Use Case Descriptions (GUI)

Use Case: Log out

Iteration: 1st iteration

Primary actor: User

Goal in context: The user logs out of the system.

Preconditions: The system is powered on, and the user is already logged in with an existing profile.

Trigger: The user clicks on the log out button.

Scenario:

1. The user clicks on the log out button.
2. The user is logged out of the system.

Post Conditions: The user is logged out.

Exceptions:

- The log out button is unresponsive
- The system encounters an error while initializing
    - Server issues

Priority: High –necessary for log-out.

When Available: Within 1 sprint (1st iteration)

Frequency of Use: At most once per system session.

Channel to actor: Interaction of a click of the log out button with their mouse

Secondary actor: N/A.

Channel to Secondary Actors: N/A

Open issues:

- The setup of the game (any information required) is finished before GUI implementation

Use Case: Manage Profile

Iteration: 2nd iteration

Primary actor: User

Goal in context: The user is given various options for changing their profile.

Preconditions: The game board was set up and is ready for user input.

Trigger: The user has clicked a tile to place their piece.

Scenario:

1. The user is viewing their own profile.
2. The user clicks on the manage profile button.
3. The user is shown options for changing different aspects of their profile.

Post Conditions: The user can change different aspects of their profile.

Exceptions:

- The server is down

Priority: Medium-Low – not necessary for the functionality of the games supported by the system.

When Available: Within 2 sprints (2nd iteration)

Frequency of Use: Whenever the user wishes to change their profile.

Channel to actor: Via the system GUI

Secondary actor: N/A

Channel to Secondary Actors: N/A

Open issues: N/A

Use Case: Sign in

Iteration: 1st iteration

Primary actor: User

Goal in context: The user is signed into the system.

Preconditions: The system is powered on, and the user has an existing profile.

Trigger: The user enters a valid username and password combination.

Scenario:

1. The user enters their username.
2. The user enters their password.
3. The user clicks on the sign in button or presses enter.

Post Conditions: A request is made to connect the user to the server.

Exceptions:

- The server is down.

Priority: High – necessary for a user to interact with the system in general.

When Available: Within 1 sprint (1st iteration)

Frequency of Use: Once per user session.

Channel to actor: Via the GUI.

Secondary actor: N/A

Channel to Secondary Actors: N/A

Open issues: N/A

Use Case: Connect to Server

Iteration: 1st iteration

Primary actor: System

Goal in context: Connect the user to the server to enable other interactions.

Preconditions: The system is powered on, and the server is running.

Trigger: A user has attempted to sign in.

Scenario:

1. A user has attempted to sign in.
2. The system connects the user to the server.

Post Conditions: The user is successfully connected to the server.

Exceptions:

- The server is down.

Priority: High – necessary to enable system functionality for the user.

When Available: Within 1 sprint (1st iteration)

Frequency of Use: Once per user session

Channel to actor: Via the network and authentication systems.

Secondary actor: N/A

Channel to Secondary Actors: N/A

Open issues: N/A

Use Case: Create Profile

Iteration: 1st iteration

Primary actor: User

Goal in context: Create a profile for the user.

Preconditions: The system is powered on.

Trigger: The user clicks on the create profile button.

Scenario:

1. The user clicks on the create profile button.
2. The user enters a username.
3. The user enters a password.
4. The user confirms their entered username and password.
5. The newly created profile is stored in the system.

Post Conditions: The created profile is stored in the system database.

Exceptions:

- The system server is down.
- The entered username or password are invalid.

Priority: High – necessary for the user to interact with the system in general.

When Available: Within 1 sprint (iteration 1)

Frequency of Use: Whenever the user attempts to create a new profile.

Channel to actor: Via the system GUI.

Secondary actor: N/A

Channel to Secondary Actors: N/A

Open issues: N/A

Use Case: Show Game Menu

Iteration: 1st iteration

Primary actor: User

Goal in context: The user is shown the games supported by the system.

Preconditions: The system is powered on, and the user has signed in.

Trigger: The user clicks on the game menu.

