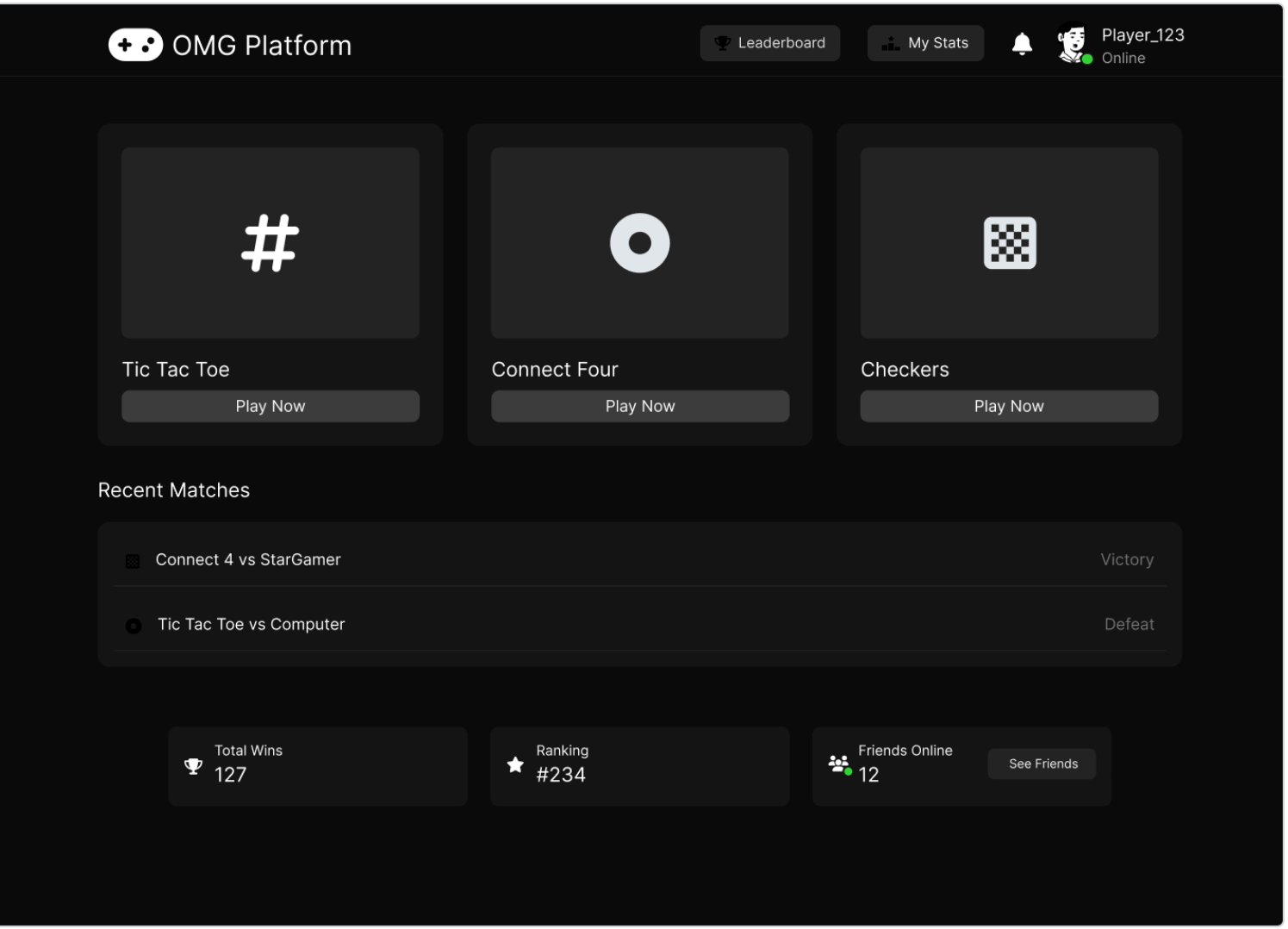
Leaderboard & Stats Integration Overview



