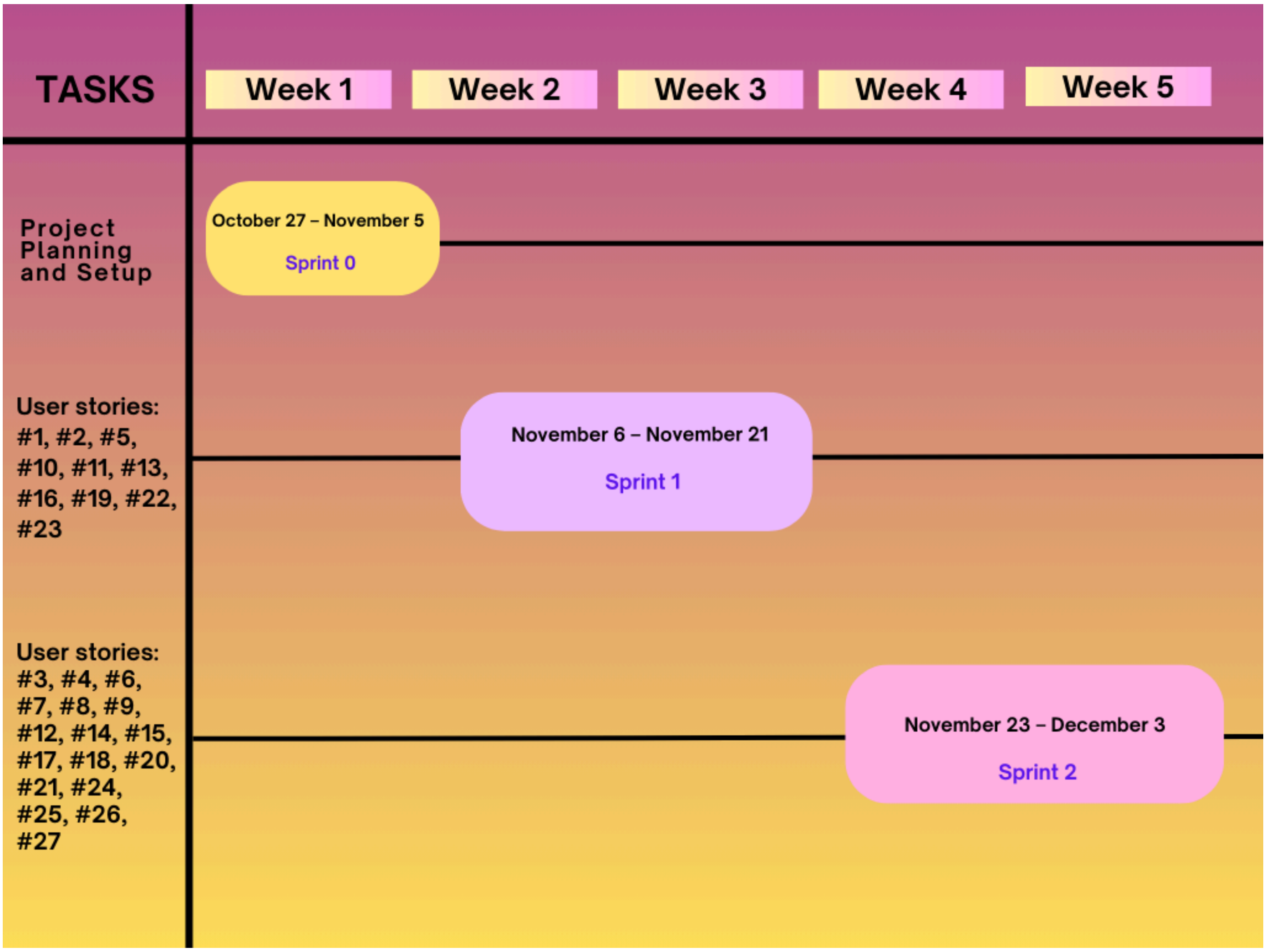
Project schedule Gantt Chart:



Critical Path Computation and Explanations

Delays in any of the critical path tasks directly affect the overall project timeline. Below is the computation we have done for the implemented user stories based on actual progress in order.

Critical tasks:

- Sprint 0: Planning and setup.
- Sprint 1: All user stories → #1 → #2.
- **Sprint 2:** #6 → #7 → #21.
- **Sprint 3:** #25 → #18.

Criteria:

These sequence and their dependencies are calculated by considering the following criteria:

1. Which user story acts as a prerequisite for another user story, meaning the user story can't be implemented if the base user story is not present

2. Which use stories are dependent on other user stories and the user story is not completed if the other user story is not present?

3. Delays in what user stories will expand the project timeline effectively?

Reasoning:

- Without User registration (#1)and User login (#2), None of the user stories can be completed as the user or admin must sign in before accessing the system.