Scenario:

1. The user clicks on the game menu.
2. The user sees the games supported by the system.

Post Conditions: The games supported by the system are shown to the user, and the user can view.

Exceptions:

- The server is down.
- The game menu is unresponsive.

Priority: High – necessary to allow users to choose which game to play.

When Available: Within 1 sprint (1st iteration)

Frequency of Use: Whenever the user attempts to view the games supported by the system.

Channel to actor: Via the GUI.

Secondary actor: N/A

Channel to Secondary Actors: N/A

Open issues: N/A

Use Case: Choose a Game

Iteration: 1st iteration

Primary actor: User

Goal in context: A game is selected so the user may queue up for a match.

Preconditions: The system is powered on, the user has signed in, and the user is on the game menu.

Trigger: The user clicks on one of the games in the game menu.

Scenario:

1. The user clicks on one of the games in the game menu.
2. The game is selected.
3. Game information and stats are shown and the user may queue for a match.

Post Conditions: The game menu showing the games supported by the system are displayed to the user.

Exceptions:

- The server is down.

Priority: High – necessary to allow users to queue up for matches and play games.

When Available: Within 1 sprint (1st iteration)

Frequency of Use: Whenever a user clicks on a game.

Channel to actor: Via the game menu.

Secondary actor: N/A

Channel to Secondary Actors: N/A

Open issues: N/A

Use Case: Choose Tic-Tac-Toe

Iteration: 1

Primary actor: User

Goal in context: The user chooses Tic-Tac-Toe from the game menu.

Preconditions: The system is powered on, the user is signed in, and the user is on the game menu.

Trigger: The user clicks on Tic-Tac-Toe from the game menu.

Scenario:

1. The user clicks on Tic-Tac-Toe from the game menu.
2. Tic-Tac-Toe information and stats are shown to the user, and the option for queueing for a Tic-Tac-Toe match is shown to the user.

Post Conditions: The user has selected Tic-Tac-Toe and is able to queue up for a Tic-Tac-Toe match.

Exceptions:

- The server is down.

Priority: High – necessary for the user to play Tic-Tac-Toe.

When Available: Within 1 sprint (1st iteration)

Frequency of Use: Whenever a user chooses Tic-Tac-Toe from the game menu.

Channel to actor: Via the game menu.

Secondary actor: N/A

Channel to Secondary Actors: N/A

Open issues: N/A

Use Case: Choose Connect4

Iteration: 1

Primary actor: User

Goal in context: The user chooses Tic-Tac-Toe from the game menu.

Preconditions: The system is powered on, the user is signed in, and the user is on the game menu.

Trigger: The user clicks on Connect4 from the game menu.

Scenario:

1. The user clicks on Connect4 from the game menu.
2. Connect4 information and stats are shown to the user, and the option for queueing for a Connect4 match is shown to the user.

Post Conditions: The user has selected Connect4 and is able to queue up for a Connect4 match.

Exceptions:

- The server is down.

Priority: High – necessary to allow users to play Connect4.

When Available: Within 1 sprint (1st iteration)

Frequency of Use: Whenever the user

Channel to actor: Via the game menu

Secondary actor: N/A

Channel to Secondary Actors: N/A

Open issues: N/A

Use Case: Queue Match

Iteration: 1

Primary actor: User

Goal in context: The user is queued up for a match.

Preconditions: The system is powered on, the user has signed in, and the user has chosen a game.

Trigger: The user clicks on the queue match button.

Scenario:

1. The user clicks on the queue match button.
2. The system searches for another user queued up for the same game.

Post Conditions: The user is queued up for a match.

Exceptions:

- The server is down

Priority: High – necessary to allow users to play matches with other users.

When Available: Within 1 sprint (1st iteration)

Frequency of Use: At least once per match.

Channel to actor: Via the GUI

Secondary actor: N/A

Channel to Secondary Actors: N/A

Open issues: N/A

Use Case: Start Match

Iteration: 1

Primary actor: System

Goal in context: Start a match between two users queued for the same game.

Preconditions: The system is powered on, the servers are running, and at least two users have signed in and are queued for a match on the same game.

Trigger: At least two users are queued up for a match for the same game.

