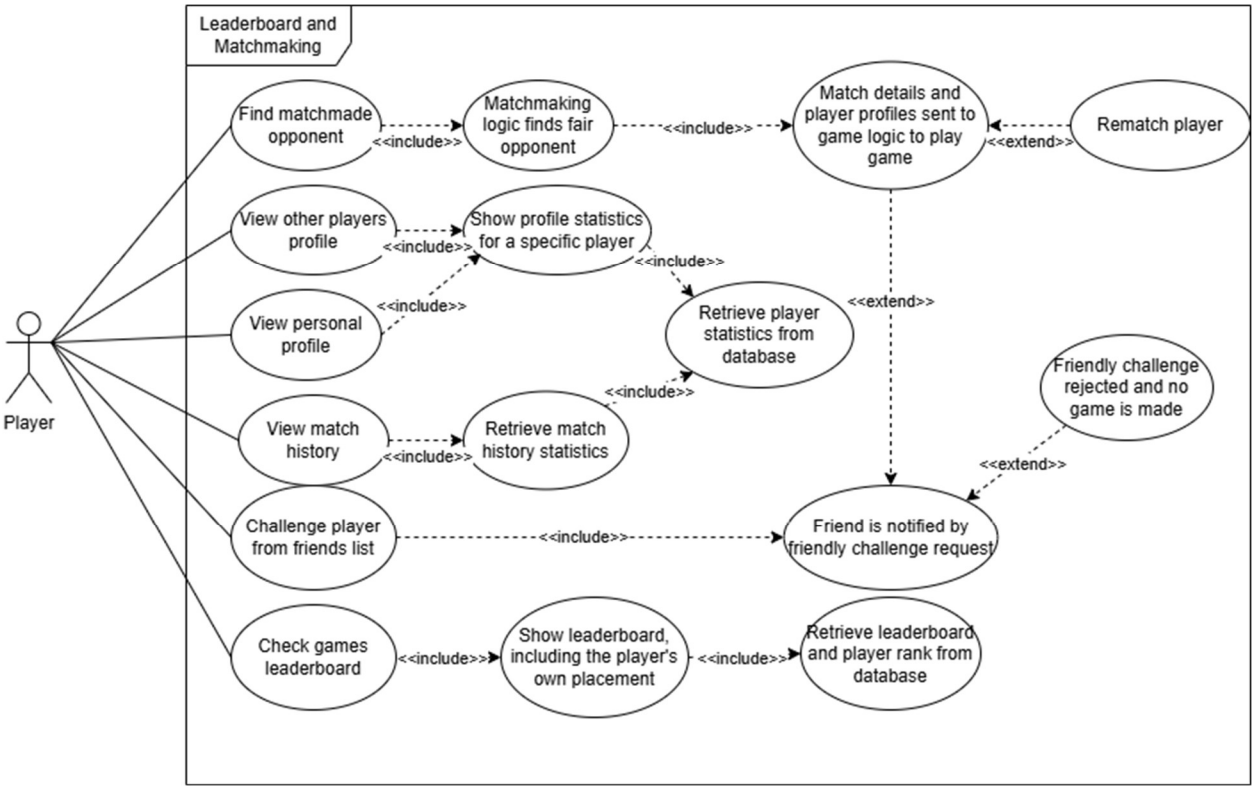

Use Case: View Other Players Profile/Statistics
Iteration: 1
Primary Actor: Game Player
Goal in Context: Allow a player to view detailed information about other players' profiles, game statistics, and performance metrics.
Preconditions:

- The system is accessible and online.
- Player is logged into the platform.
- The player whose profile is being viewed has a public profile or appropriate privacy settings.

Trigger: Player selects another player's username from a leaderboard, match history, or search results.
Scenario:

1. Player finds another player through leaderboard, search function, or match history.
2. Player selects/clicks on the other player's username.
3. System retrieves the selected player's profile data and statistics from database.
4. System displays the profile page showing the player's: a. Username and profile information b. Game-specific statistics (wins, losses, draws) c. Ranking and rating across different games d. Recent match history e. Achievements and badges
5. Player can toggle between different statistical views (by game type, by time period).
6. Player can select the option to challenge this player to a game if desired.

