



**NANYANG  
TECHNOLOGICAL  
UNIVERSITY**  

---

**SINGAPORE**

# **Drawdiculous**

## **Use Case Descriptions**

Version 1.1

Prepared by Team Luck

Bian Hengwei	U1923732B
He Yinan	U1922693C
Jin Han	U1922733A
Lee Yu Jie Melvin	U1922533G
Renganathan Ramasamy	U1922494L
Sannabhadti Shipra Deepak	U1822459L

## Document Change Record

Name	Date	Reason For Changes	Version
Bian Hengwei	8 Sep 2021	Initial version	1.0
Bian Hengwei	30 Oct 2021	Updated use cases	1.1

## Use Case List

Primary Actor	Use Cases
User, Firebase	1. Login
User, Firebase	2. Register
User, Firebase	3. Forgot Password
User, Server	4. Play Game
User, Server	5. Draw
User, Server	6. Update Leaderboard
User, Server	7. Update Drawing
User, Server	8. Update Guesses
User, Server	9. Join Game Room
User, Server	10. Create Game Room
User, Server	11. Delete Player

## Use Case Description

Use Case ID:	1		
Use Case Name:	Login		
Created By:	He Yinan	Last Updated By:	Bian Hengwei
Date Created:	7 September 2021	Date Last Updated:	8 September 2021

Actors:	User (initiating), Firebase
Description:	Allow users to login to the application to perform all functions provided by the application.
Trigger:	User starts to enter email or password
Preconditions:	1. User must have an valid registered account
Postconditions:	1. User is successfully logged in 2. User is redirected to Home page
Normal Flow:	1.0 1. User enters email address and password 2. User clicks the “Login” button 3. Application takes user input and send to Firebase 4. Server verifies email address and password 5. Firebase retrieves user information from database 6. Firebase responds a success message to application 7. User is logged into the application 8. User is redirected to Home page
Alternative Flows:	1.1 Wrong Password 1. Wrong password triggers exception 2. Server respond a “Wrong Password” message to application 3. Error message “Invalid email address or password” displayed 4. Use case returns to step 1.0.1
Exceptions:	1.0.E.1 Failed to connect to Firebase 1. App shows “Failed to connect to database. Please check internet connection”
Includes:	-
Priority:	High
Frequency of Use:	Once a week per user
Business Rules:	-
Special Requirements:	-

Assumptions:	-
Notes and Issues:	-

Use Case ID:	2		
Use Case Name:	Register		
Created By:	He Yinan	Last Updated By:	Bian Hengwei
Date Created:	7 September 2021	Date Last Updated:	8 September 2021

Actors:	User (initiating), Firebase
Description:	Allow user to register for an account to login
Trigger:	User clicks "Register" button
Preconditions:	<ol style="list-style-type: none"> <li>1. User must have downloaded the Drawdiculous application</li> <li>2. User must have a valid email address</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. User successfully signed up</li> <li>2. System will redirect user to Login page</li> </ol>
Normal Flow:	<p>2.0</p> <ol style="list-style-type: none"> <li>1. User clicks "Register" button</li> <li>2. Application prompts the user to enter "Username", "Email Address", "Password"</li> <li>3. User enters all required inputs and clicks "Confirm" button</li> <li>4. Application sends user information to Firebase</li> <li>5. Firebase sends registration success message to application</li> <li>6. Application displays message "Registration is successful"</li> </ol>
Alternative Flows:	<p>2.1 Email is already used for registration</p> <ol style="list-style-type: none"> <li>1. Firebase sends registration fail message (email already exist) to application</li> <li>2. Application display error message "Email already in use"</li> <li>3. Use case return to step 2.0.2</li> </ol> <p>2.2 Invalid email</p> <ol style="list-style-type: none"> <li>1. Firebase sends invalid email message to application</li> <li>2. Application display error message "Invalid email"</li> <li>3. Use case return to step 2.0.2</li> </ol> <p>2.3 Invalid password</p> <ol style="list-style-type: none"> <li>1. Firebase sends invalid password message to application</li> </ol>

	2. Application display error message “Password not secure” 3. Use case return to step 2.0.2
Exceptions:	2.0.E.1 Failed to connect to Firebase 1. App shows “Failed to connect to database. Please check internet connection”
Includes:	-
Priority:	High
Frequency of Use:	Once per user
Business Rules:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	3		
Use Case Name:	Forgot Password		
Created By:	Bian Hengwei	Last Updated By:	Bian Hengwei
Date Created:	7 Sep 2021	Date Last Updated:	7 Sep 2021

Actors:	User(initiating), Firebase
Description:	User forgets the account password before logging in and changes using email
Trigger:	User clicks “Forgot Password” button on user profile page
Preconditions:	1. User has a valid account
Postconditions:	1. Password is successfully updated
Normal Flow:	3.0 <ol style="list-style-type: none"> <li>1. User keys in the account email address</li> <li>2. App forwards the email address to Firebase</li> <li>3. Firebase checks the validity of the email address</li> <li>4. Firebase sends a change password link to the email</li> <li>5. User keys in a new password using the link</li> <li>6. Firebase updates the new password</li> </ol>
Alternative Flows:	3.1 Email address is not valid <ol style="list-style-type: none"> <li>1. App shows “invalid email”</li> <li>2. Goes to step 3.0.1</li> </ol>
Exceptions:	3.0.E.1 Failed to connect to Firebase <ol style="list-style-type: none"> <li>1. App shows “Failed to connect to database. Please check internet connection”</li> </ol>
Includes:	-
Priority:	Low
Frequency of Use:	Once a year per user
Business Rules:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-