1. **Main Dashboard Screen**

- Options:
  - "Play Now" → Notification prompt asking what kind of game the user wants to play→ leads to Matchmaking Lobby thereafter
  - Game Modes → choose between Tic-Tac-Toe, Checkers, Connect Four
  - "Leaderboard" → directs user to Leaderboard screen
  - "My Stats"/User Profile → directs user to their personal Stats
  - "Profile Settings" (via Avatar) → leads to profile settings or Log Out selection
  - **"Recent Matches" section** - displays the user's recent matches (perhaps last two recent matches?) as well as whether they have won or not
  - **Total Wins, Current Ranking,**
    - How does Ranking get determined? Could also display their current "ELO" instead?
    - Total Wins - total wins across all 3 board games

- Friends Online → can go to “See Friends’

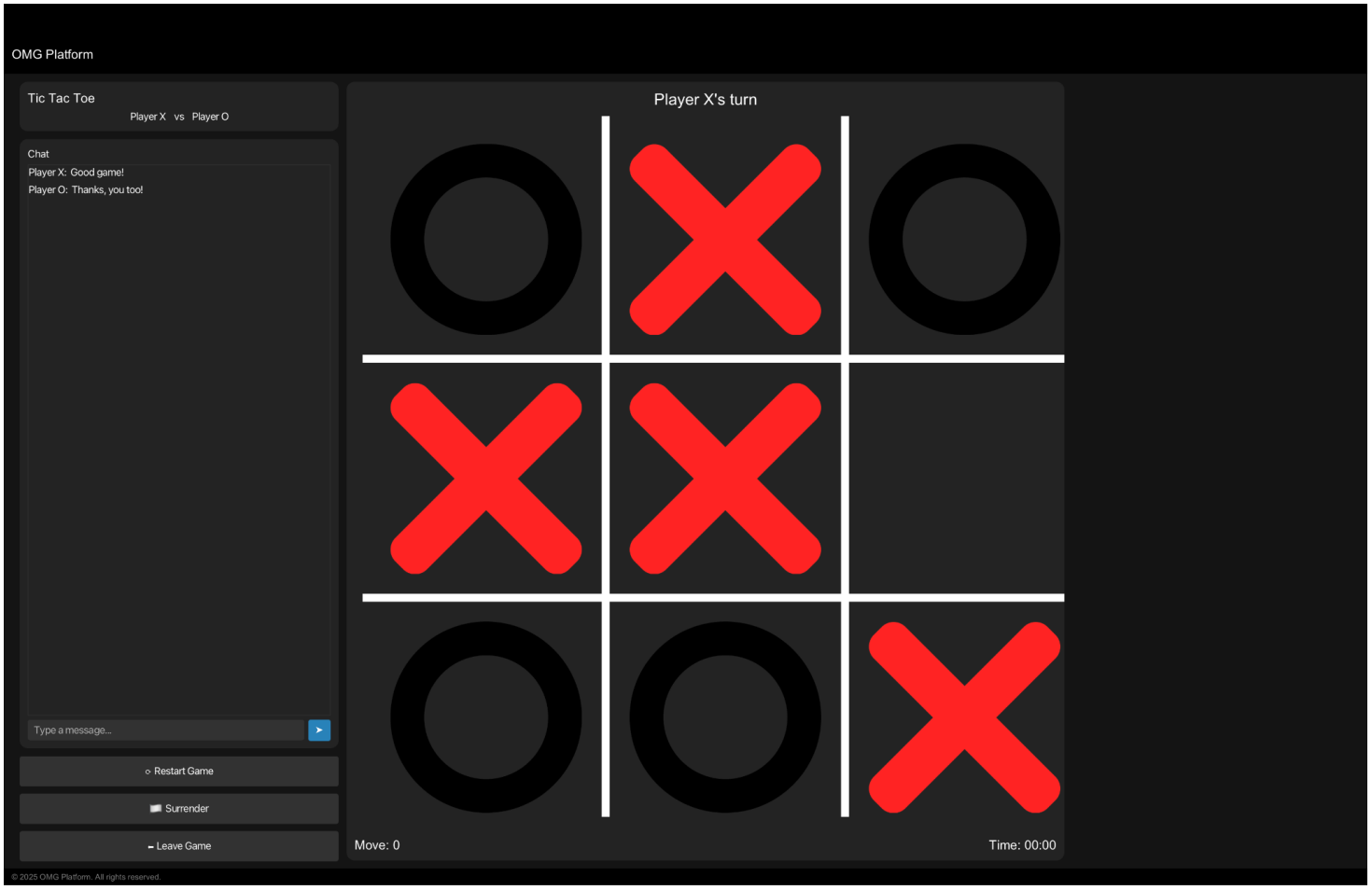
2. **Matchmaking Lobby**

- **GUI Responsibilities:**
  - Show current matchmaking status: "Searching for opponent..."
  - Display when a match is found and auto-transition to game screen.
  - Option to cancel matchmaking.
- **Leaderboard & Matchmaking Team Responsibilities:**
  - Handle backend matchmaking queue logic.
  - Push real-time match found event (via Networking team).
  - Provide opponent info and game type to GUI.
- **Expected Integration:**
  - GUI calls `joinQueue(gameType)` and receives a "match found" WebSocket push.
  - GUI transitions user into the appropriate game with assigned player slot:
    - *Example:* Supan and Anica are playing checkers. Supan is assigned Player 1, Anica is assigned Player 2
      - Matchmaking system sends back:

```
{
  "gameType": "Checkers",
  "playerSlot": 2,
  "opponent": "Supan",
  "gameMode": "ranked",          // or "casual"
  "playerUsername": "Anica",     // Optional: current user's name
  "matchId": "match_982374",     // Optional: for tracking session
  "initialGameState": {}         // Optional: preloaded board state from Game Logic
}
```

      - The GUI receives this and performs the following:
        - Opens the Checkers game screen
        - Sets Anica’s UI to show her as Player 2
        - Initializes the game board with the correct orientation
        - Displays “Your opponent is Supan. Waiting for their move…”
          - This ensures each player is placed on the correct side of the board,
          - Guarantees the correct turn order and move behavior
          - Helps Networking/Game Logic teams sync game state properly