Scenario:

1. At least two users are queued for a match for the same game.
2. The system searches for users queued for the same game.
3. The system matches two users queued for the same game.
4. The system starts a match between two users queued for the same game.

Post Conditions: A match is started between two users queued for the same game.

Exceptions:

- The server is down.

Priority: High – necessary for users to play games with each other.

When Available: Within 1 sprint (1st iteration)

Frequency of Use: Once per match.

Channel to actor: Via the network system.

Secondary actor: N/A

Channel to Secondary Actors: N/A

Open issues: N/A

Use Case: Search Profile

Iteration: 1

Primary Actor: User

Goal in context: Allows user to search for a profile within the system they want to either view or challenge to a match.

Preconditions: The user has an active session in the main interface where they can access the search function.

Trigger: User initiates a search to find another player profile they would like to view or challenge to a match.

Scenario:

1. User is in the main interface ready to search for another player
2. User enters the name or identifier of the player
3. Search gets processed and retrieves player profile options
4. User selects a player profile from their search results where they can choose to view profile or challenge player

Post-conditions: User lands on selected player profile successfully where they can view or challenge player

Exceptions:

- Unable to find player profile based on search results
- Desired player profile is unavailable
- Unable to search for players if search field is unresponsive

Priority: High. Enhances user experience through being able to interact with other players to challenge and/or view player information.

When available: Within 1st iteration.

Frequency of use: When user wants to search for other players on the platform

Channel to actor: Interaction through search query interface.

Secondary actors: N/A.

Channel to secondary actors: N/A.

Open issues: N/A.

Use Case: Challenge Player

Iteration: 1

Primary Actor: User

Goal in context: Allows user to challenge another player to a match.

Preconditions: The user has an active session in the main interface where they can access the search function, has identified a player, and is ready to initiate challenge to a game.

Trigger: User initiates challenge after selecting a player profile from their search results.

Scenario:

1. User has selected player to challenge based on search results
2. User initiates challenge to selected player

Postconditions: Challenge request has been sent to selected player

Exceptions:

1. Option to challenge player is unresponsive
2. System fails to send challenge request to desired player

Priority: High. Allows multiplayer gameplay to function with competitive play.

When available: Within 1st iteration.

Frequency of use: Whenever user wants to challenge another player to a match per game session.

Channel to actor: Interaction through challenge player option after landing on desired opponent.

Secondary actors: N/A.

Channel to secondary actors: N/A.

Open issues: N/A.

Use Case: View Profile

Iteration: 1

Primary Actor: User

Goal in context: Allows user to view selected player profile.

Preconditions: The user has an active session in the main interface where they can access the search function and has identified a player they want to view.

Trigger: User selects a player profile from search results they want to view details of.

Scenario:

1. User has selected a player from search results
2. User initiates profile overview for desired player profile

Postconditions: User is able to view selected player's profile successfully where they can access match history, wins/losses, etc…

Exceptions:

1. Unable to find player profile based on search results
2. System fails to retrieve profile details regarding desired player profile

Priority: High. Enhances user experience through being able to view other user statistics.

When available: Within 1st iteration.

Frequency of use: Whenever user wants to view other player profiles.

Channel to actor: Interaction through view player profile option upon selecting player they want to view.

Secondary actors: N/A.

Channel to secondary actors: N/A.

Open issues: N/A.

Use Case: Exit Application

Iteration: 1

Primary Actor: User

Goal in context: Allows user to exit out of the main application.

Preconditions: The user is in the main interface and wishes to exit out of the application.

Trigger: User presses the “Exit” button to exit session.

Scenario:

1. User is in the main menu interface
2. User selects the “Exit” button from the application menu to quit session

Postconditions: Application is closed and the user is out of the application.

Exceptions:

1. “Exit” button fails to exit user out and their application remains open

Priority: High. Users should be able to exit out of the application whenever they desire.

When available: Within 1st iteration.

Frequency of use: Once per active user session.

Channel to actor: “Exit” button in the main menu interface.

Secondary actors: N/A.

Channel to secondary actors: N/A.

Open issues: N/A.