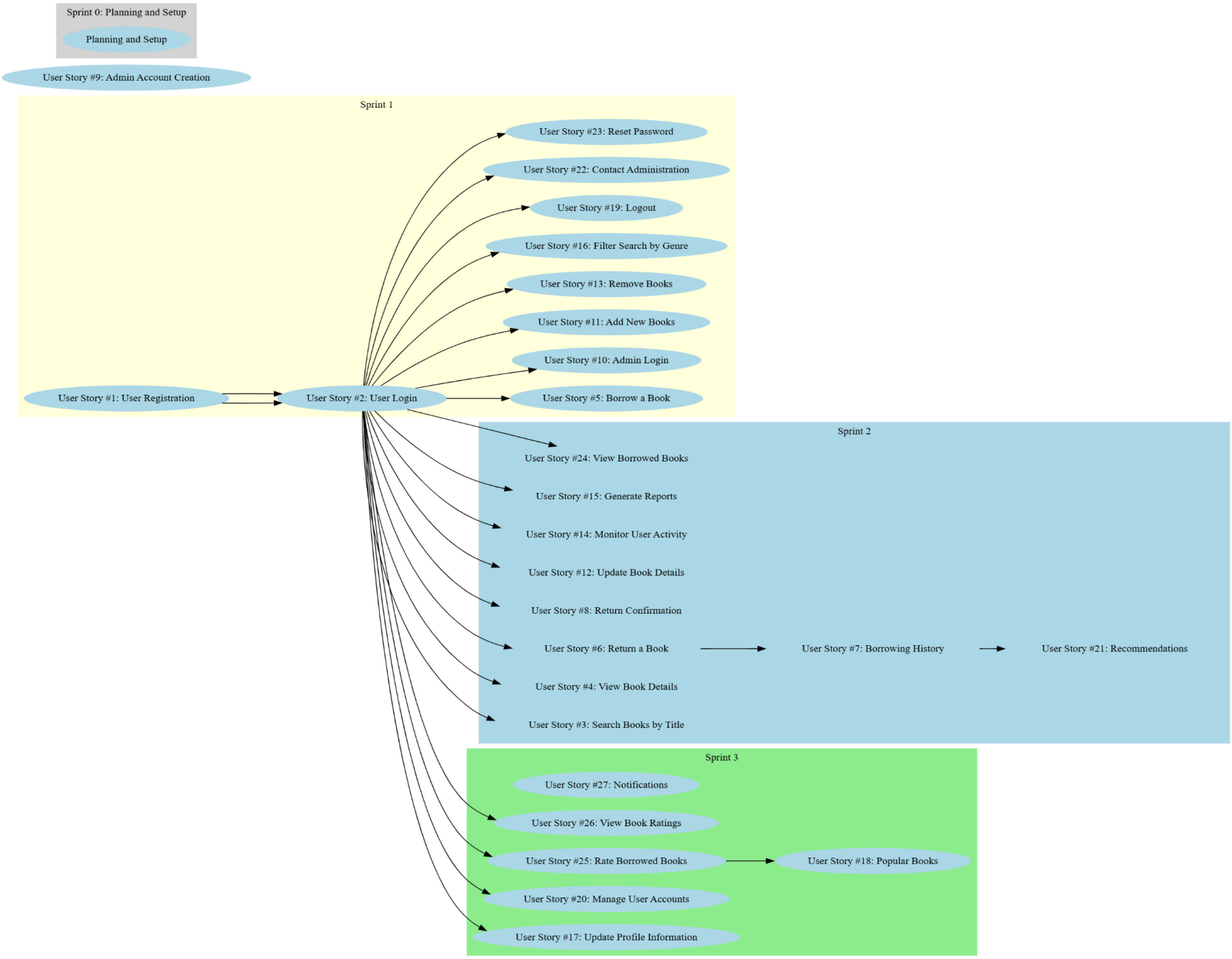
- Recommendations (#21) depend on returning books (#6) and borrowing history (#7), as the user can't see the recommendation without the user actually borrowing a book first.

- Popular books (#18) cannot be displayed without book ratings (#25), as the popular books will only be shown if the user has provided ratings.

- Delays in critical tasks, such as #5 (Borrow a book), will delay tasks like #6 (Return a book) and #21 (Recommendations).

- Completing non-critical tasks earlier will not impact the overall timeline, as they are not on the critical path.

Network Diagram:



Sprint 0: Planning and Setup
Planning and Setup
User Story #9: Admin Account Creation
Sprint 1
User Story #23: Reset Password
User Story #22: Contact Administration
User Story #19: Logout
User Story #16: Filter Search by Genre
User Story #13: Remove Books
User Story #11: Add New Books
User Story #10: Admin Login
User Story #1: User Registration
User Story #2: User Login
User Story #5: Borrow a Book
Sprint 2
User Story #24: View Borrowed Books
User Story #15: Generate Reports
User Story #14: Monitor User Activity
User Story #12: Update Book Details
User Story #8: Return Confirmation
User Story #6: Return a Book
User Story #7: Borrowing History
User Story #21: Recommendations
User Story #4: View Book Details
User Story #3: Search Books by Title
Sprint 3
User Story #27: Notifications
User Story #26: View Book Ratings
User Story #25: Rate Borrowed Books
User Story #18: Popular Books
User Story #20: Manage User Accounts
User Story #17: Update Profile Information

Keeping the Sprint on Schedule

To ensure the sprint stayed on schedule, we focused on in-advance planning and equal task allocation. We distributed the tasks among members equally to ensure progress.

We broke it into smaller, doable tasks, such as designing the database schema for notifications, creating the backend logic for generating notifications, and implementing the frontend display. Daily standups were held to track progress, address blockers, and adjust priorities if needed. Regular communication between members was made through Discord.

Unfinished user story:

User Story #27 had not been completed within sprint 2 only due to lack of time. As we wanted to have a comprehensive project, we tried to add more user stories, but the time allocated wasn't enough to prioritize implementing this user story. Next time, we would do a better time allocation and a more realistic amount of tasks to be able to complete them all in the specified time.

Sprint velocity comparison:

The planned velocity for Sprint 2 was higher compared to Sprint 1 because it included a larger number of user stories to implement. In Sprint 1, the team completed 10 user stories due to the setup phase and initial learning curve. By Sprint 2, the team's velocity increased as members became more familiar with the project and GitHub collaboration. The other reason was in sprint 2 we each implemented 3 user stories (15 in total) which required faster task execution and better collaboration.