3. In Game Screen

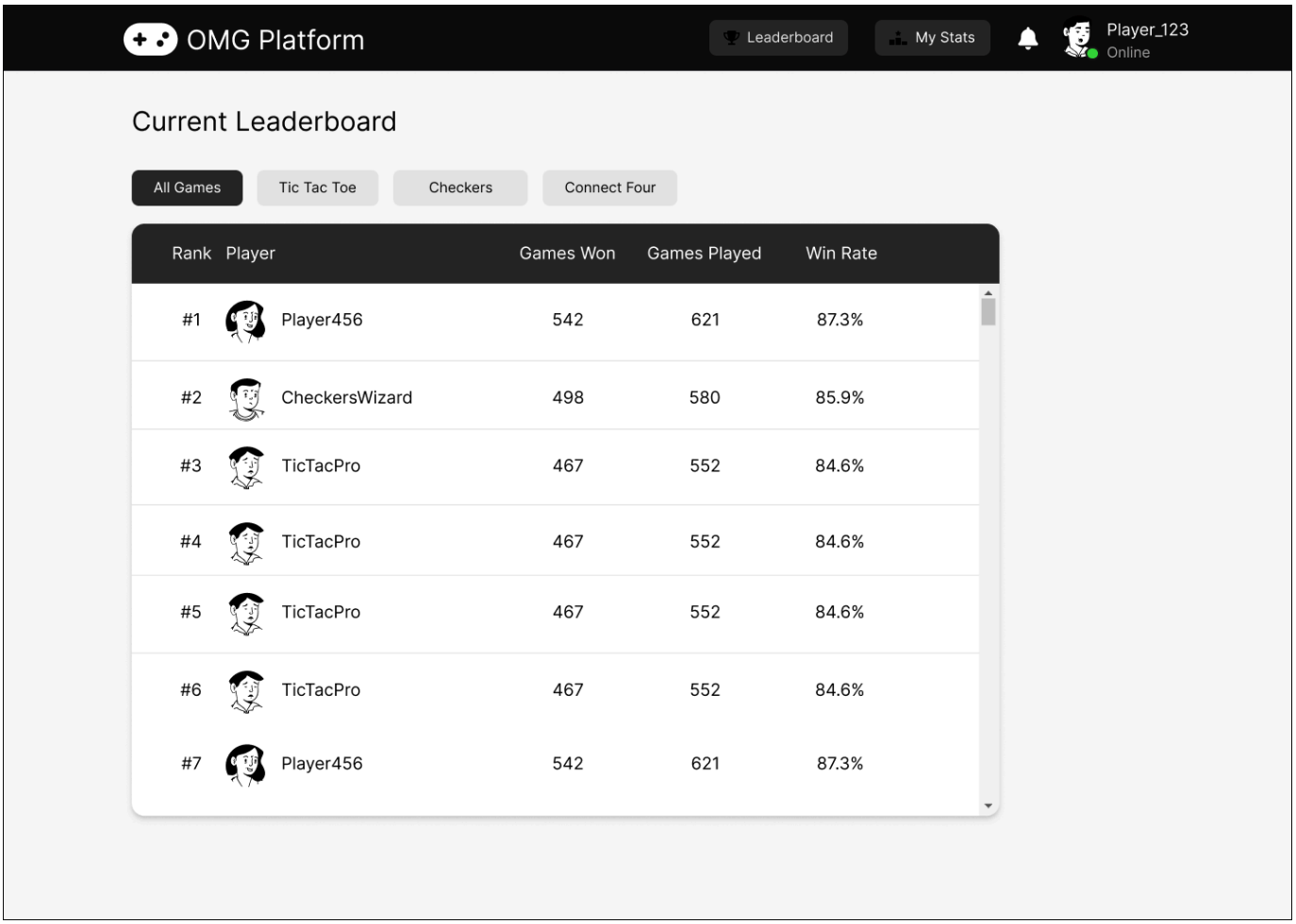


Board Games will have **two modes**: Casual Mode (for playing with friends) vs Ranked Mode (when versing an opponent with similar ELO/rank)

Each Game Screen should include:

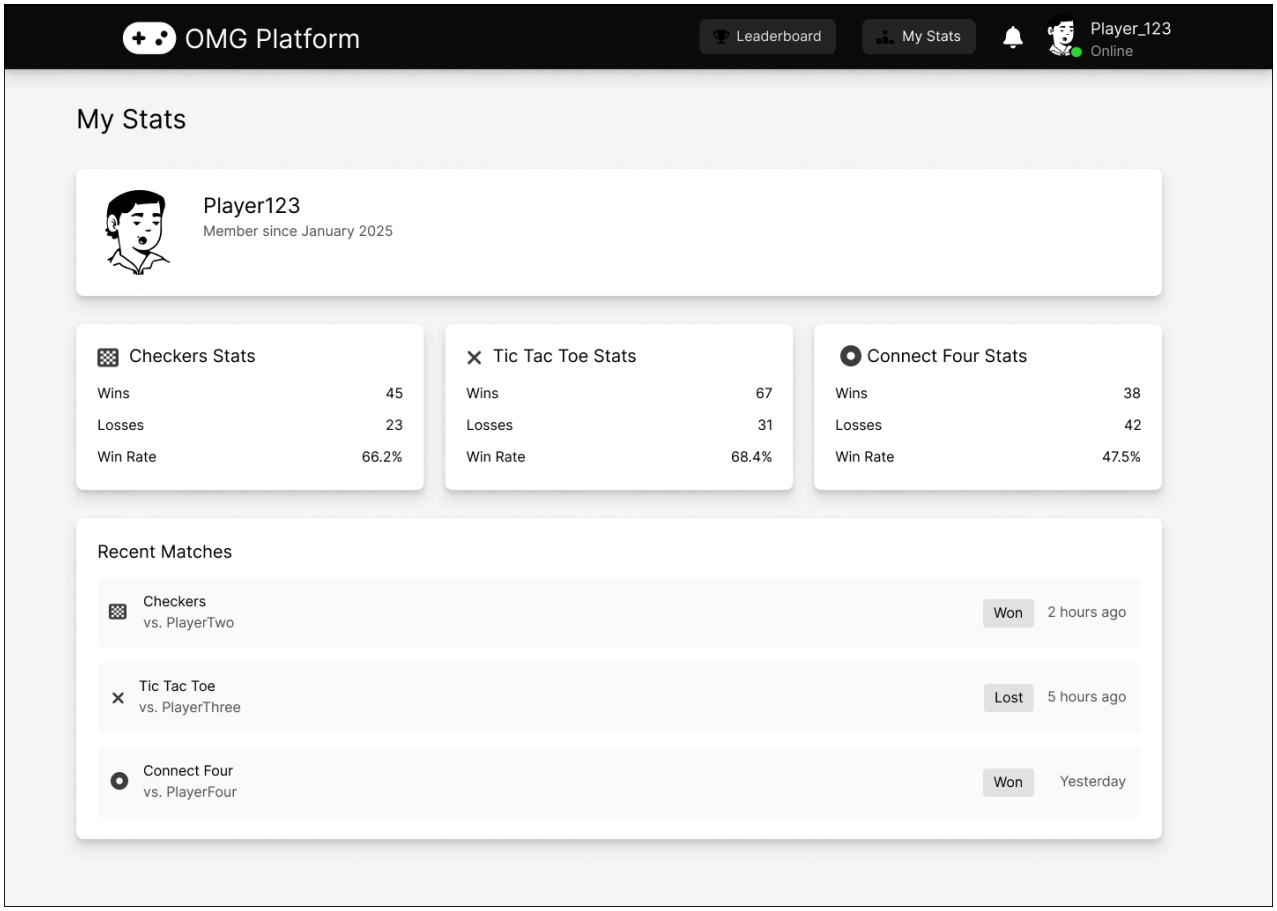
- Game board rendering
- Interactive pieces (clickable tiles, drag/drop if needed)
- Player names / turns indicator
- Game chat panel (see below)
- **Action Buttons**
    - Rematch
    - Surrender
    - Draw
        - **Player should lose points when "Surrendering" (for Rank Mode)**
        - **RE: When a player Draws: should they "lose" as well? How does that work with ELO rating/rank?**