Use Case ID:	4		
Use Case Name:	Play Game		
Created By:	Jin Han	Last Updated By:	Jin Han
Date Created:	8 Sep 2021	Date Last Updated:	8 Sep 2021

Actors:	User (initiating), Server
Description:	User plays the drawing game
Trigger:	Host player clicks on “Start Game”
Preconditions:	<ol style="list-style-type: none"> <li>1. User has logged in into a valid account</li> <li>2. Game room contains at least two players</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. User with the most points declared winner</li> </ol>
Normal Flow:	<p>4.0</p> <ol style="list-style-type: none"> <li>1. Server assigns first player to be “artist”</li> <li>2. Server provides user with a prompt</li> <li>3. User draws the prompt</li> <li>4. Server displays the drawing to other players in real-time</li> <li>5. Server awards user points based on number of players who answered correctly</li> <li>6. Server chooses the next player in line to be drawer</li> <li>7. Repeat steps 4.0.2 to 4.0.7 until all players have been assigned as “artist” once</li> <li>8. Server calculates which player has the most points</li> <li>9. Server declares player with most points as winner</li> </ol>
Alternative Flows:	<p>4.1 User is not assigned as “artist”</p> <ol style="list-style-type: none"> <li>1. Server assigns user to be guesser</li> <li>2. Server displays the “artist’s” drawing to user in real-time</li> <li>3. User guesses the drawing and types their answer into the answer box</li> <li>4. Server determines if user’s answer is correct</li> <li>5. Goes to step 4.1.3 if answer is incorrect</li> <li>6. Server awards user points based on the speed at which they got the answer. If user did not get the answer within the time limit, user does not get any points</li> <li>7. Goes to step 4.0.7</li> </ol>
Exceptions:	-
Includes:	Draw, Update Leaderboard, Update Drawing, Make Guesses

Priority:	High
Frequency of Use:	Three times a day per user
Business Rules:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	5		
Use Case Name:	Draw		
Created By:	Bian Hengwei	Last Updated By:	Bian Hengwei
Date Created:	30 Oct 2021	Date Last Updated:	30 Oct 2021

Actors:	User (initiating), Server
Description:	User draws on canvas
Trigger:	User is assigned as the “artist”
Preconditions:	<ol style="list-style-type: none"> <li>1. User has logged in into a valid account</li> <li>2. Game room contains at least two players</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. User canvas is updated</li> <li>2. Server is updated with the new information</li> </ol>
Normal Flow:	<p>5.0</p> <ol style="list-style-type: none"> <li>1. User use finger to draw on the canvas</li> <li>2. System stores the stroke information</li> <li>3. System displays the strokes</li> <li>4. System sends the stroke information to the server</li> </ol>
Alternative Flows:	-
Exceptions:	-
Includes:	-
Priority:	High
Frequency of Use:	Three times a day per user
Business Rules:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	6		
Use Case Name:	Update Leaderboard		
Created By:	Jin Han	Last Updated By:	Bian Hengwei
Date Created:	8 Sep 2021	Date Last Updated:	30 Oct 2021

Actors:	User (initiating), Server
Description:	System updates the leaderboard
Trigger:	Server sends a update leaderboard request
Preconditions:	<ol style="list-style-type: none"> <li>1. User has logged in into a valid account</li> <li>2. User is currently in a game room</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. User views the points of each player</li> </ol>
Normal Flow:	6.0 <ol style="list-style-type: none"> <li>1. Server sends the updated leaderboard containing usernames and their corresponding updated points to the client</li> <li>2. Application displays the list of players and their corresponding points</li> <li>3. User views the points of each player</li> </ol>
Alternative Flows:	-
Exceptions:	-
Includes:	-
Priority:	Medium
Frequency of Use:	Three times per game
Business Rules:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	7		
Use Case Name:	Update Drawing		
Created By:	Bian Hengwei	Last Updated By:	Bian Hengwei
Date Created:	30 Oct 2021	Date Last Updated:	30 Oct 2021

Actors:	User (initiating), Server
Description:	Client updates the real-time canvas
Trigger:	Server sends a update drawing request
Preconditions:	<ol style="list-style-type: none"> <li>1. User has logged in into a valid account</li> <li>2. User is currently in a game room</li> <li>3. User is not the “artist”</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. Users sees the updated canvas</li> </ol>
Normal Flow:	<p>7.0</p> <ol style="list-style-type: none"> <li>1. Server receives the updated stroke information from “artist” client</li> <li>2. Server proceed these messages to other players in the same game</li> <li>3. Application process the list of stroke information one by one</li> <li>4. Canvas UI is updated accordingly</li> <li>5. User sees the updated canvas</li> </ol>
Alternative Flows:	-
Exceptions:	-
Includes:	-
Priority:	High
Frequency of Use:	20 times per second
Business Rules:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	8		
Use Case Name:	Update Guesses		
Created By:	Bian Hengwei	Last Updated By:	Bian Hengwei
Date Created:	30 Oct 2021	Date Last Updated:	30 Oct 2021

