1

Title:	Display Login Error
Description:	Displays an error when login credentials don't match any in the database
System Under Design:	Squared System
Primary Actor:	Users
Participants:	Users
Goal:	to Inform the person trying to login that the information they entered in invalid
Following Use Cases:	Extends "Login"
Invariant:	Must be logged out, connected to internet
Precondition:	Must be logging in
Success Postcondition:	N/A

STEPS:	ALTERNATIVES:
1. User enters Login credentials 2. System compares the credentials entered to the credentials in the database 3. The system can't match the credentials to an account 4. The system displays a login error message	

2

Title:	Use Chat
Description:	The chat allows the players and potential spectators to communicate with each other with typed messages.
System Under Design:	Squared System
Primary Actor:	User sending the message
Participants:	User receiving the message
Goal:	Communicating with the opponent and it will reach the other user in less than 5 seconds.
Following Use Cases:	N/A
Invariant:	The user has a stable internet connection, is in a game, and has Javascript enabled
Precondition:	N/A
Success Postcondition:	N/A

STEPS:	ALTERNATIVES:
1. The user clicks on the chat bar on the bottom of the chat box. 2. The system asks "Type your message". 3. The user presses enter to send the typed message. 4. The system sends the message to each player and displays it in chat.	

3

Title:	Roll Die
Description:	Rolls a die that can roll 1 through 6.
System Under Design:	Squared System
Primary Actor:	User
Participants:	User (it's their roll)
Goal:	To get a random number 1 through 6 to find out how many cells the player can take or if the turn will be skipped.
Following Use Cases:	Includes "Show Possible moves"
Invariant:	Internet connectivity, API server is responding, Javascript is enabled, and It is the player's turn.
Precondition:	N/A
Success Postcondition:	Show Possible moves

STEPS:	ALTERNATIVES:
1. User presses the roll button. 2. Die Roll System will roll the die 3. Die Roll System will display the number rolled 4. Die Roll System will pass on the roll	

4

Title:	Show Possible Moves
Description:	This feature tells the player all of the possible cells that they can reach with their dice roll.

System Under Design:	Squared System
Primary Actor:	User
Participants:	User (it's their valid moves)
Goal:	To highlight cells that can be moved to and block the cells that are illegal to move to.
Following Use Cases:	Included by "roll die"
Invariant:	Javascript is enabled (this is purely a client-side improvement)
Precondition:	die is rolled
Success Postcondition:	N/A

STEPS:	ALTERNATIVES:
1.System takes the number returned from the die roll and makes sure it isn't a 5 or 6. 2.System highlights all cells that can be reached by that number of moves. 3.System blocks access to every other cell 4.Player chooses the path they want to take.	5a.System sends an error message if the user chooses an invalid cell. 5b. System moves the user if the cell is valid

5

Title:	**Pick Position**
Description:	This allows the user to click a new position on the board
System Under Design:	Squared System
Primary Actor:	User whose turn it is
Participants:	User
Goal:	To give the player the freedom to choose any valid position they want to.
Following Use Cases:	Includes "Verify Position" Extended by "Display position error"
Invariant:	User is in game and it is users turn
Precondition:	N/A
Success Postcondition:	Position is verified

STEPS:	ALTERNATIVES:
1. User presses a position they want to move to 2. System records that position and sends it to get verified	3a. If the position is valid, allow the user to move there. 3b. If the position is invalid, don't allow the user to move there.

6

Title:	Verify Position
Description:	This will verify if the chosen position can be reached with the die roll received
System Under Design:	Squared System
Primary Actor:	User whose turn it is
Participants:	User
Goal:	To stop players from moving to positions that shouldn't be reachable with the die roll they got
Following Use Cases:	Included by "Pick Position"
Invariant:	Must be in a game and users turn
Precondition:	User Picked Position
Success Postcondition:	Display error is sent or the character is moved

STEPS:	ALTERNATIVES:
1. System receives position and die roll 2. System verifies if the position can be reached with the amount of moves the user has	3a. If position is valid: move character to the position 3b.If position is invalid: Send an error message and ask for another position.

7

Title:	Login
Description:	Login to a specific account

System Under Design:	Squared System
Primary Actor:	User
Participants:	User
Goal:	To successfully login to the desired account.
Following Use Cases:	Extended by "display login error" Includes "Verify credentials"
Invariant:	A stable internet connection User is not already logged in
Precondition:	N/A
Success Postcondition:	System verifies credentials

STEPS:	ALTERNATIVES:
1. User presses the login button. 2. The system displays boxes to enter username and password 3. User enters username and password and presses "login". 4. System logs user in and displays home screen	5. Username and password is incorrect and the System asks for username and password again.

8

Title:	**Display Position Error**
Description:	This will tell the user if the position they chose is out of reach based on the die roll
System Under Design:	Squared System
Primary Actor:	User in game
Participants:	User
Goal:	To inform the user that the move the tried to make is invalid and illegal
Following Use Cases:	Extends "Pick Position"
Invariant:	User must be in a game and on their turn
Precondition:	User picked a position outside the bounds of their dice roll
Success Postcondition:	N/A

STEPS:	ALTERNATIVES:
1. System deemed chosen position invaid 2. System displays an error message 3. User dismisses the message and picks another position	

9

Title:	**Verify Credentials**
Description:	This verifies the credentials entered in the cases of logging in and registering
System Under Design:	Squared System
Primary Actor:	User
Participants:	User
Goal:	To make sure two accounts aren't created with the same email and admit access to those who enter their credentials correctly
Following Use Cases:	Included by "Log in" and "register account"
Invariant:	Stable internet connection,
Precondition:	Log in attempt, register attempt
Success Postcondition:	admitted to your account

STEPS:	ALTERNATIVES:
1. User Logs in or Registers for an account 2.System takes entered credentials and compares them to a database	3a. If credentials are approved, user is admitted into their account 3b. If credentials aren't approved, the user is prompted to login/register again.

10

Title:	**Register Account**
Description:	This allows the user to register for an account
System Under Design:	Squared System

Primary Actor:	User
Participants:	User
Goal:	To get an account setup
Following Use Cases:	Includes "Verify Credentials"
Invariant:	Not logged in, connected to internet
Precondition:	N/A
Success Postcondition:	Credentials get verified

STEPS:	**ALTERNATIVES**:
1. User presses the register button 2. System prompts the users with an input for an email and password 3. User enters credentials 4. System Verifies Credentials	 5a. System displays an email is already linked to an account error message 5b. Account is created