4. Leaderboard Screen



- **GUI Responsibilities:**
  - Display player rankings, game-specific filters, win rates, and games played.
  - Implement tab switcher to filter by game (Tic Tac Toe, Checkers, Connect Four).
  - Games Won
  - Games Played
  - ELO
  - Win Rate
  - Ability to scroll or filter based on type of game
- **Leaderboard & Matchmaking Team Responsibilities:**
  - Provide an API or method to retrieve leaderboard data, filtered by game type.
  - Return player ranking, win %, games played/won, and avatar reference.
- **Expected Integration:**
  - GUI calls `getLeaderboard(gameType)` (?) to populate the table.
  - Needs to support pagination or scrolling for long lists. - How long should the leaderboard be?

5. My Stats Screen



- **GUI Responsibilities:**
  - Display per-game win/loss stats for the logged-in user.
  - Show user profile section and recent match history.
- **Leaderboard & Matchmaking Team Responsibilities:**
  - Provide stats per game for a user ID.
  - Return recent match history (opponent, game, result, timestamp).
- **Expected Integration:**
  - GUI calls `getUserStats(userId)` and `getRecentMatches(userId)`.(?)
  - Should return aggregated win/loss counts per game and last 5-10 matches.

General Integration Expectations

- Game Logic and Matchmaking teams will expose clearly named, version-controlled methods/APIs.
- Returned objects should include all data needed for display (e.g., player stats, usernames, avatars, win rate, etc.).
- GUI team will handle error messages and edge cases like: "Match not found", "Stats not available", etc.

Questions to ask:

1. Ranked vs Casual Mode Tracking

Are we planning to maintain separate stats for Ranked and Casual matches?

- Will only **Ranked** games contribute to leaderboard rankings?
- Will the database or API distinguish between the two modes?

  (e.g., `gameMode: ranked` vs `casual` in game results)

2. Scoring System / Ranking Logic

"How will player rankings be calculated?"

- Will we use **ELO**, **win/loss ratio**, or both?
- How are ties, forfeits, or disconnects handled in terms of rank?

3. Leaderboard Display

What fields will be available to the GUI team when rendering the leaderboard?

- Will we receive:
  - Username
  - Rank / ELO score
  - Games played
  - Win rate
- Will the leaderboard be global or game-specific (e.g., a leaderboard per game type)?

- Should users be able to filter the leaderboard (e.g., by game type, top 100, friends only)?

4. My Stats Screen

What data will be available for a user's personal stats screen?

5. Edge Cases

- How will the leaderboard handle edge cases?
- If a player disconnects or leaves mid-game, is it a loss?

- If a game is ended early by mutual consent (via 'Leave Game'), does it count?

- What happens if a player surrenders?

Meeting Minutes

Question: How is everything being ranked?