Use Case: Leaderboard Button

Iteration: 1

Primary Actor: User

Goal in context: Allows user to view the leadership board for various games.

Preconditions: The user is in the main interface and wants to view the leadership board for various games.

Trigger: User presses the “Leaderboard” button

Scenario:

1. User is in the main menu interface
2. User selects the “Leaderboard” button from the menu

Postconditions: Display of leaderboards for Tic-Tac-Toe, Checkers, and Connect4 are available.

Exceptions:

1. “Leadership” button is unresponsive
2. Leaderboard button fails to retrieve data for leadership statistics of various games

Priority: High. Access to leaderboard is crucial for user engagement and competitive play.

When available: Within 1st iteration.

Frequency of use: Whenever user wants to check out leaderboard options.

Channel to actor: “Leaderboard” button in main menu interface.

Secondary actors: N/A.

Channel to secondary actors: N/A.

Open issues: N/A.

Use Case: Connect4 Leaderboard

Iteration: 1

Primary Actor: User

Goal in context: Allows user to view the leadership board for Connect 4.

Preconditions: The user is in the main interface and wants to view the leadership board for Connect 4.

Trigger: User presses the “Connect4 Leaderboard” button

Scenario:

1. User is in the main menu interface
3. User selects the “Leaderboard” button from the menu
2. User selects the “Connect4 Leaderboard” button

Postconditions: Display of leaderboard for Connect 4 made available for user.

Exceptions:

1. “Connect4 Leadership” button is unresponsive
2. System fails to retrieve information regarding Connect4 leaderboard data

Priority: High. Access to leaderboard is crucial for user engagement and competitive play.

When available: Within 1st iteration.

Frequency of use: Whenever user wants to view leadership information for Connect4

Channel to actor: “Connect4 Leaderboard” button after selecting “Leaderboard” option

Secondary actors: N/A.

Channel to secondary actors: N/A.

Open issues: N/A.

Use Case: TicTacToe Leaderboard

Iteration: 1

Primary Actor: User

Goal in context: Allows user to view the leadership board for Tic-Tac-Toe game.

Preconditions: The user is in the main interface and wants to view the leadership board for Tic-Tac-Toe.

Trigger: User presses the “Tic-Tac-Toe Leaderboard” button

Scenario:

1. User is in the main menu interface
2. User selects the “Leaderboard” button from the menu
3. User selects the “Tic-Tac-Toe Leaderboard” button

Postconditions: Display of leaderboard for Tic-Tac-Toe made available for user.

Exceptions:

1. “Tic-Tac-Toe Leadership” button is unresponsive
2. System fails to retrieve information regarding Tic-Tac-Toe leaderboard data

Priority: High. Access to leaderboard is crucial for user engagement and competitive play.

When available: Within 1st iteration.

Frequency of use: Whenever user wants to view leadership information for Tic-Tac-Toe

Channel to actor: “Tic-Tac-Toe Leaderboard” button after selecting “Leaderboard” option

Secondary actors: N/A.

Channel to secondary actors: N/A.

Open issues: N/A.

Use Case: Checkers Leaderboard

Iteration: 1

Primary Actor: User

Goal in context: Allows user to view the leadership board for Checkers game.

Preconditions: The user is in the main interface and wants to view the leadership board for Checkers.

Trigger: User presses the “Checkers Leaderboard” button

Scenario:

1. User is in the main menu interface
2. User selects the “Leaderboard” button from the menu
3. User selects the “Checkers Leaderboard” button

Postconditions: Display of leaderboard for Checkers made available for user.

Exceptions:

1. “Checkers Leaderboard” button is unresponsive
2. System fails to retrieve information regarding Checkers leaderboard data

Priority: High. Access to leaderboard is crucial for user engagement and competitive play.

When available: Within 1st iteration.

Frequency of use: Whenever user wants to view leadership information for Checkers.

Channel to actor: “Checkers Leaderboard” button after selecting “Leaderboard” option

Secondary actors: N/A.

Channel to secondary actors: N/A.

Open issues: N/A.

Use Case: Choose Checkers

Iteration: 1

Primary Actor: User

Goal in context: Allows user to play Checkers game if that is the game they want to play.

