	Old password = "a1234567" New password = "b1234567" New password again = "b1234567"	The password will be changed to "b1234567" and display the login and registration page.	The test fails if it raises an error or does not change the password.

8.2.2.6 Feedbacks

Test set	Purpose	Inputs	Expected Output	Pass/Fail Criteria
1	To check if users can post feedback without any input.	Feedback = ""	Raise an error	The test fails if the feedback is posted.
2	To check if users can post feedback with valid input.	Feedback = "Hello World"	Feedback is posted and it is shown at the top of the list for feedback	The test fails if the raises an error or the feedback does not display.

8.2.3 Game page

Test set	Purpose	Inputs	Expected Output	Pass/Fail Criteria
1	To check if users can enter the game session without clicking the activated play button.	Clicking the disabled 'play' button (displayed as 'preparing camera')	Nothing happens	The test fails if the user enters the game session while the camera input is not yet ready.
2	To check if the hand detection messages are displayed correctly.	Putting the hand outside of/far away from/close to the camera	Display 'hand not detected', 'hand is too far away', 'respectively' after game start countdown is over	The test fails if any of these messages are displayed wrongly.

8.2.4 Admin page

8.2.4.1 Navigation bar

Test set	Purpose	Inputs	Expected Output	Pass/Fail Criteria
1	To check if the website could display all user accounts.	Click on “All User account”	Display all usernames and their user ID in a table.	The test fails if it displays the wrong page or information.
2	To check if the website could display the leaderboard.	Click on “LeaderBoard”	Display all usernames and its score in a table.	The test fails if it displays the wrong page or information.
3	To check if the website could display the form for resetting the user's password.	Click on “Reset User Password”	Display the “Reset User password” page.	The test fails if it displays the wrong page.
4	To check if the website could display the form for deleting the user's gameplay.	Click on “Delete user's gameplay”	Display the “Delete user's gameplay” page.	The test fails if it displays the wrong page or information.
5	To check if the website could display the form for deleting the user's account.	Click on “Delete user's account”	Display the “Delete user's account” page.	The test fails if it displays the wrong page.
6	To check if the website could log out the user account.	Click on “Log-out”	Log out the account and display the login and registration page.	The test fails if it does not log out of the account or go back to the login and registration page.

8.2.4.2 Reset User Password

Test set	Purpose	Inputs	Expected Output	Pass/Fail Criteria
1	To check if the admin can reset the user's password without any input.	Username = “”	Raise an error	The test fails if any user's password is reset.
2	To check if the admin can reset the user's password with a non-existing username.	Username = “1234testing”	Raise an error	The test fails if any user's password is reset.

3	To check if the admin can reset the user's password with the valid username.	Username = "testing"	The password for "spec" is reseted and an email is sent to user "spec".	The test fails if the user's password is not reset, another user's password is reset or the email is not sent.
---	--	----------------------	---	--

8.2.4.3 Delete user's gameplay

Test set	Purpose	Inputs	Expected Output	Pass/Fail Criteria
1	To check if the admin can delete the user's gameplay without any input.	Username = ""	Raise an error	The test fails if any user's gameplay is deleted.
2	To check if the admin can delete the user's gameplay with a non-existing username.	Username = "1234testing"	Raise an error	The test fails if any user's gameplay is deleted.
3	To check if the admin can delete the user's gameplay with a valid username.	Username = "testing"	The gameplay for "testing" is deleted.	The test fails if the user's gameplay is not deleted or another user's gameplay is deleted.

8.2.4.3 Delete user's account

Test set	Purpose	Inputs	Expected Output	Pass/Fail Criteria
1	To check if the admin can delete the user's account without any input.	Username = ""	Raise an error	The test fails if any user's account is deleted.
2	To check if the admin can delete the user's account with a non-existing username.	Username = "1234testing"	Raise an error	The test fails if any user's account is deleted.
3	To check if the admin can delete the user's account with the valid username.	Username = "spec"	The account for "spec" is deleted.	The test fails if the user's account is not deleted or another user's account is deleted.

8.2.5 Page routings

Test set	Purpose	Inputs	Expected Output	Pass/Fail Criteria
1	To check if the user can reach the home page without logging in.	isAuthenticated = false UML = “http://localhost:3000/user”	Redirect to the login and registration page.	The test passes if and only if the login and registration page is displayed.
2	To check if the admin can reach the admin page without logging in.	isAuthenticated = false UML = “http://localhost:3000/admin”	Redirect to the login and registration page.	The test passes if and only if the login and registration page is displayed.
3	To check if the user can skip the login and registration page after logging in.	isAuthenticated = true isAdmin = false UML = “http://localhost:3000”	Redirect to the home page.	The test passes if and only if the home page is displayed.
4	To check if the admin can skip the login and registration page after logging in.	isAuthenticated = true isAdmin = true UML = “http://localhost:3000”	Redirect to the admin page.	The test passes if and only if the admin page is displayed.

8.2.6 Unexpected action

Test set	Purpose	Inputs	Expected Output	Pass/Fail Criteria
1	To check if the login and registration page will display correctly when repeatedly changing between login and registration while resizing.	Resize the browser while changing between login and registration at the login and registration page.	Display correctly at any time.	The test fails if the rules or the form displayed incorrectly in the process.
2	To check if the login and registration page will display properly if the user clicks outside the alert box instead of the “OK” button.	Directly check “GO” on the login page without filling in the input box and click outside the alert box.	Display the form for login again.	The test fails if the form does not display or display other pages.

8.2 Images used as walls in the game

