Use Cases

for

Drawdiculous

Version 1.0 approved

Prepared by: Bian Hengwei, He Yinan, Jin Han, Lee Yu Jie Melvin, Renganathan Ramasamy, Sannabhadti Shipra Deepak

Team Luck

7 September 2021

Revision History

Name	Date	Reason For Changes	Version
Bian Hengwei	8 Sep 2021	Create descriptions	1.0.0

Use Case List

Primary Actor	***Use Cases***
User, Firebase	1. Login
User, Firebase	2. Register
User, Firebase	3. Edit Profile
User, Firebase	4. Change Password
User, Firebase	5. Forget Password
User, Server	6. Play Game
User, Server	7. View Leaderboard
User, Server	8. Join Game Room
Server, Database	9. Access and Modify Data
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:	1. User must have downloaded the Drawdiculous application 2. User must have a valid email address
Postconditions:	1. User successfully signed up 2. System will redirect user to Login page
Normal Flow:	2.0 1. User clicks “Register” button 2. Application prompts the user to enter “Username”, “Email Address”, “Password” and “Confirm password” 3. User enters all required inputs and clicks “Confirm” button 4. Application sends user information to Firebase 5. Firebase verify email and password is valid 6. Verification is sent to user’s email 7. User clicks the link in verification email to verify email 8. Firebase sends registration success message to application 9. Application displays message “Registration is successful” 10. User is redirected to Login page
Alternative Flows:	2.1 Email is already used for registration 1. Firebase sends registration fail message (email already exist) to application 2. Application display error message “Email already in use” 3. Use case return to step 2.0.2 2.2 Invalid email 1. Firebase sends invalid email message to application 2. Application display error message “Invalid email” 3. Use case return to step 2.0.2 2.3 Invalid password 1. Firebase sends invalid password message to application 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:	Edit Profile		
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 edit their personal information
Trigger:	User clicks “Edit Profile” button
Preconditions:	1. User must have an existing account and is logged in
Postconditions:	1. User profile is updated according to the changes user made
Normal Flow:	3.0 1. User modifies information 2. User clicks “Confirm” button 3. Application sends updated information to Firebase 4. Updated user information is saved to database 5. Firebase sends update success message to application 6. Application displays message “Username is successfully updated”
Alternative Flows:	3.1 User modifies email address 1. Server verify email is valid 2. Verification is sent to user’s email 3. User clicks the link in verification email to verify email 4. Updated email is saved to database 5. Server sends update success message to application 6. Application displays message “Email is successfully updated”
Exceptions:	3.0.E.1 Failed to connect to Firebase 1. App shows “Failed to connect to database. Please check internet connection”
Includes:	-
Priority:	Low
Frequency of Use:	Twice a year per user
Business Rules:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	4		
Use Case Name:	Change 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 changes the account password
Trigger:	User clicks “Change Password” button on user profile page
Preconditions:	1. User has logged in into a valid account
Postconditions:	1. User password is successfully updated to Firebase
Normal Flow:	4.0 1. User keys in the new password 2. User keys in again to confirm the new password 3. User clicks “Confirm” button to submit the new password 4. App verifies the two passwords matches and forwards to Firebase 5. Firebase updates the new password 6. User is logged out from the system and is required to login again using the new password
Alternative Flows:	4.1 Two passwords does not match 1. App shows “password does not match” 2. App requires user to re-enter the passwords 3. Goes to step 4.0.0
Exceptions:	4.0.E.1 Failed to connect to Firebase 1. App shows “Failed to connect to database. Please check internet connection”
Includes:	-
Priority:	Low
Frequency of Use:	Twice a year per user
Business Rules:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	5		
Use Case Name:	Forget 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 “Forget Password” button on user profile page
Preconditions:	1. User has a valid account
Postconditions:	1. Password is successfully updated
Normal Flow:	5.0 1. User keys in the account email address 2. App forwards the email address to Firebase 3. Firebase checks the validity of the email address 4. Firebase sends a change password link to the email 5. User keys in a new password using the link 6. Firebase updates the new password
Alternative Flows:	5.1 Email address is not valid 1. App shows “invalid email” 2. Goes to step 5.0.1
Exceptions:	5.0.E.1 Failed to connect to Firebase 1. App shows “Failed to connect to database. Please check internet connection”
Includes:	-
Priority:	Low
Frequency of Use:	Once a year per user
Business Rules:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	6		
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:	1. User has logged in into a valid account 2. Game room contains at least two players
Postconditions:	1. User with the most points declared winner
Normal Flow:	6.0 1. Server assigns first player to be drawer 2. Server provides user with 3 prompts 3. User selects 1 of the 3 available prompts 4. User draws the selected prompt 5. Server displays the drawing to other players in real-time 6. Server awards user points based on number of players who answered correctly, when the time limit is up or when all other players have guessed the answer 7. Server chooses the next player in line to be drawer 8. Repeat steps 6.0.2 to 6.0.7 until all players have been assigned as drawer once 9. Server calculates which player has the most points 10. Server declares player with most points as winner
Alternative Flows:	6.1 User is not assigned as drawer 1. Server assigns user to be guesser 2. Server displays the drawer’s drawing to user in real-time 3. User guesses the drawing and types their answer into the answer box 4. Server determines if user’s answer is correct 5. Goes to step 6.1.3 if answer is incorrect 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 7. Goes to step 6.0.7
Exceptions:	-
Includes:	Access and Modify Data
Priority:	High
Frequency of Use:	Three times a day per user
Business Rules:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	7		
Use Case Name:	View Leaderboard		
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 views the leaderboard
Trigger:	User clicks on the “View Leaderboard” button
Preconditions:	1. User has logged in into a valid account 2. User is currently in a game room.
Postconditions:	1. User views the points and ranking of each player
Normal Flow:	7.0 1. Server compares the number of points each player has accumulated 2. Server ranks every player according to their points 3. Application displays the list of players, in order of their ranking, and their corresponding points 4. User views the ranking of each player 5. User clicks on “View Leaderboard” again to remove the list of rankings
Alternative Flows:	7.1 Multiple players have the same number of points 1. Server gives the same rank to players with the same number of points 2. Application displays the order of players such that for players with the same rank, they are ordered according to their order of entry 3. Goes to step 7.0.4
Exceptions:	-
Includes:	-
Priority:	Medium
Frequency of Use:	Three times per game
Business Rules:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	8		
Use Case Name:	Join Game Room		
Created By:	Shipra	Last Updated By:	Bian Hengwei
Date Created:	8 Sep 2021	Date Last Updated:	8 Sep 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:	8.0 1. User receives a prompt to enter Game Room ID 2. User keys in a Game Room ID 3. Application takes user input and send to server 4. Server verifies Game Room ID 5. Server responds with a success message to application 6. User receives a prompt for password 7. User keys in a password 8. Application takes user input and send to server 9. Server verifies password 10. Server responds with a success message to application 11. User joins the Game Room
Alternative Flows:	8.1 Random Game room 1. User clicks “Random Room” 2. Server searches for a random game room for the user 3. Jump to 8.0.11 8.2 Public Game Room 1. User receives a prompt to enter Game Room ID 2. User keys in a Game Room ID 3. Application takes user input and send to server 4. Server verifies Game Room ID 5. Server responds with a success message to application 6. Jump to 8.0.11 8.3 Invalid Game Room ID 1. Invalid Game Room ID triggers exception 2. Server responds with a “Invalid Game Room ID” message to application 3. Error message “Invalid Game Room ID” displayed 4. Use case returns to 8.0.1 8.4 Invalid Password 1. Invalid password triggers exception 2. Server responds with a “Invalid Password” message to application 3. Error message “Invalid Password” displayed 4. Use case returns to 8.0.6
Exceptions:	-