Preconditions: The user is in the main interface, they are on the interface to choose a game, Checkers is available to play for the user.

Trigger: User presses the “Choose Checkers” button

Scenario:

1. User is in the game menu
2. User selects the “Choose Checkers” option from the list of available games to play

Postconditions: Display of Checkers grid will be available for user to play.

Exceptions:

1. Checkers game fails to load due to technical issues
2. Interface is unresponsive when Checkers is selected

Priority: High. Users should be able to choose Checkers as an available game option.

When available: Within 1st iteration.

Frequency of use: Whenever user wants to play Checkers.

Channel to actor: “Choose Checkers” option as one of the games available for the user to select in the game menu.

Secondary actors: N/A.

Channel to secondary actors: N/A.

Open issues: N/A.

Use Case: Setup Game

Iteration: 1

Primary Actor: User

Goal in context: Sets up game necessary for user to engage in.

Preconditions: The user has selected the game they want to play from the options available and has started a match

Trigger: User has an active game match session.

Scenario:

1. User has started a game session after matchmaking finished
2. The interface displays and loads the game environment
3. User is presented with the game board they chose and resources to play

Postconditions: Game is fully set-up for user to engage in.

Exceptions:

1. Game fails to set-up properly through configuration errors
2. Interface for game mode is dysfunctional

Priority: High. Game needs to be set-up in order for gameplay to begin.

When available: Within 1st iteration.

Frequency of use: Once per active game session

Channel to actor: Through the selective game environment of their preferred game.

Secondary actors: N/A.

Channel to secondary actors: N/A.

Open issues: N/A.

Use Case: Send Message in Chat

Iteration: 1

Primary Actor: User

Goal in context: Allows communication between users in-game.

Preconditions: User has entered the game environment they have chosen after starting a match and game has been set-up successfully with the chat feature.

Trigger: User enters a message into the chat field and sends it to the entire game room.

Scenario:

1. User has entered the game session after matchmaking is complete
2. The game environment is set-up including the chat feature
3. User types a message into the chat field
4. User presses “Send” button
5. The message is displayed for all players to see inside the game session

Exceptions:

1. Chat feature fails to display from configuration issues
2. Chat feature is unresponsive preventing messages to be sent
3. Messages are not visible to everyone

Postconditions: Message is sent and visible to everyone engaged in the space

Priority: Medium. Non-essential but allows in-game communication between players.

When available: Within 1st iteration.

Frequency of use: Multiple times within one game session.

Channel to actor: Chat field within chosen game session.

Secondary actors: N/A.

Channel to secondary actors: N/A.

Open issues: N/A.

Use Case: Player Move

Iteration: 1

Primary Actor: User

Goal in context: Allows player to take their turn

Preconditions: Game environment has been properly set-up with players in session

Trigger: User initiates a move when it is their turn

Scenario:

1. User has entered the game session after matchmaking is complete
2. The game environment is set-up properly
3. It has indicated that it is the user's turn
4. User makes their move (ex. moving a piece, selecting a grid space)

Exceptions:

1. Move failed to process
2. User turn-taking is dysfunctional
3. Interface is unresponsive

Postconditions: Player's move is processed and the game state is updated accordingly

Priority: High. Allows active gameplay to happen.

When available: Within 1st iteration.

Frequency of use: Once per turn iteration.

Channel to actor: Game environment interface where moves can be made

Secondary actors: N/A.

Channel to secondary actors: N/A.

Open issues: N/A.

Use Case: Show Win Alert

Iteration: 1

Primary Actor: User

Goal in context: Alerts user if a player has won the game (either user or opponent)

Preconditions: User is in game session and a win condition is met by either user or opponent

Trigger: Game detects that a player has met the win condition after a move

Scenario:

1. User is in-game session
2. Game is set-up
3. User makes move when it is their turn
4. Game logic processes the move to check for win condition
5. If win condition met, then a win alert is displayed and the match is ended

Exceptions:

1. Win condition fails to process
2. Win alert fails to display on screen

Postconditions: User is made aware of win condition via alert and the match is concluded

Priority: High. Should indicate to user if a win condition is met to end the match

When available: Within 1st iteration.

