

# UniVerse Sprint 2 Schedule and Critical Path

## 1. Purpose

This document outlines the features in the UniVerse application, their dependencies, the network diagram, critical path, strategies for keeping the sprint on schedule, and lessons learned during Sprint 2. The goal is to ensure a clear understanding of our workflow, identify key areas for improvement, and provide actionable insights for any future sprints.

---

## 2. Task Identification and Dependencies

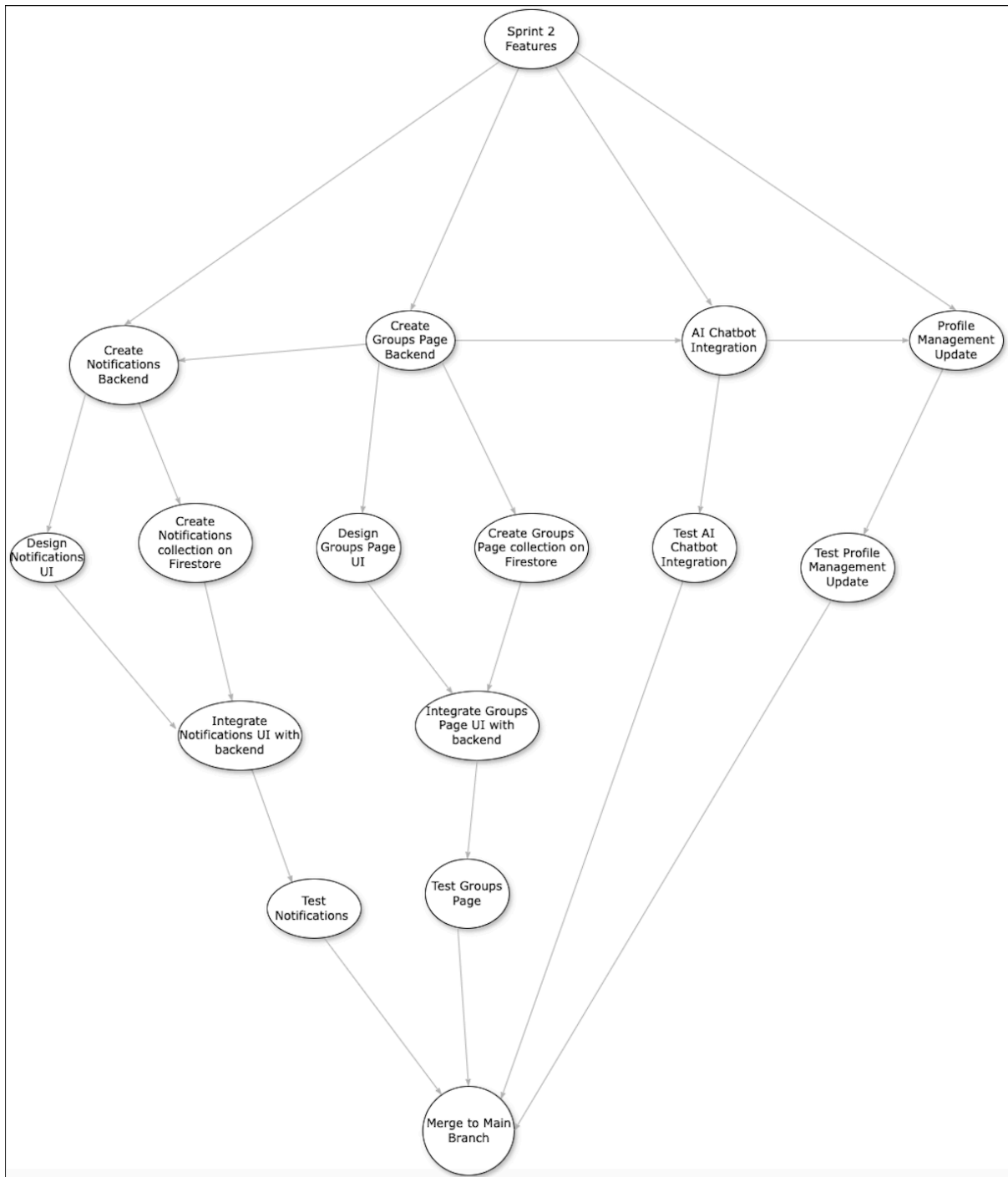
Below is the table of tasks performed in Sprint 2, along with their descriptions and dependencies:

| Task   | Description   | Dependencies                          |
|--|---|---------------------------------------|
| Create Notifications Backend                 | Set up backend classes for notifications.   | None                                  |
| Design Notifications UI                      | Create the interface for the notifications page.  | None                                  |
| Create Notifications collection on Firestore | Ensure the notifications backend is fully functional and is being stored in Firestore (database). | Notification Backend                  |
| Integrate Notifications UI with backend      | Connect the notifications UI with the backend.  | Notification Backend, Notification UI |
| Create Groups Page Backend                   | Set up backend classes for group functionality.   | None                                  |
| Design Groups Page UI                        | Create the interface for the groups page.   | None                                  |
| Create Groups Page collection on Firestore   | Ensure the groups page backend is fully functional and is being stored in Firestore (database).   | Groups page Backend                   |