Post conditions:

- Player has viewed the requested profile and statistics.

Exceptions:

1. Private Profile: If the selected player has set their profile to private, system displays limited information or a privacy notice.
2. Data Unavailable: If player has no match history or statistics, system displays appropriate message indicating no data available.
3. System Error: If database connection fails, system displays error message and suggests trying again later.

Priority: Medium – Important for social and competitive aspects of the platform.
When Available: First increment.
Frequency of Use: High – Players will regularly check other players' statistics for competitive analysis.
Channel to Actor: Click of a button using touchscreen or mouse click.
Secondary Actors: Database System
Channel to Secondary Actors: Data retrieval interface.
Open Issues:

- Should players be able to hide certain statistics from public view?

Use Case: Challenge Player from Friends List
Iteration: 1
Primary Actor: Game Player
Goal in Context: Allow a player to directly challenge a friend to a game without affecting either player's ranking.
Preconditions:

- The system is accessible and online.
- Player is logged into the platform.
- Player has at least one friend added to their friends list.
- The friend is currently online and available. Trigger: Player selects a friend from their friends list and clicks "Challenge to Game".

Scenario:

1. Player navigates to their friends list in the platform.
2. System displays all online friends with their current status (available, in-game, busy).
3. Player selects a specific friend who is marked as available.
4. Player chooses "Challenge to Game" option.
5. System presents game selection menu with available board games.
6. Player selects desired game type (e.g., Chess, Connect Four).
7. Player selects "Friendly Match" option to indicate the game won't affect rankings.
8. Player confirms and sends challenge request.
9. System delivers notification to friend.
10. Friend receives and accepts challenge.
11. System initializes the game session between both players.

Post conditions:

- Both players are placed in the same game session with selected parameters.

- Match is recorded in the system as a "Friendly Match" that doesn't affect player rankings.
- Players' match history is updated to include the friendly match.

Exceptions:

1. Friend Unavailable: If the selected friend changes status to unavailable before challenge is sent, system notifies player and prevents challenge.
2. Challenge Declined: If friend declines challenge, system notifies challenger and returns to friends list.
3. Challenge Timeout: If friend doesn't respond within 2 minutes, challenge expires and system notifies challenger.

Priority: Medium – Important for social aspects but not critical to core gameplay.
When Available: First increment.
Frequency of Use: Moderate – Will be used regularly by players who prefer playing with friends.
Channel to Actor: Click of a button using touchscreen or mouse click.
Secondary Actors: Friend (Challenged Player)
Channel to Secondary Actors: Game client interface, notification system.
Open Issues:

1. Should players be able to spectate friendly matches between other players?
2. Should there be an option to convert a friendly match into a ranked match if both players agree?
3. Should friendly matches have different rules or settings available compared to ranked matches?

Use Case: Find Matchmade Opponent
Iteration: 1
Primary Actor: Game Player
Goal in Context: Enable a player to be matched against another player of similar skill level for a selected game.
Preconditions:

- The system is accessible and online.
- Player is logged into the platform.
- Player has completed enough games to have an established skill rating.

Trigger: Player chooses a game type and selects 'Find Opponent'.
Scenario:

1. Player chooses a game type and selects 'Find Opponent'.
2. System retrieves player's current skill rating and game history from database.
3. System searches for other online players who are also seeking matches for the same game.
4. System identifies players within an appropriate skill range (±100 rating points).
5. System selects the best match based on skill similarity, connection quality, and wait time.
6. System notifies both players that a match has been found.
7. Both players confirm readiness within 30 seconds.

8. System initializes the game session with both players.
9. Game begins with standard rules and settings.

Post conditions:

- Both players are placed in the same game session.
- Match is recorded in the system for future skill calculations.
- Players' status is changed to "In Game".