Frequency of use: Once per game session when win condition is detected

Channel to actor: A win alert is displayed on the screen for user to view

Secondary actors: N/A.

Channel to secondary actors: N/A.

Open issues: N/A.

Use Case: Show Loss Alert

Iteration: 1

Primary Actor: User

Goal in context: Alerts user if a player has lost the game (either user or opponent)

Preconditions: User is in game session and a loss condition is met by either user or opponent (the other player has won)

Trigger: Game detects that a player has met the win condition after a move

Scenario:

1. User is in-game session
2. Game is set-up
3. User makes move when it is their turn
4. Game logic processes the move to check for win condition
5. If win condition met for either player, then the other player receives a loss

Exceptions:

1. Win condition fails to process
2. Loss alert fails to display on screen

Postconditions: User is made aware of loss via alert and the match is concluded

Priority: High. Should indicate to user if a loss is met to end the match

When available: Within 1st iteration.

Frequency of use: Once per game session when win condition is detected

Channel to actor: A loss alert is displayed on the screen for user to view

Secondary actors: N/A.

Channel to secondary actors: N/A.

Open issues: N/A.

Use Case: Match Ends

Iteration: 1

Primary Actor: User

Goal in Context: The user is notified that the match has concluded after a win or loss condition is met.

Preconditions: The user is engaged in a game session, and a win/loss condition has been achieved by either player.

Trigger: The system detects a win/loss condition has been met by a player.

Scenario:

1. The user is in a game session.
2. The game environment is set up.
3. Game logic processes the latest move to check for a win/loss condition.
4. The system displays the appropriate alert indicating match conclusion.

Postconditions: The match is concluded, and the user is presented with options to either play again or return to the game menu.

Exceptions:

1. The win/loss condition fails to process.
2. The match end alert fails to display.

Priority: High. Essential for concluding the match and providing next actions to the user.

When Available: Within 1st iteration.

Frequency of Use: Once per game session when a win/loss condition is detected.

Channel to Actor: Match conclusion is displayed on the user's screen.

Secondary Actors: N/A.

Channel to Secondary Actors: N/A.

Open Issues: N/A.

Use Case: Play Again Button

Iteration: 1

Primary Actor: User

Goal in Context: The user can initiate a new match immediately after a game ends without returning to the main menu.

Preconditions: The previous match has concluded, and a win/loss condition was met.

Trigger: The user selects the "Play Again" button after the match ends.

Scenario:

1. The system detects the match conclusion after a win/loss condition.
2. The match concludes, and the option to play again is shown to the user.
3. The user clicks the "Play Again" button.

Postconditions: The user is taken to the game setup screen, ready to initiate a new match.

Exceptions:

1. "Play Again" button is unresponsive.
2. The game environment fails to reset for a new match.

Priority: High. Provides an easy way to continue playing without navigating back to the main menu.

When Available: Within 1st iteration.

Frequency of Use: Once per game session after a match concludes.

Channel to Actor: Via the "Play Again" button in the interface.

Secondary Actors: N/A.

Channel to Secondary Actors: N/A.

Open Issues: N/A.

Use Case: Return to Game Menu

Iteration: 1

Primary Actor: User

Goal in Context: The user can return to the game menu after a match concludes to select other options.

Preconditions: A match has ended with a win/loss condition achieved.

Trigger: The user selects the "Return to Game Menu" option after the match ends.

Scenario:

1. The system processes the match conclusion.
2. The system displays the win/loss result to the user.
3. The user chooses "Return to Game Menu."

Postconditions: The user is navigated back to the main game menu.

Exceptions:

1. The "Return to Game Menu" button is unresponsive.
2. Failure to load the main menu.

Priority: High. Enables the user to exit the game and access other features.

When Available: Within 1st iteration.

Frequency of Use: Once per game session after a match concludes.

Channel to Actor: "Return to Game Menu" button displayed after match conclusion.

Secondary Actors: N/A.

Channel to Secondary Actors: N/A.

Open Issues: N/A.