Use Case: Start Matchmaking
Primary Actor: Player
Goal in Context: To allow players to enter a matchmaking queue and find a suitable match
Preconditions:

- Player is logged on
- Internet Connection is stable
- Player has reached a rank high enough

Trigger: Player clicks “Start Game”
Scenario:

- Player navigates to the matchmaking menu
- Player selects game mode (Casual, Ranked, and etc)
- System searches opponents depending on factors such as rank, preference, region, and etc.
- System finds a match and displays confirmation
- System adds player into the lobby

Postconditions:

- Player is delivered into the game lobby with an opponent of a similar rank

Exception(s):

- No match is found for a player
- Network connection is unstable, system cannot proceed
- Player disconnects mid-process

Priority: High
When Available: Within 1 to 2 sprints
Channel to Actor:

- Through physical interaction using a touch screen on the players device or input devices (keyboard/mouse)
- Game UI

Secondary Actors: Player Database (To access ranks)
Channel to Actor: System Communication
Open Issues:

- If a player leaves a match mid process, do they suffer consequences?
- Should there be a confirmation before a player is placed into a game lobby with the opponent?

Use Case: Leave Matchmaking
Primary Actor: Player
Goal in Context: To allow players to leave a matchmaking process before being placed in a game lobby
Preconditions:

- Player is logged on
- Internet Connection is stable
- Payer has initiated the matchmaking process

Trigger: Player clicks “Cancel”
Scenario:

- Player has initiated the matchmaking process.
- **Case 1:** Player does not want to play the game anymore and does not want to suffer consequences of leaving mid game
- **Case 2:** Player does not have good network connection and risks disconnecting mid game

Postconditions:

- Player is removed from the matchmaking queue and delivered back to select game interface

Exceptions:

- System has already found a match for the player and does not let player cancel
- Player has weak network and cancel request is not sent fast enough

Priority: High
When available: 1 to 2 sprints
Channel to Actor:

- Through physical interaction using a touch screen on the players device or input devices (keyboard/mouse)
- Game UI

Secondary Actors: None
Channel to Actor: N/A
Open Issues:

- Should a timeout be a consequence of frequently cancelling matchmaking requests

Use Case: Match Confirmation
Primary Actor: System
Goal in Context: To display confirmation of match found
Preconditions:

- Player is logged on
- Internet Connection is stable
- Payer has initiated the matchmaking process

Trigger: System has found an appropriate match
Scenario:

- Player has initiated the matchmaking process.
- System finds an appropriate match and displays the opposition players name on the game interface

Postconditions:

- The system displays opposition players ranking and name

Exceptions:

- System does not find any appropriate match
- Player has weak network

Priority: High
When available: 1 to 2 sprints
Channel to Actor:

- Game UI

Secondary Actors: None
Channel to Actor: N/A
Open Issues:

- Should the system display a timeout message if there are no matches found

Use Case: Timeout
Primary Actor: Player
Goal in Context: Repercussion for players who have left ongoing matches
Preconditions:

- Player has left one less than minimum number of matches in progress

Trigger: Player leaves a minimum number of matches in progress
Scenario:

- Player has initiated the matchmaking process.
- System finds an opponent
- Match begins
- Player leaves ongoing match

Postconditions:

- System displays a message, showing the penalty for leaving an on-going match: A period of time for which the player will not be able to play any games

Exceptions:

- Player has weak network signal and the game disconnects

Priority: High
When available: 1 to 2 sprints
Channel to Actor:

- Game UI

Secondary Actors: None
Channel to Actor: N/A
Open Issues:

- Should the timeout period be increased if the player leaves matches over and over again
- Should the first time be a warning
- Should only ranked matches have this timeout consequence or even casual game mode have it too

Use Case: Finding Opponent
Primary Actor: System
Goal in Context: Pairing two similar ranking opponents
Preconditions:

- Player has a minimum rank

Trigger: Player clicks Start Game
Scenario:

- Player has initiated the matchmaking process.
- System goes through the queue list of players waiting to matched with an opponent in the same game
- System finds an opponent with a ranking within a range of the with the pairs ranking

Postconditions:

- System displays a “Match Found” confirmation

Exceptions:

- Player has weak network signal
- There are no players in queue

Priority: High
When available: 1 to 2 sprints
Channel to Actor:

- Game UI
- Leaderboard database and Queue data structure

Secondary Actors: Players
Channel to Actor: Game UI
Open Issues:

- What if the system cannot find an opponent player with ranking within the range? Should the algo. expand its range each time it fails?
- What if the player does not have a minimum rank

Use Case: Game Invite
Primary Actor: Player
Goal in Context: Sending game invite to friend
Preconditions:

- Player has opponent as friend

Trigger: Player clicks “Send Challenge”
Scenario:

- Player has good network connectivity
- Player’s friend is online and player sends an invite

Postconditions:

- System sends an invite to to opponent’s screen

Exceptions:

- Opponent player disconnects

Priority: Medium
When available: 2 to 3 sprints
Channel to Actor:

- Game UI

Secondary Actors: System
Channel to Actor: System Database and internal communication
Open Issues:

- Should there be an invite timeout if the opponent doesn’t respond?
- Should there be a notification system if the opponent is idle?

Use Case: Invite Confirmation
Primary Actor: Player
Goal in Context: Accepting Game Invite
Preconditions:

- Player has received an invite from a friend

Trigger: Player clicks "Accept"
Scenario:

- Player has good network connectivity
- Player receives invite
- Player accepts invite

Postconditions:

- System delivers both players into game room
- Game session is initialized

Exceptions:

- Either one of the player loses network connectivity during the process
- Player declines invite

Priority: Medium
When available: 1 to 2 sprints
Channel to Actor:

- Game UI

Secondary Actors: System
Channel to Actor: System database and Internal communication
Open Issues:

- Should there be a buffer period before sending another invite if the opponent doesn't respond?
- Should there be a notification system if the opponent is idle?

Use case: View Leaderboard

Iteration: 1

Primary Actor: Player

Goal in context: Allow players to view the ranking of top players based on performance metrics

Preconditions:

- Player must have access to the game interface
- Leaderboard data are available and updated

Trigger: Player navigates and selects the leaderboard section in game menu

Scenario:

1. Player selects the leaderboard option in game menu
2. System/server retrieves and updates the latest player rankings
3. GUI displays the list of top players sorted by rating (descending order by default)
4. Player can scroll through the leaderboard list
5. Player can search for specific users

Post conditions: Player can successfully view the leaderboard

Exceptions:

2a. No ranking data is available

2b. Network error preventing leaderboard data from being retrieved

Priority: High

When available: In 1 sprint, by end of project iteration 3

Frequency of use: Varies on player interest

Channel to actor:

- In game GUI
- Physical interaction using input devices such as keyboard or mouse

Secondary actors: Player database (for retrieving ranking data)

Channel to secondary actors: System communication

Open issues:

- Should there be additional filtering options for different ranking types (eg, game stats)?

Use case: Update Leaderboard Data

Iteration: 1

Primary Actor: Game backend

Goal in context: Ensure leaderboard displays most recent player ranking and/or performance data

Preconditions:

- Player completes a ranked game
- Ranked game results must be available

Trigger: A ranked game is completed

Scenario:

1. System receives match results (win/loss data, etc.)
2. System updates player ratings and rankings based on performance (remove rating score if player loses, grants rating score if player wins)
3. System sorts the updated list and stores it in database

Post conditions: Leaderboard data accurately reflects the most recent rankings

Exceptions:

1,2. Network/database failure – preventing updates

Priority: High

When available: 1 sprint, by end of project iteration 2

Frequency of use: After every ranked match completion

Channel to actor: N/A, System communication

Secondary actors: Player database

Channel to secondary actors: System communication

Open issues: Do we have to have the leaderboards updated after every match or could we update them less to reduce load?

Use case: Sort top players

Iteration: 1

Primary Actor: Game backend

Goal in context: Retrieve an ordered list of players based on ranking metrics

Preconditions:

- Leaderboard data is available

Trigger: Player requests top players

Scenario:

1. System receives a request for player rankings
2. System receives the highest ranked players from the leaderboard
3. System sorts players based on ranking criteria
4. System sends the sorted list to the requesting player

Post conditions: Request player rankings are displayed or used in another system process

Exceptions:

1,2. Database failure preventing retrieval

3. No data is available for processing

Priority: Medium/high

When available: 1 sprint, by end of project iteration 3

Frequency of use: Varies on player interest.