Exceptions:

1. No Suitable Opponent: If no suitable opponents are found within 45 seconds, system expands search criteria to include wider skill range.
2. Search Cancellation: If player cancels search, system removes player from matchmaking queue.
3. Readiness Confirmation Failure: If matched player doesn't confirm readiness, system cancels match setup and searches again.

Priority: High – Essential for the matchmaking functionality of the platform.
When Available: First increment.
Frequency of Use: Very frequent – Will be used constantly as players seek matches.
Channel to Actor: Click of a button using touchscreen or mouse click.
Secondary Actors: Matchmaking System, Other Players
Channel to Secondary Actors: Game client interface, network communication.
Open Issues:

1. How quickly should the skill range expand if matches aren't found?
2. Should players be able to set preferences for opponents beyond skill level?
3. How to handle disconnections during the matchmaking process?

Use case: Player can view Game Leaderboard
Iteration: 1
Primary actor: Player
Goal in context: To allow the player check their regional/global ranking and able to see their rank with other players
Preconditions:

- The player has an active and verified account
- The player has played all of their placement game(s) and received a ranking
- The leaderboard system is active and accessible

Trigger: The player select the leaderboard button from the game menu
Scenario:

1. The player selects the Leaderboard option from the game menu.
2. The player can select the filter to display the local regional leaderboard or the global leaderboard
3. The system retrieves and displays the top 100 ranked players based on Elo/MMR.
4. The player sees their current rank, statistics and match history.

Exceptions:

- If the leaderboard fails to load, the system displays an error message.

- If the player hasn’t played any ranked matches, the system will display an unranked status

Priority: High
When available: First increment
Frequency of use: Frequent as players like to check their ranking
Channel to actor: When leaderboard button in menu is selected
Secondary actors: Server database (fetching and updating leaderboard)
Channel to secondary actors: API request to server database
Open issues: None

Use case: Player viewing their own statistics
Iteration: 1
Primary actor: Player
Goal in context: To allow the player to view their personal profile, including their game statistics, ranking, match history, and highest ranked achieved
Preconditions:

- The player has an active and verified account
- The player has played all of their placement game(s) and received a ranking

Trigger: The player select their profile from the game menu
Scenario:

1. The player clicks on their profile from the game menu
2. The system retrieves the player's stored profile data from the database.
3. The system displays key information, including:
   - Username and profile picture
   - Current ranking and rank title
   - W/L ratio
   - Recent match history
   - Total games played
4. The player clicks on any one of the options above and each options will display its own features (i.e. when clicked on recent match history it will show the player and opponent name and ranking, results, match duration)

Exceptions:

- If there is a server issue or missing data, the system displays an error message and prompts the player to retry
- If the player hasn’t played any ranked matches but has played casual matches, the system will display an unranked status and statistics for casual matches will be provided and vice versa

Priority: High
When available: First increment
Frequency of use: Frequent as players like to check their profile as they want to check for improvements
Channel to actor: When profile in game menu is selected
Secondary actors: Server database (fetching and updating profile data)

Channel to secondary actors: API request to server database
Open issues: None

Use case: Rematch Request
Iteration: 1, last modification: Mar 7 by Min Oh
Primary actor: Player
Goal in context: To allow the player to send a rematch request to their most recent opponent
Preconditions:

- The player has an active and verified account
- The player just completed a match
- The opponent is still online and agrees to the rematch

Trigger: The player selects “Request Rematch” after game ends
Scenario:

1. After finishing a game the request rematch prompt appear under the continue prompt
2. The player selects rematch request
3. The system sends a rematch invitation to the opponent.
4. The opponent can accept or decline.
5. If accepted, the system starts a new game

Exceptions:

- If the opponent is no longer online the system will prompt a message “Your opponent is no longer online.”
- If the opponent refused the rematch then the system will prompt a message “Rematch Declined”

Priority: High
When available: First increment
Frequency of use: Frequent as players like to check their profile as they want to check for improvements
Channel to actor: In game screen after a finished match
Secondary actors: Server database (fetching and updating info from most recent match)
Channel to secondary actors: API request to server database
Open issues: None