- Currently, everything is being ranked by ELO
- **ELO rating for each board game, not together *****
- Sorting each player into "rank" - **Rank tiers will be planned as an additional feature**
- Separate leaderboard structure for each board game; **each game will be tracking: games won, games played, as well as win ratio**

- **RE: Calculating Draws:**
  - "If a game ties up, are you calculating that?"
    - Specifically for "My Stats"/User Profile screen
    - **Himanshu mentioned that he is waiting on Kevin (Networking Team)?**
      - Need server and player classes
      - Issue brought up RE: "too much information" that is not needed to be tracked
    - Himanshu mentioned that he is creating a placeholder player class which gives the functionality - it is not final at all, it is there to help them with their code, have a basic command line implementation - should have it completed by "tonight"

- **RE: Match History (such as for Main Dashboard screen)**
  - "we haven't thought about match history"
    - Match history? We cannot store match history for a single player, we would have to have a different class(?) - alot of work for what it is - **have to talk to Boya about this** - he suggested JSON files - could be a nice to have

- **RE: Matchmaking Lobby ; how matchmaking is happening**

    - More than 500 elo difference - 2 -3 elo points per draw for the person with the lower ranking
    - First two seconds - 200 plus or minus
    - Then it goes to 400 plus or minus
    - If it doesn't get any person in a minute - it pans out
    - **It spits out an error that you couldn't find a person**

- **RE: Random Match addition**
    - Discussion around using Random Match selection once matchmaking lobby "times out" - the user has the option to match with someone "randomly"
        - Timed Out selection would be:
            - Random Match
            - Go Back (to Main Dashboard)
            - Try Again (try to attempt to match with someone again)

    - Mentioned to add a "Warning" when a user decides to do Random Matching - such as when a user with a lower ELO matches with someone with a higher ELO and wins - could impact higher ELO's score.

- **RE: Player disconnection**
    - Himanshu stated that the user would lose points to their ELO - "same rate as if they lost the game"

Summary of Meeting:

Ranking & ELO System

- **All games currently use ELO rating systems.**

- ELO is **tracked separately for each board game** (Tic Tac Toe, Checkers, Connect Four).

- There will be **individual leaderboards per game**, displaying:
    - Games won
    - Games played
    - Win ratio
- **Rank tiers** are planned as an additional feature to group players

Match History (Dashboard Feature)

- Match history is **not currently implemented**.

- Himanshu mentioned:

    - Storing match history per user would require a separate data structure or class.
    - Considered a **"nice-to-have"** feature.
    - Boya suggested using **JSON files** for lightweight match history storage.
    - To be discussed further.

Draw Calculation

- **Draws are accounted for** in leaderboard logic.
- In a draw:
    - **Low ELO players may gain 2–3 points**
    - **High ELO players may lose a small number** — depending on the ELO difference
- Draws are **ELO-neutral** in close-match situations but may affect scores in high/low mismatches.

Matchmaking Logic

- Matchmaking is ELO-based with **expanding search range**:

    - First 2 seconds: ±200 ELO
    - Then increases to ±400
    - After 1 minute: matchmaking times out
      If no match is found → returns error

Post-timeout Options (UI Suggestions):

1. **Random Match** (match with any available player)
2. **Try Again** (restart matchmaking process)
3. **Return to Dashboard**

- A **warning should appear** if the user selects Random Match, informing them of potential ELO risks (e.g., low ELO user beating high ELO could significantly affect scores).

Action Items & Next Steps

Action Item	Owner(s)	Due Date / Notes
Finalize **placeholder Player class** to support ELO and game stat tracking	Himanshu	Expected by end of day (today)
Confirm ELO calculation logic for **draws** and ELO mismatch scenarios	Leaderboard Team	Document example cases for GUI + testing
Coordinate with **Networking Team (Kevin)** to complete backend support for stats/draws	Himanshu, Kevin	Required for full “My Stats” + leaderboard features (?)
Decide whether to implement **Match History** (or defer to future version)	Leaderboard Team, Boya	Optional — explore JSON-based lightweight solution
GUI Team to design **matchmaking timeout screen** with 3 options:	GUI Team	Include Random Match, Try Again, Return to Dashboard
Implement **warning popup** for Random Match feature (ELO risk explanation)	GUI Team	Wording + design required
Decide on approach for **Rank tiers**	Leaderboard Team	To be scoped as an enhancement
Clarify how to handle **player disconnections** in ranked matches	Leaderboard + Networking	Confirm ELO penalty rules and player feedback
Finalize leaderboard API fields for GUI integration	Leaderboard Team	Include: wins, losses, draws, win rate, rank,etc