

Overall

The feedback we received was generally of poor quality. Both groups made factually incorrect statements, like how group A asserted that access modifiers were missing from the class structure diagram, yet did not show where they were missing, and how group B asserts that we used an integer for error handling yet did not show where. Both groups assumed errors where none were present, like how group A asserts security flaws then fails to explain where they are present, and group B asserts that having one class responsible for the network leads to performance issues. Finally, group B suggested many additional features, while we appreciate the effort, we do not have the resources to implement all of these. For these reasons it is impractical to show all the changes they suggested, instead assume all changes not mentioned in this document were disregarded for similar reasons.

Summary of Changes

Networking

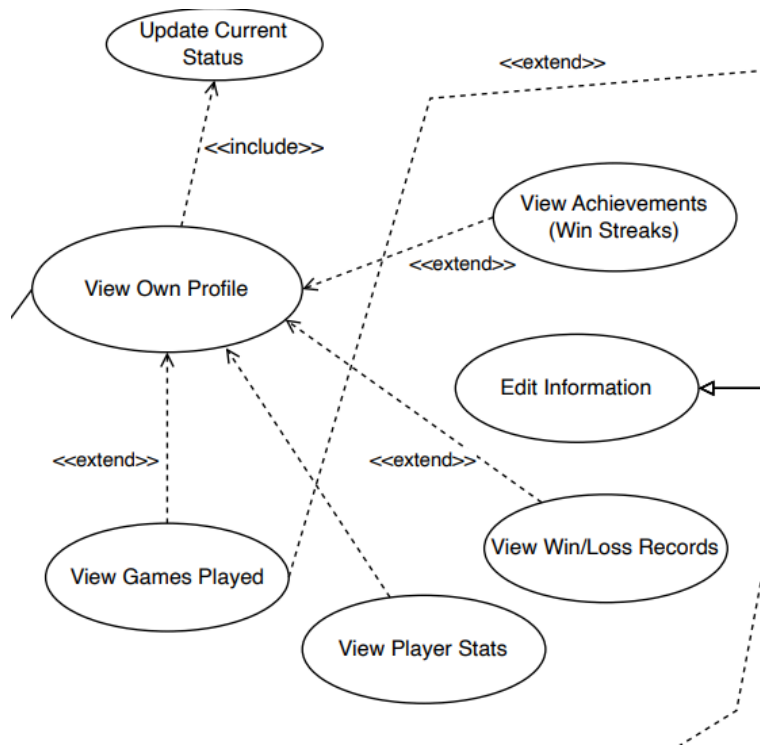
Clarified roles of network classes. Added a PlayerHandler class that represents a player in the server.

Game Logic

Made game state universal across all games.

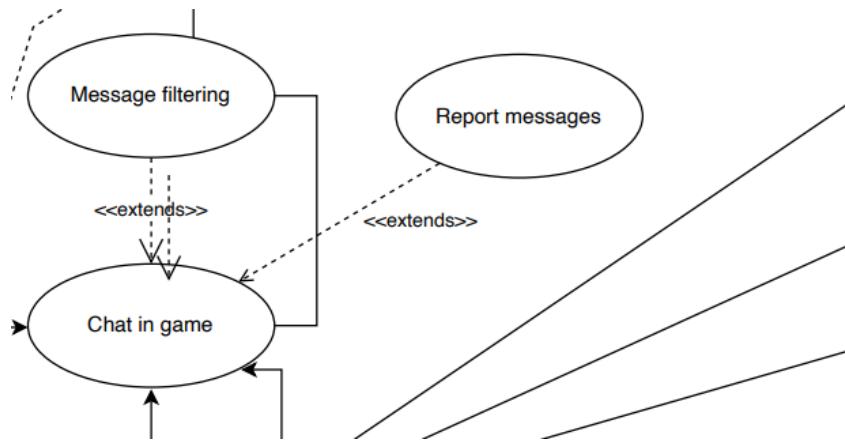
Achievement & Badge System

An achievement and badge system has been successfully implemented. Achievements are awarded based on milestones such as the number of games played or won. These metrics are tracked via the backend SQL database. Player statistics are continuously compared against predefined achievement criteria, and badges are awarded automatically upon meeting those requirements. Achievements are displayed in the user interface on the profile page.



Offensive Message Censorship

A basic censorship mechanism has been added to prevent the sending of offensive messages. A predefined list of censored words is maintained on the client side. Before any message is sent, it is scanned locally for any flagged words. If such content is detected, the message is blocked from being sent to the server, and a warning notification is shown to the user.



Game Specific Matchmaking

Game-specific matchmaking functionality has been implemented as planned. This was achieved by maintaining separate matchmaking queues for each supported game. During the connection process, the client specifies the game they wish to play, and they are added to the corresponding matchmaking queue. Additionally, a skill based matchmaking system has been implemented, where each player has been assigned a rank based on the number of wins in that game, and the player will be matched with another player with the closest rank.

Post-Match Analysis & Insight

The Post-Match Analysis & Insights feature was not implemented due to time constraints during this development iteration. The feature would have required significant effort, including the integration of AI modules for each game to analyze gameplay and provide feedback, as well as UI modifications to present the insights. Given the scope and complexity, it was deemed unfeasible within the current timeline.