Grade Proposal:

Proposed Grade: B+

Justification:

System Design and Documentation:

The project should get praise for its modular architecture, effective use of abstraction and encapsulation (especially with the separation of GUI and game logic), and clear strengths in components like the networking and leaderboard systems. However, it should also be highlighted that there are several areas for improvement: inconsistencies in the level of detail across documents, redundant or unclear elements in the use case diagrams, and a lack of integration in some UI designs. These issues, particularly the discrepancies in technical detail and the appearance of features in isolated parts of the documentation, may complicate implementation and overall system cohesion.

Overall, while the project has a solid foundation and demonstrates thoughtful design practices, the inconsistencies and integration challenges prevent it from reaching excellence. With a few targeted refinements—such as improving the consistency across documents, enhancing error handling, and better aligning the UI with system components—the project could be significantly improved. Given its current state, a grade of **B+** seems appropriate, as the work shows promise but still requires further refinement to achieve a higher level of quality.

Planning and Timeline:

Echo team's planning was graded as **B**, because they didn't account for Iteration 2 or their demo videos, but the presentation was very professional, and they paid attention to potential conflicts in their timeline.

Originality and Functionality:

This project is notably original in its approach to developing the games, leveraging contemporary development practices like Agile Scrum and Git-based collaboration.

In terms of functionality, the project plan details a robust architecture that incorporates comprehensive unit and integration testing, continuous deployment practices, and clear sprint milestones to ensure timely delivery of features. The emphasis on modular design and team collaboration not only enhances maintainability and scalability but also ensures that the system can handle the complexities of multiplayer interactions and real-time updates effectively.