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

Use Case ID:	9		
Use Case Name:	Access and Modify Data		
Created By:	Shipra	Last Updated By:	Bian Hengwei
Date Created:	8 Sep 2021	Date Last Updated:	8 Sep 2021

Actors:	Server, Database
Description:	Server accesses and/or modifies data in the database
Trigger:	One of the following — “Create Game Room”, “Join Game Room”, “Delete Player”, “Play Game”— is successful
Preconditions:	1. Server requests for database access
Postconditions:	1. Database is updated accordingly 2. Information is returned to server
Normal Flow:	9.0 1. Server sends request to the Database to access and update table row for the specified game room 2. Request is received by the Database and table row is updated if requested 3. Data is returned to Server
Alternative Flows:	-
Exceptions:	9.0.E.1. Server cannot connect to Database 1. App shows “Failed to connect to database”
Includes:	-
Priority:	High
Frequency of Use:	250 times per hour
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 selects language of the game (English/ Chinese/ Malay/ Tamil) 3. User clicks on the “Private Room” checkbox 4. User clicks on the “Finish” button 5. Server generates a unique room ID and password 6. Server creates a private game room 7. User is directed to newly created game room
Alternative Flows:	10.1 User creates public game room 1. User does not click on the “Private Room” checkbox 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:	Access and Modify Data
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:	1. User has logged in into a valid account 2. User is the host player of the game room.
Postconditions:	1. Player chosen by the host is removed from the game room
Normal Flow:	11.0 1. User clicks on the “X” button next to the name of the player that they wish to remove from the game 2. Application displays a pop-up, confirming “Do you wish to remove this player from the game?” 3. User clicks on the “Yes” button 4. Server removes the chosen player from the game
Alternative Flows:	11.1 User chooses “No” during deletion confirmation 1. Server does not remove player from the game 2. Goes to step 11.0.1
Exceptions:	-
Includes:	Access and Modify Data
Priority:	Low
Frequency of Use:	Three times a day
Business Rules:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-