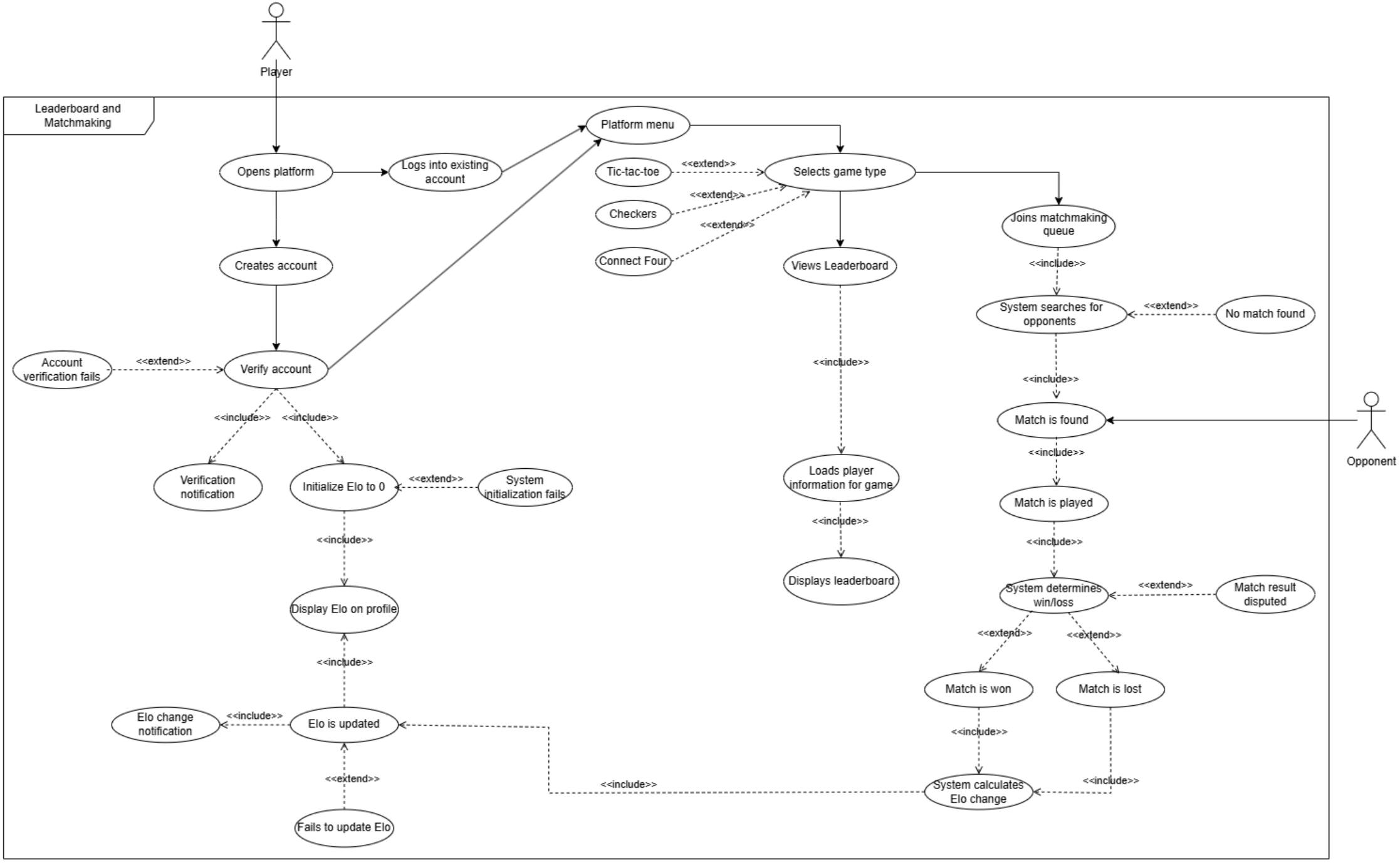
Player
Leaderboard and Matchmaking
Opens platform
Logs into existing account
Platform menu
Creates account
Tic-tac-toe
Checkers
Connect Four
<<extend>>
Selects game type
Views Leaderboard
Joins matchmaking queue
<<include>>
System searches for opponents
No match found
Account verification fails
Verify account
<<include>>
Match is found
Opponent
Verification notification
Initialize Elo to 0
System initialization fails
Loads player information for game
Displays leaderboard
Match is played
Display Elo on profile
System determines win/loss
Match result disputed
Match is won
Match is lost
Elo change notification
Elo is updated
Fails to update Elo
System calculates Elo change

