| **Title:** | Functioning Board |
| --- | --- |
| **Description**: | The board the game is played on. This updates constantly for all players to represent the current game. |
| **System Under Design:** | Squared System |
| **Primary Actor**: | The end-user |
| **Participants**: | Other players |
| **Goal**: | The user is able to interact with the game board at will. They’re able to move their piece around along with seeing other players moving around. |
| **Following Use Cases**: | 1. Showing the player their possible moves. The extend relationship is that having the board component be functional would allow these to be computed 2. Adding movement animations. This is an extend relationship since being able to update the board would allow an animation to be provided while updating the position |
| **Invariant**: | A stable internet connection as well as other players to make their moves to advance the game |
| **Precondition**: | The user must be logged in and in a game |
| **Success Postcondition**: | N/A  The user is able to interact with the board, and the system will be able to implement the users’ interactions |

| **STEPS**:   1. The user clicks on the board to the space they want to move 2. Our react frontend sends a request to the Flask API to move to that location 3. The Flask API checks if the move is valid 4. If the Flask API says it’s valid, have the move reflect on the board 5. Repeat per player | **ALTERNATIVES**:  4. If the move is invalid, alert the player that it’s invalid and tell them to try a valid square |
| --- | --- |

| 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**: | Squared Player sending the message |
| **Participants**: | Squared Players receiving the message |
| **Goal**: | Communicating with the opponent and it will reach the other user in less than 5 seconds. |
| **Following Use Cases**: | Add rate limiting to the chat to prevent spamming, add a swear filter, block links, fuzzing (to ensure no code exploits can be used) |
| **Invariant**: | The user has a stable internet connection and has Javascript enabled |
| **Precondition**: | User must be in a lobby with another user |
| **Success Postcondition**: | All players in the game can instantly communicate with one another during a game |

| **STEPS**:  1. Click on the chat bar on the bottom of the chat box.  2. Type your message.  3. Press enter to send the typed message.  4. The system sends the message to each player and displays it in chat. | **ALTERNATIVES**: |
| --- | --- |

| Title: | Roll Die |
| --- | --- |
| **Description**: | Rolls a die that can roll 1 through 6. |
| **System Under Design:** | Squared System |
| **Primary Actor**: | Squared Player |
| **Participants**: | Squared Player (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**: | Play an animation when doing dice rolls |
| **Invariant**: | Internet connectivity, API server is responding, Javascript is enabled |
| **Precondition**: | It is the player’s turn. |
| **Success Postcondition**: | The player is able to see their rolls on their screen |

| **STEPS**:  1. Press the roll button.  2. Die Roll System will roll the die  3. Die Roll System will display the number rolled | **ALTERNATIVES**: |
| --- | --- |

| 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**: | Squared Players |
| **Participants**: | Squared Players (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**: | See other player’s potential squares to strategize |
| **Invariant**: | Javascript is enabled (this is purely a client-side improvement) |
| **Precondition**: | Player just rolled the die |
| **Success Postcondition**: | The player can see their potential moves |

| **STEPS**:  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. | **ALTERNATIVES**:  5.System sends an error message if the player chooses an invalid cell. |
| --- | --- |

| Title: | Create Lobby |
| --- | --- |
| **Description**: | This feature will create the lobby where the game will be played with friends. |
| **System Under Design:** | Squared System |
| **Primary Actor**: | Squared Player |
| **Participants**: | Squared players that want to start a lobby |
| **Goal**: | to create a lobby that connects all players to the same board |
| **Following Use Cases**: | Change game settings |
| **Invariant**: | A stable internet connection |
| **Precondition**: | User is not already connected to a lobby |
| **Success Postcondition**: | A lobby for other users to join will be created |

| **STEPS**:  1. Press create lobby button  2. System asks for lobby code  3. Player enters a code other players will use to join the lobby.  4. System creates a lobby  5. System places player into lobby | **ALTERNATIVES**: |
| --- | --- |

| Title: | Join Lobby |
| --- | --- |
| **Description**: | This feature will allow a user to enter a lobby code to join a specific lobby. |
| **System Under Design:** | Squared System |
| **Primary Actor**: | Squared Player |
| **Participants**: | Squared players that want to join a lobby |
| **Goal**: | To join the lobby of another player in order to play the game together. |
| **Following Use Cases**: | N/A |
| **Invariant**: | A stable internet connection as well as other players in a lobby to join |
| **Precondition**: | User is not already connected to a lobby |
| **Success Postcondition**: | User will be connected to the specified lobby |

| **STEPS**:  1. Player presses join lobby button  2. System prompts player with a box to type in a lobby code  3. Player enters a lobby code given to them by another player  4. System recognizes code  7. System connects Player to specified lobby | **ALTERNATIVES**:  5.System doesn’t recognize code as a valid lobby code  6.System asks for a code again |
| --- | --- |

| Title: | Login |
| --- | --- |
| **Description**: | Login to a specific account |
| **System Under Design:** | Squared System |
| **Primary Actor**: | Squared Players |
| **Participants**: | Squared Players |
| **Goal**: | To successfully login to the desired account. |
| **Following Use Cases**: | N/A |
| **Invariant**: | A stable internet connection |
| **Precondition**: | User is not already logged in |
| **Success Postcondition**: | System will sign in to the specified account |

| **STEPS**:  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 | **ALTERNATIVES**:  5. Username and password is incorrect and the System asks for username and password again. |
| --- | --- |

| Title: | Logout |
| --- | --- |
| **Description**: | Logout of account |
| **System Under Design:** | Squared System |
| **Primary Actor**: | Squared Players |
| **Participants**: | Squared Players |
| **Goal**: | To successfully logout of account. |
| **Following Use Cases**: | N/A |
| **Invariant**: | A stable internet connection |
| **Precondition**: | User is already logged in |
| **Success Postcondition**: | system will log out of the account logged in |

| **STEPS**:  1. User presses the logout button.  2. The system asks if the user is sure they want to log out.  3. User chooses yes to log out.  5. System logs user out | **ALTERNATIVES**:  4. User chooses no if they don’t want to log out. |
| --- | --- |

| Title: | Choose Color |
| --- | --- |
| **Description**: | Pick the color that you want to represent your character |
| **System Under Design:** | Squared System |
| **Primary Actor**: | Squared Players |
| **Participants**: | Squared Players |
| **Goal**: | To choose the color square that you want to play as |
| **Following Use Cases**: | error if color is picked already |
| **Invariant**: | Stable internet connection |
| **Precondition**: | User is in a lobby |
| **Success Postcondition**: | color of character is chosen |

| **STEPS**:  1. User presses customize button  2. System will display the customize menu  3. User chooses change color  4. System displays all character color  5. User picks the preferred color.  7. System saves chosen color | **ALTERNATIVES**:  6. system says color is already chosen and asks you again. |
| --- | --- |

| Title: | Choose Board Theme |
| --- | --- |
| **Description**: | Pick the theme that you want to represent your board |
| **System Under Design:** | Squared System |
| **Primary Actor**: | Squared Players |
| **Participants**: | Squared Players |
| **Goal**: | To choose the board theme that you want to play with |
| **Following Use Cases**: | N/A |
| **Invariant**: | Stable internet connection |
| **Precondition**: | User is logged in |
| **Success Postcondition**: | Board theme is changed |

| **STEPS**:  1. User presses the customize button  2. System will display the customize menu  3. User selects theme change  4. System displays all theme choices  5. user picks preferred theme  6. System saves theme | **ALTERNATIVES**: |
| --- | --- |

