# GUI Critique Summary and Solutions

**Planning Document Critiques & Solutions**

For the most part our review come clear. We were posed to finish our assignment in due time, with the reviewing group noting that our timeline, for the most part, was feasible and possible to achieve. There were, however, certain recommendations given.

In terms of risks the following were identified:

- GUI depends highly on Networking and Game-Logic teams, to mitigate this problem be sure to have constant communication so that issues can be identified early.
- The scope of GUI is large, there are many things that must be completed by this team, to mitigate the problem ask teams that have nothing going on for some help and spread the most important tasks among your own members.
- GUI Development Plan Enhancements could prove to be a challenge such as, UI components that can be reused across different game interfaces, and ensuring the UI adapts well to different screen sizes, including mobile devices. Which are really good ideas, but I don't think it should be a focus of the project.

Due to some of these risks, the following suggestions were provided:

- One suggestion we have is to start tests early that way issues can be found faster.
- Another would be to focus on the main components first, such as making sure that games work, or that they are easy to understand/play... and then if there is time, then the enchantments can be done. Which is what your plan seems to be, but we can't tell for sure.
- And one more suggestion we have is for each team member to focus on one thing, for example one member focuses on Game Dashboard, another on Moving pieces... that way no one will be overwhelmed with responsibilities.

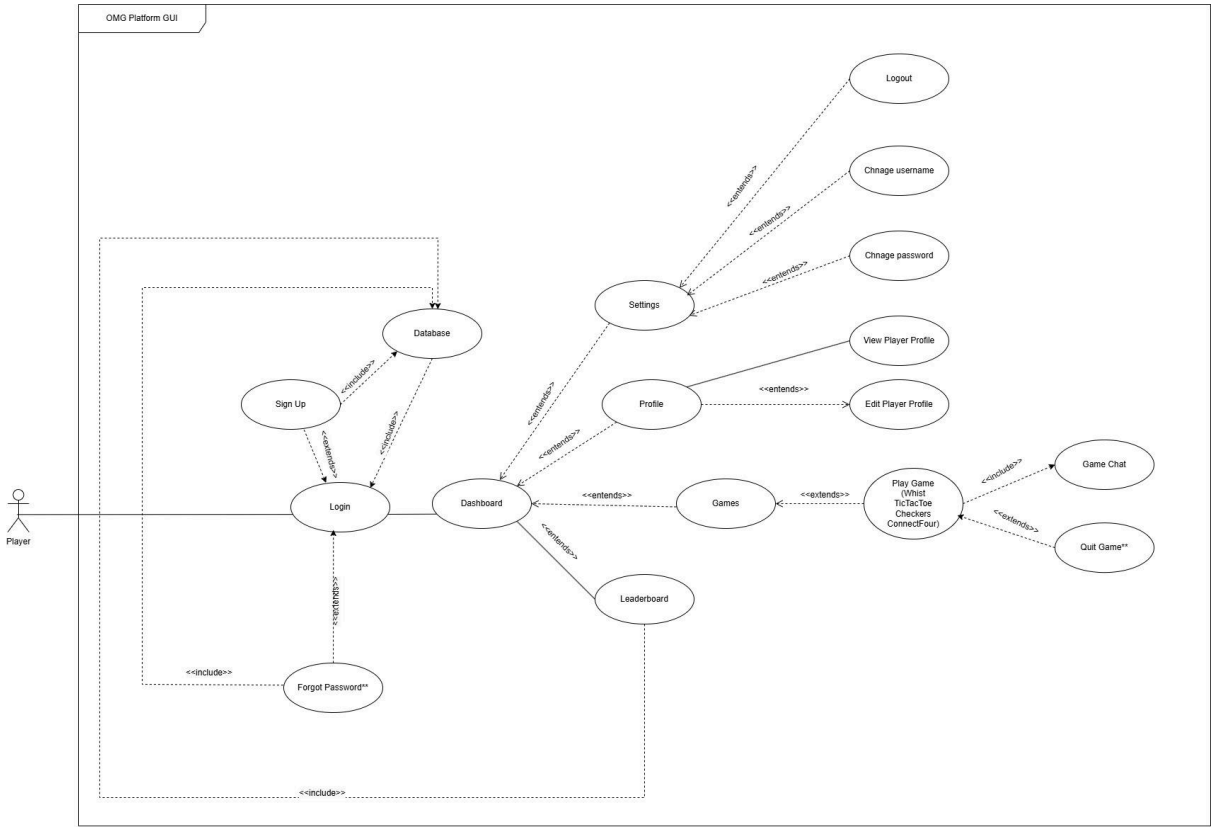
## Use Case Diagram Critique and Solutions

There were multiple recommendations and changes made to our GUI use case diagram and below are some of the critiques along with their recommendations.

The following critiques were given:

- Extends and includes arrows point incorrectly at times.
- There are no post-conditions in leveling/matchmaking use cases.
- There are multiple vague GUI interactions (e.g: Profile).