Use Case: Initialize Player Elo

Iteration: 1

Primary Actor: Player

Goal In Context: Initialize the elo to 0 for all new player accounts

Preconditions:
- The player has successfully registered and verified their account

Trigger: User finishes creating account

Scenario:
1. The player completes the registration process.
2. The system verifies the player's account.
3. The system initializes the player's elo to 0.
4. The player is notified that their account setup is complete and the elo rating is displayed on the player's profile**.**

Post-conditions:
- Account is initialized with elo of 0
- Player receives confirmation of account setup and is ready to join matchmaking queue

Exceptions:
- Account verification fails
- System initialization fails

Priority: High, basic game function

When available: First increment or iteration

Frequency of use: Each time a new player registers

Channel to actor: Game interface

Secondary actors: N/A

Channel to secondary actors: N/A

Open issues:
- How are we going to handle duplicate accounts?
- How can we ensure elo initialization is tamper-proof?

Use Case: Matchmaking Queue

Iteration: 1

Primary Actor: Player

Goal In Context: Allow a player to join a matchmaking queue and be paired with an opponent of similar elo rating

Preconditions:
- The player has logged into the multiplayer platform
- The player has a valid player profile and elo rating

Trigger: The player selects the “Join Matchmaking Queue” option

Scenario:
1. The player selects the matchmaking option.
2. The system searches the matchmaking queue for opponents with similar elo ratings.
3. Once a match is found, the system notifies both players.
4. The game session is initialized.

Post-conditions:
- The player is paired with an opponent and starts a game

Exceptions:
- No match is found, and the player remains in the queue until an appropriate opponent joins

Priority: High, core game mechanic

When available: First increment or iteration

Frequency of use: Each time a player wants to play a match

Channel to actor: Game interface

Secondary actors: Opponent Player

Channel to secondary actors: Game interface, matchmaking system

Open issues:
- How long to keep players waiting in the queue if no suitable opponent is found?
- If no suitable opponent is found, how long till the player is assigned someone that is out of their elo rating range?

Use Case: Update Player Elo

Iteration: 1

Primary Actor: Player

Goal In Context: Update the player's elo based on the outcome of a match

Preconditions:
- A game has concluded, and the winner and loser are determined
- Both players have valid elo ratings

Trigger: Match ends and results are recorded

Scenario:
1. The game ends, and the system determines the winner and loser.
2. The system calculates the elo change based on the match outcome.
3. The player's elo is updated accordingly.
4. The player is notified of their change in elo, which is visible on their profiles.

Post-conditions:
- Player's elo rating is updated
- Player receives notification of the change in elo

Exceptions:
- Match result is disputed
- System fails to update the elo ratings

Priority: High, basic game function

When available: First increment or iteration

Frequency of use: After each match

Channel to actor: Game interface

Secondary actors: Opponent player

Channel to secondary actors: Game interface

Open issues:
- How can we ensure accurate and fair elo calculations?

Use Case: View Leaderboard

Iteration: 1

Primary Actor: Player

Goal In Context: View the leaderboard to see the top-ranked players based on elo rating for each game type

Preconditions:
- Players have played games and have valid elo ratings
- Leaderboard data is updated regularly

Trigger: User selects "View Leaderboard" from the menu

Scenario:
1. The user selects the "View Leaderboard" option.
2. The system retrieves the list of players sorted by their elo ratings, highest to lowest.
3. The leaderboard is displayed for the selected game type.
4. The user views the leaderboard rankings.

Post-conditions:
- The leaderboard is shown with updated player rankings

Exceptions:
- Leaderboard data fails to load

Priority: Medium, players are not required to view leaderboard

When available: First increment or iteration

Frequency of use: Regularly by players

Channel to actor: Game interface

Secondary actors: N/A

Channel to secondary actors: N/A

Open issues:
- How many players should the leaderboard display?