Actors:	User (initiating), Server
Description:	Client updates the chat box
Trigger:	User send a new guess
Preconditions:	<ol style="list-style-type: none"> <li>1. User has logged in into a valid account</li> <li>2. User is currently in a game room</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. User views the points and ranking of each player</li> </ol>
Normal Flow:	<p>8.0</p> <ol style="list-style-type: none"> <li>1. User enter something</li> <li>2. User clicks “Send”</li> <li>3. Application requests the server to validate the answer</li> <li>4. Server confirms that the answer is correct</li> <li>5. Server updates the leaderboard</li> <li>6. Server response to update the chat box with “Bingo”</li> <li>7. Applications update the chat boxes</li> </ol>
Alternative Flows:	<p>8.1 Wrong guess</p> <ol style="list-style-type: none"> <li>1. Server response to update the chat box with the guess</li> <li>2. Applications update the chat boxes</li> </ol>
Exceptions:	-
Includes:	-
Priority:	High
Frequency of Use:	Three times per round
Business Rules:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	9		
Use Case Name:	Join Game Room		
Created By:	Shipra	Last Updated By:	Bian Hengwei
Date Created:	8 Sep 2021	Date Last Updated:	30 Oct 2021

Actors:	User (initiating), Server
Description:	User joins a game room
Trigger:	User clicks on “Join Game Room” button
Preconditions:	1. User has logged in into a valid account
Postconditions:	1. User is redirected to an existing game room
Normal Flow:	<p>9.0</p> <ol style="list-style-type: none"> <li>1. User receives a prompt to enter Game Room ID and (optional) password</li> <li>2. User keys in a Game Room ID</li> <li>3. User keys in a password</li> <li>4. Application takes user input and send to server</li> <li>5. Server verifies room details</li> <li>6. Server responds with a success message to application</li> <li>7. Server updates all other players in the game room</li> <li>8. User joins the Game Room</li> </ol>
Alternative Flows:	<p>9.1 Random Game room</p> <ol style="list-style-type: none"> <li>1. User clicks “Join Random Room”</li> <li>2. Server searches for a random public game room for the user</li> <li>3. Goes to 9.0.5</li> </ol> <p>9.2 Public Game Room</p> <ol style="list-style-type: none"> <li>1. User does not key in the password.</li> <li>2. Goes to 9.0.4</li> </ol> <p>9.3 Invalid Game Room ID/Password</p> <ol style="list-style-type: none"> <li>1. Invalid Game Room ID/Password triggers exception</li> <li>2. Server responds with an Error message to application</li> <li>3. Error message displayed</li> <li>4. Use case returns to 9.0.1</li> </ol>
Exceptions:	-
Includes:	-
Priority:	High
Frequency of Use:	Three times a day per user

Business Rules:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-



Use Case ID:	10		
Use Case Name:	Create Game Room		
Created By:	Jin Han	Last Updated By:	Jin Han
Date Created:	8 Sep 2021	Date Last Updated:	8 Sep 2021

Actors:	User (initiating), Server
Description:	User creates a game room (Public/Private)
Trigger:	User clicks on "Create Room" button
Preconditions:	1. User has logged in into a valid account
Postconditions:	1. Either a public or private game room is created 2. User is redirected to newly created game room
Normal Flow:	10.0 1. User is directed to the room creation page 2. User enters a password 3. User clicks on the "Finish" button 4. Server generates a unique room ID 5. Server creates a private game room 6. User is directed to newly created game room
Alternative Flows:	10.1 User creates public game room 1. User does not enter the password 2. User clicks on the "Finish" button 3. Server generates a unique room ID 4. Server creates a public game room 5. User is directed to newly created game room
Exceptions:	-
Includes:	-
Priority:	High
Frequency of Use:	One time a day per user
Business Rules:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	11		
Use Case Name:	Delete Player		
Created By:	Jin Han	Last Updated By:	Jin Han
Date Created:	8 Sep 2021	Date Last Updated:	8 Sep 2021

Actors:	User (initiating), Server
Description:	User removes a player from the game
Trigger:	Host player chooses a player and clicks on “X”
Preconditions:	<ol style="list-style-type: none"> <li>1. User has logged in into a valid account</li> <li>2. User is the host player of the game room.</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. Player chosen by the host is removed from the game room</li> </ol>
Normal Flow:	<p>11.0</p> <ol style="list-style-type: none"> <li>1. User clicks on the “X” button next to the name of the player that they wish to remove from the game</li> <li>2. Server removes the chosen player from the game</li> <li>3. Server updates all players in the room about the removal</li> <li>4. Application updates the game room UI for the removal</li> </ol>
Alternative Flows:	-
Exceptions:	-
Includes:	-
Priority:	Low
Frequency of Use:	Three times a day
Business Rules:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-