Feasibility Assessment – Echo Team

---

1. Overview of Planning Principles

Echo Team's project plan is generally well-structured, following a logical progression from **Design (P1) → Implementation (P2) → Integration & Testing (P3)**. Tasks have been broken down and assigned systematically, ensuring a smooth transition from planning to coding across all teams.

However, to execute the integration and testing effectively in P3, it may be necessary to revisit the prioritization and scheduling of a few tasks. Efficient execution within the limited timeframe of P3 requires early testing and phased validation.

---

2. Correctly Prioritized Goals

The following tasks were appropriately scheduled and contribute positively to the overall feasibility, particularly as the team enters P3:

- **UI Wireframes & Design (P1)**
  Completed in Sprint 1, this enabled implementation teams to work with clear UI references and contributed to a smooth P2 transition.
- **Game Logic & Class Structures (P1)**
  Early completion of class diagrams and use case definitions for all three games reduced ambiguity during implementation and improved efficiency.
- **Foundational API and Communication Layer Implementation (P2)**
  With key components of authentication and networking implemented in P2, the team is well-prepared for P3 integration and testing.
- **Well-distributed GUI Implementation Tasks (P2–P3)**
  The GUI team's tasks in Sprint 3 are broken into small, well-defined elements (e.g., animations, responsiveness, integration), which supports steady and feasible progress.

---

3. Incorrectly Prioritized Goals

The following items present scheduling or prioritization concerns that could cause bottlenecks if left unaddressed:

- **Backend Integration (Authentication & Networking) is Too Late in P3**
  Major integration tasks involving authentication APIs, networking, and matchmaking are scheduled for mid-to-late P3 (March 29 onward). This leaves little buffer for debugging

or adjustments. A recommendation is to begin basic connectivity and integration tests earlier (March 22–25).

- **Error Handling and Logging Implementation is Missing or Delayed**
  Error logging and failure response handling are not clearly specified as part of the integration plan. These features are crucial for debugging and should be implemented and tested in early P3.

---

4. Missing Goals

To mitigate integration risks, the following tasks should be introduced or emphasized in P3:

- **Pre-Integration Connectivity Testing (GUI–Auth–Network)**
  Since each team developed components on separate branches, minimum viable testing for API communication should be conducted between March 22–25. This includes confirming basic login and message-passing functionality.
- **API Failure Scenario Testing**
  While success cases (e.g., login success) are covered, there is no mention of failure cases such as network drop, incorrect input, or timeouts. Basic error-handling UI and logs should be tested and refined.
- **Clear Communication of Integration Dependencies**
  With dependencies spanning authentication, networking, matchmaking, and game logic, it is recommended to document and communicate the intended integration sequence (e.g., Auth → Network → Matchmaking → GUI → Game Logic) to avoid misalignment.

---

5. Conclusion & Recommendations

Echo Team's plan is well thought out and executed effectively through P1 and P2. The design work and implementation were completed with solid vision and coordination. However, the upcoming P3 phase introduces new risks, primarily due to the heavy load of integration and testing tasks in a limited timeframe.

Recommendations:

1. **Run Initial Integration Tests between March 22–24, 25**
   - Verify basic API connectivity among Auth, Network, and GUI.
   - Confirm that components share compatible input/output formats.
2. **Move Part of the Load Testing to March 25–28**
   - Test simultaneous logins and matchmaking requests early to identify performance bottlenecks.
3. **Clarify and Share Integration Sequence**

- Suggested order: Authentication → Networking → Matchmaking → GUI → Game Logic
- A clear plan reduces miscommunication and simplifies debugging.

4. **Implement and Test Logging & Error Handling Mechanisms**
    - Ensure user-friendly error messages and backend logs are in place to assist with final-stage troubleshooting.