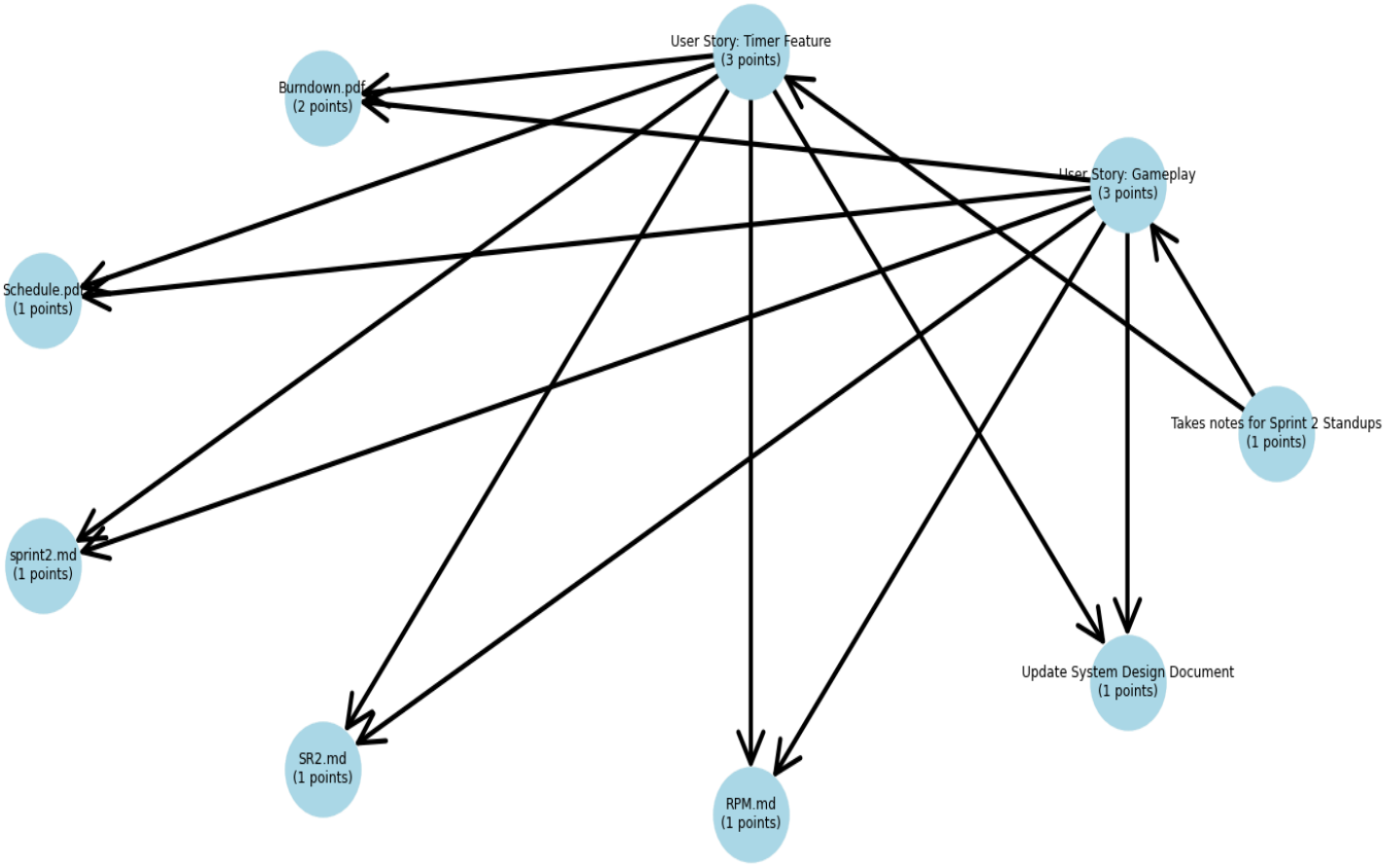
Sprint 2 Network Diagram



The edges in the network diagram represent dependencies between tasks, showing which tasks must be completed before others can begin. Here's a breakdown of the edges and their meaning:

1) **Takes Notes for Sprint 2 Standups:**

    -This is the starting task. Other tasks depend on it to gather updates, organize workflows, and coordinate team efforts.

    - Dependent Tasks:

        - User Story: Gameplay

- User Story: Timer Feature

2) User Story: Gameplay and **User Story: Timer Feature**:

- These tasks are dependent on "Takes Notes for Sprint 2 Standups."

- Both tasks are foundational user stories required for the sprint.

- Dependent Tasks:

- Burndown.pdf

- Schedule.pdf

- sprint2.md

- SR2.md

- RPM.md

- Update System Design Document

3) Documentation and Reporting Tasks (Burndown.pdf, Schedule.pdf, sprint2.md, SR2.md, RPM.md, Update System Design Document):

- These tasks rely on the completion of "User Story: Gameplay" and "User Story: Timer Feature" as they document or implement their outcomes.

Critical Path Analysis:

The critical path represents the longest sequence of dependent tasks, which determines the minimum time needed to complete the sprint.

1. **Critical Path**:
   - **Takes Notes for Sprint 2 Standups → User Story: Gameplay → Burndown.pdf**
   - This path takes the longest time:

     - Takes Notes for Sprint 2 Standups (1 point)

     - User Story: Gameplay (3 points)

     - Burndown.pdf (2 points)

     - Total Time/Points: 6 points

2. **Other Paths**:

- Non-critical paths, such as tasks like **SR2.md** and **RPM.md**, are parallel to the critical path but take less time/points.

To keep the sprint on schedule, regular standup meetings ensure all team members are aligned and aware of dependencies. We also continuously track task progress using an agile board and adjusting priorities if needed. We also prompt resolution of blockers, identified during standups.Lastly, the effective distribution of work across team members to reduce delays in non-critical tasks.

What went wrong, and what did you learn from it?

- **What went wrong:**
  - Delay in completing "User Story: Gameplay" or "User Story: Timer Feature" could cascade and delay dependent tasks.
  - Team coordination issues during standup meetings, such as unclear goals or unaddressed blockers.
- **What did we learn from it**:
  - Clear communication of task priorities and dependencies is essential.
  - Buffer time should be included for critical tasks to account for unforeseen delays.
  - Effective workload distribution can prevent overburdening team members handling critical tasks.