|                                       |   |   |
|---------------------------------------|---|---|
| Integrate Groups Page UI with Backend | Connect the groups page UI with the backend.  | Groups Page Backend, Groups Page UI     |
| AI Chatbot Integration                | Integrate the chatbot into the messaging system.  | None                                    |
| Test Notifications                    | Test the notifications feature to ensure it works correctly.  | Integrate Notifications UI with backend |
| Test Groups Page                      | Test group creation and membership management.  | Integrate Groups Page UI with Backend   |
| Test AI Chatbot Integration           | Test chatbot messaging functionality.   | AI Chatbot Integration                  |
| Profile Management                    | Ensure users can edit interests and bio after it has been created. Ensure all images is converted to Base64 and stored in Firestore | None                                    |
| Test Profile Management Update        | Test editing interests, bio and uploading profile pictures.   | Profile Management                      |

### 3. Network Diagram

The network diagram visually represents all the features and their dependencies in Sprint 2. Each node corresponds to a task, and the arrows indicate dependencies between tasks. Tasks without dependencies are the starting points in the diagram.



**Diagram Key:**

- Nodes represent tasks.
  - Arrows represent dependencies.
  - The critical path is highlighted as the longest dependent chain of tasks.
-

## 4. Critical Path

The critical path represents the sequence of dependent tasks that determine the sprint's minimum completion time. Any delay in these tasks will delay the entire sprint.

### Critical Path Computation

Using the task dependencies, the critical path for Sprint 2 is:

1. Create Notifications Backend
2. Create Notifications Collection on Firestore
3. Integrate Notifications UI with Backend
4. Test Notifications System
5. Create Groups Page Backend
6. Create Groups Page Collection on Firestore
7. Integrate Groups Page UI with Backend
8. Test Groups Page

### Why These Features Are Critical

- **Notifications and Groups Features:** These are the backbone of the sprint. Both require backend readiness before integration and testing can begin.
  - **Testing Phase:** Delays in testing will postpone the sprint's final delivery.
  - **AI Integration:** Allow users to interact with an AI chatbot for quick support to any questions they may have.
  - **Profile Management:** This ensures that users have the flexibility to edit their bio and profile picture whenever they wish to.
- 

## 5. Strategies to Keep the Sprint on Schedule

To ensure timely completion of Sprint 2, the following strategies were implemented:

### Frequent Standups

- Conducted short meetings to review progress and identify blockers.
- Tasks were reassigned as needed to maintain momentum.
- Stand up meeting notes were taken to ensure all team members are on the same page.

### Task Prioritization

- Tasks on the critical path were prioritized for resources and attention.
- Secondary tasks were subsequently worked on to prevent resource contention.

### Pair Programming

- For tasks with higher complexity (e.g., Notifications and Groups Features), two team members collaborated to accelerate progress and reduce errors.

## Buffer Time for Testing

- Allocated extra time in the sprint schedule for testing and debugging phases, ensuring bugs and issues were resolved promptly.

## Open Communication

- Maintained transparent communication across the team using tools like Trello and Discord to ensure alignment on task priorities and deadlines.
- 

# 6. Lessons Learned

While Sprint 2 achieved significant milestones, certain challenges highlighted areas for improvement:

## Challenges

1. **Backend Readiness:**
  - Delays in backend development for the Notifications and Groups pages created a bottleneck for integration and testing phases.
2. **Complex Features:**
  - AI Chatbot integration required more effort than initially anticipated, delaying its testing phase.
  - Integrating the three tier architecture for the Notifications and Groups pages also took more effort than required.
3. **Iterative UI Fixes:**
  - Multiple design adjustments for the Groups page UI slowed down its finalization and integration.

## What Went Wrong

Using the network diagram, it is evident that:

- The complexity of AI Chatbot integration was underestimated during planning, leading to longer development and testing phases.
- Backend dependencies for Notifications and Groups introduced significant challenges, delaying integration and testing.
- Insufficient buffer time for tasks like the page a user will be taken to, when the user joins a group, caused testing to spill over into the final days of the sprint.

## Lessons Learned

1. **Early Backend Development:**

- Ensure backend readiness before starting dependent tasks like integration and testing.
  - 2. **Allocate Buffer Time for Complex Features:**
    - Identify high-complexity tasks during planning and allocate additional buffer time for unexpected challenges.
  - 3. **Stronger Initial UI Designs:**
    - Invest more time in initial UI design to minimize iterative adjustments during development.
  - 4. **Improved Communication:**
    - Encourage more frequent updates and cross-team collaboration to identify and resolve bottlenecks earlier.
- 

## 7. Conclusion

Sprint 2 showcased the team's ability to deliver key features like Notifications, Groups Page, AI Chatbot integration and Profile Management. However, delays in backend readiness and underestimated complexity highlighted areas for improvement. By addressing these issues, future sprints can ensure smoother workflows and more predictable timelines.