

# **Use Case: Log Game Result**

**Primary Actor:** Game Session Manager

**Secondary Actors:** The game players, the game's leaderboard system, and database.

**Goal in Context:** To allow the system to record the outcome of a finished match, while updating player stats and adjusting leaderboards accordingly.

**Preconditions:**

- The game has to have been played and complete with a result.
- The system has to have access to the relevant game data, so players, scores, and outcomes
- Players must be logged in to store and ensure result tracking.

**Trigger:** The session ends, and system is prompted to load and store results

**Scenario:**

1. The game completes through a win, draw, or forfeit.
2. The game session leader/manager then initializes game data
3. The system then formats the data, and validates accordingly based on formatting The
4. The leaderboard system then processes the final game result to determine any rankings
5. The leaderboard then adjusts player rankings and statistics based on stat conditions
6. The final result is stored in the game database
7. The players then receive a notification confirming their match result has been recorded

**Postconditions:**

- The game result is recorded in the system.
- Players stats and leaderboards are adjusted
- The system is then able to retrieve match data when needed.

**Exceptions:**

- If the system fails to log the result due to a server issue, data should be retried or temporarily stored
- If data is missing or corrupted, an error is logged, and then data may need manual evaluation

- If a player disconnects, the system should assign a forfeit to the player

***Priority:***

High as this is vital for maintaining game stats , and player performance history.

***Channel to Actor:*** Server-side process

***Open Issues:***

- Should unranked games be logged separately from ranked matches?
- How long should data be stored before archiving?
- Should players be able to review past match results in their profiles?