Channel to actor: System communication

Secondary actors: Player database

Channel to secondary actors: System communication

Open issues: What other sorting options do the leaderboard settings offer?

Use case: Sort leaderboard by different criteria

Iteration: 1

Primary Actor: Player

Goal in context: Allow player to sort the leaderboard statistics based on different ranking criteria (win/loss ratio, number of wins, rating, or etc.)

Preconditions:

- Leaderboard must be accessible
- Leaderboard has enough data for sorting

Trigger: Player changes a sorting option from the leaderboard menu

Scenario:

1. Player opens up leaderboards in game menu
2. Player selects a sorting option from available choices
    - Win/loss ratio
    - Number of wins
    - Rating points
3. System gets leaderboard data
4. System sorts leaderboard based on chosen criteria
5. Sends sorted leaderboard to GUI to display

Post conditions:

Exceptions:

3. Network error prevents leaderboard data from updating

4. No data for the selected sorting criteria

Priority: Low

When available: 1 sprint, by end of iteration 3

Frequency of use: Varies based on player interest

Channel to actor: Player database, system communication

Secondary actors: Player database

Channel to secondary actors: system communication

Open issues: Should system remember the last sorting preference for the player?

Use case: Toggle ascending/descending order on leaderboard

Iteration: 1

Primary Actor: Player

Goal in context: Allow players to change sorting order of the leaderboard to either ascending or descending.

Preconditions:

- Leaderboard is accessible
- Player is viewing leaderboard menu
- Sorting criterion is selected

Trigger: Player toggles the ascending/descending options

Scenario:

1. Player navigates to leaderboard
2. Player selects a sorting criterion (win/loss ratio, number of wins, etc.)
3. Player toggles the ascending/descending option
4. System updates leaderboard to reflect selected sorting order
5. Leaderboard displays players in specified sorting order

<u>**Post conditions:**</u> Leaderboard sorting order is updated and displayed

Exceptions:

4. Network prevents sorting order from updating

5. Not enough data for leaderboard to sort

Priority: Low

When available: 1 sprint, end of project iteration 3

Frequency of use: Varies on player interest

Channel to actor: GUI

Secondary actors: Player database

Channel to secondary actors: System communication

Open issues: Should system remember the last sorting preference for the player?

Use case: Search for specific player (including themselves) on the leaderboard

Iteration: 1

Primary Actor: Player

Goal in context: Allow player to search for specific player (by username or maybe other identifiers) to view their ranking

Preconditions:

- Leaderboard must be accessible
- Player is viewing leaderboard
- Searched player exists in the ranking database

Trigger: Player searches a username in leaderboard menu

Scenario:

1. Player navigates to leaderboard

2. Player searches for a username

3a. If username found, system removes all other players except the searched player

3b. If username not found, system displays a "player not found" error message

Post conditions: Player can view the searched player's ranking and statistics

Exceptions:

2.Searched player does not exist

3. Database error fails to retrieve player statistics

3,4. Network error prevents GUI and database from communicating

Priority: Medium

When available: 2 sprints, end of project iteration 3.

Frequency of use: Varies on player interest

Channel to actor: GUI

Secondary actors: Player database

Channel to secondary actors: System communication

Open issues: Should search feature support partial matches?

Use case: View friends on leaderboard

Iteration: 1

Primary Actor: Player

Goal in context: Allow players to filter the leaderboard to view only their friend's rankings

Preconditions:

- Player has access to leaderboard menu
- Leaderboard data and friends list must be available and updated

Trigger: Player selects the option to view only friends on leaderboard

Scenario:

1. Player navigates to leaderboard menu
2. Player selects "View Friends" filter option
3. System retrieves player's friends list
4. System filters the leaderboard to display rankings of the player's friends
5. Player can scroll and view the filtered list

Post conditions:

Exceptions:

3,4. Network error prevents leaderboard data from being retrieved

3. Network error prevents access to friend's list

Priority: Medium

When available: 1 sprint, by end of project iteration 3

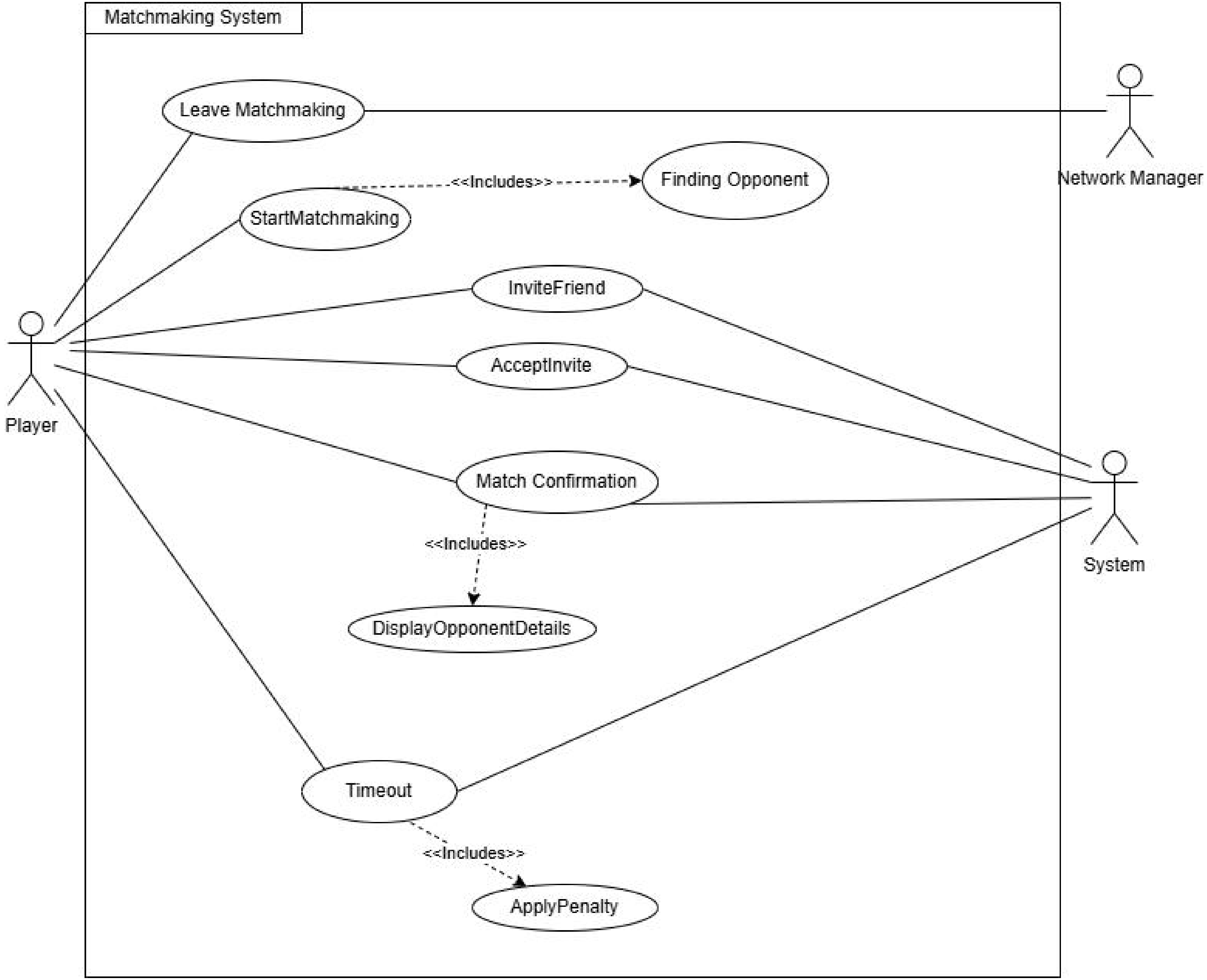
Frequency of use: Varies on player interest