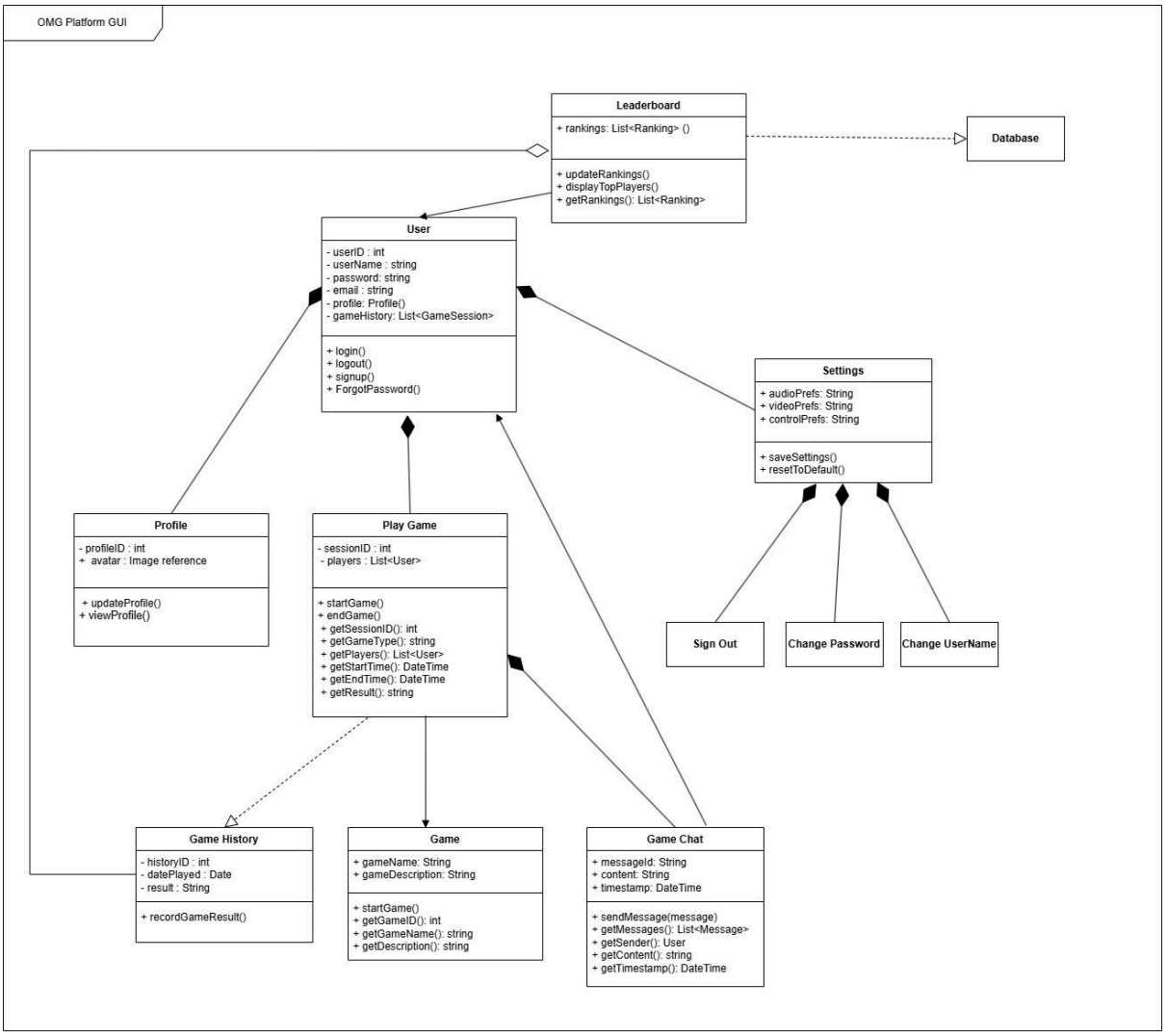
Below are the changes made to the use case diagram based on the critiques:




The following diagram aligns with the changes that were made from both the recommendations and future development with the project.

## Class Structure Diagram Critique and Solutions

There were multiple recommendations and changes made to our GUI use case diagram and below are some of the critiques along with their recommendations. The main recommendations was the luck of arrows to indicate dependencies or associations. Below are the changes made to the use case diagram based on the critiques:




The following class structure diagram aligns more with the changes that were made from both the recommendations and future development with the project.

**Feature Proposal**

Based on our work, we were given several recommendations in order to improve functionality and user experience. Below is the list:

1. Customizable Themes & Accessibility Modes:
    - This feature enables users to personalize UI colors and switch between preset themes (e.g., Dark Mode, Light Mode, Colorblind-friendly) to accommodate visual preferences and accessibility needs. This enhances platform appeal and supports a broader user base.
2. Session Owner Kick Control:
    - This feature allows the lobby creator to remove disruptive or unwanted players before a Whist match begins, with an AI bot replacing any mid-game disconnects. AI takeover prevents abandoned matches, preserving game flow and player satisfaction.
3. Password Visibility Toggle:
    - This feature involves adding a small toggle or icon in the login screen's password field to switch text between hidden and visible.
4. Spectate Mode:
    - This feature would offer a real-time viewing option for ongoing games, with dedicated spectator slots and a separate chat channel.