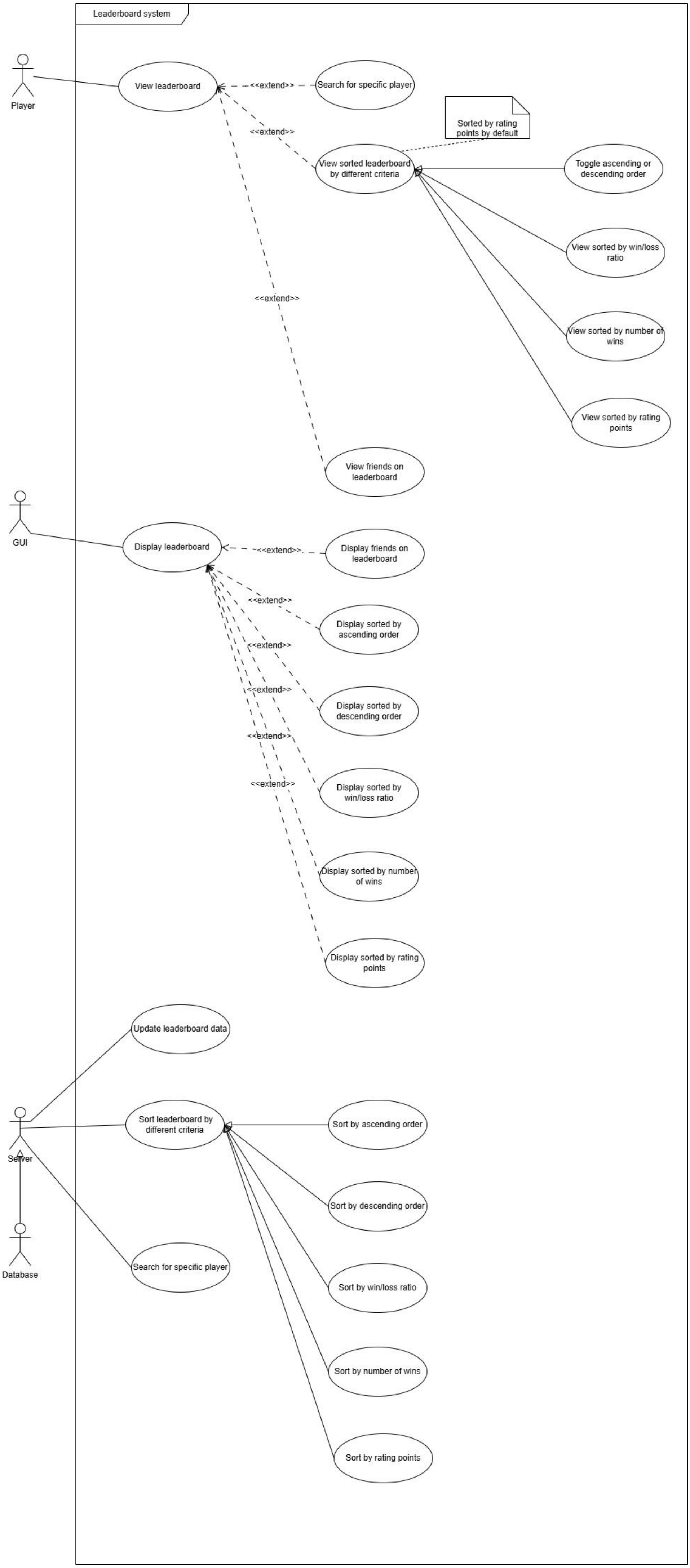
Channel to actor: GUI

Secondary actors: Player database

Channel to secondary actors: System communication

Open issues: Should players be able to sort the friend's leaderboard separately?


Matchmaking System
Leave Matchmaking
Network Manager
<<Includes>>
Finding Opponent
StartMatchmaking
InviteFriend
Player
AcceptInvite
Match Confirmation
System
<<Includes>>
DisplayOpponentDetails
Timeout
<<Includes>>
ApplyPenalty


Leaderboard system
Player
View leaderboard
<<extend>>
Search for specific player
<<extend>>
Sorted by rating points by default
View sorted leaderboard by different criteria
Toggle ascending or descending order
View sorted by win/loss ratio
View sorted by number of wins
View sorted by rating points
<<extend>>
View friends on leaderboard
GUI
Display leaderboard
<<extend>>.
Display friends on leaderboard
<<extend>>
Display sorted by ascending order
<<extend>>
Display sorted by descending order
<<extend>>
Display sorted by win/loss ratio
<<extend>>
Display sorted by number of wins
<<extend>>
Display sorted by rating points
Update leaderboard data
Server
Sort leaderboard by different criteria
Sort by ascending order
Sort by descending order
Sort by win/loss ratio
Sort by number of wins
Sort by rating points
Database
Search for specific player