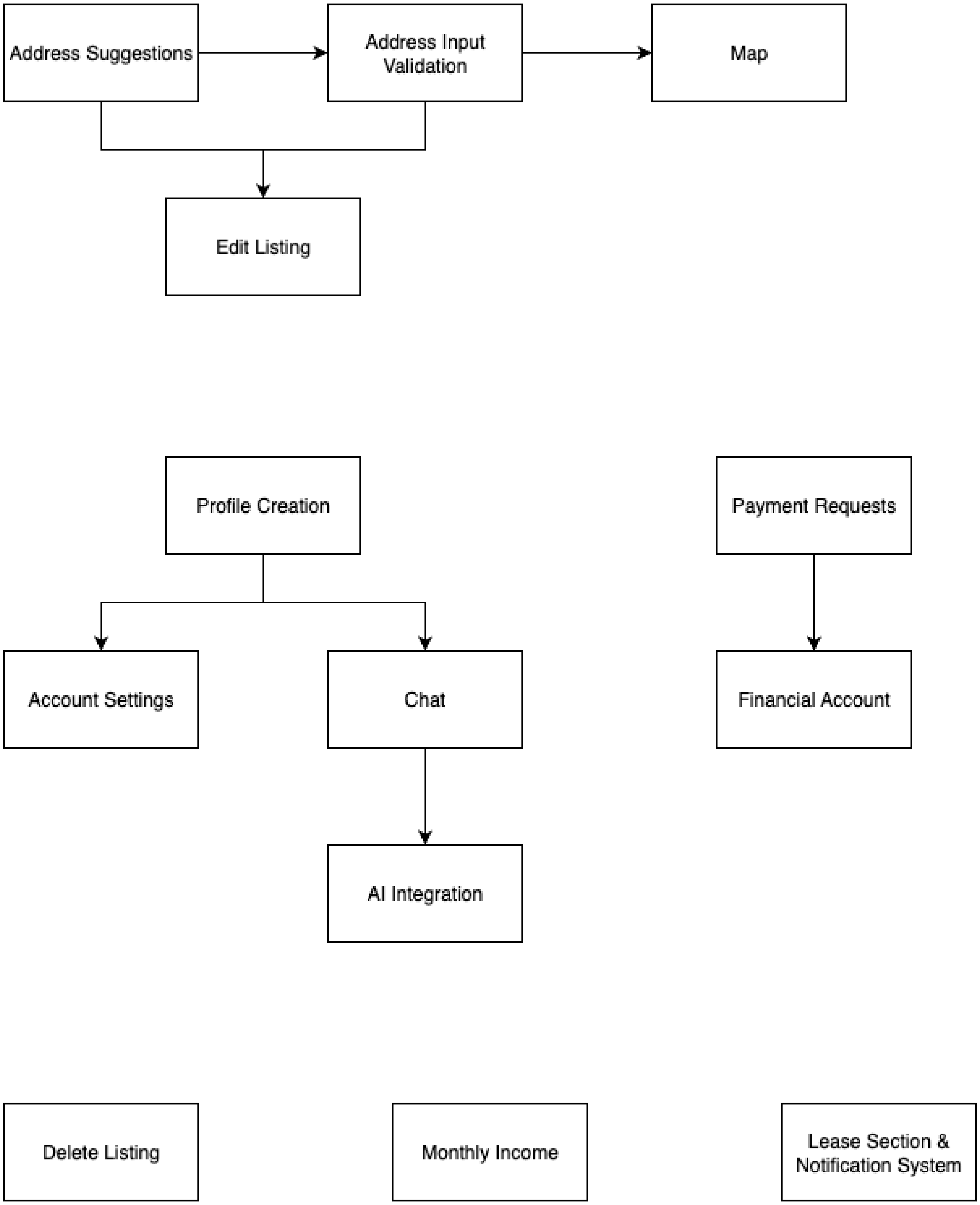
Schedule & Dependencies Analysis

Network Diagram



Map and AI Chat Integration were the main features of sprint 4. The remaining tasks were bug fixes and smaller features.

Dependencies

- Address & Listings
  - Address Input Validation depends on Address Suggestions
    - Map is dependent on Address Input Validation
  - Edit Listing is dependent on Address Suggestions and Address Input Validation
- Profile & Chat
  - Account Settings and Chat depend on Profile Creation
    - AI Integration is dependent on Chat
- Payments
  - Financial Account is dependent on Payment Requests
- Independent Tasks
  - Delete Listing, Monthly Income, and Lease Section & Notification System are independent of the other tasks

Critical Path

There are three equally long critical paths:

1. Address Suggestions → Address Input Validation → Map
2. Address Suggestions → Address Input Validation → Edit Listing
3. Profile Creation → Chat → AI Integration

Any delay in these steps will delay the overall project delivery.

Risk Areas

- Two dependencies of Address Suggestions and Address Input Validation:
  Edit Listing and Map, a key feature of this sprint, rely on the correct implementation of Address Suggestions and Address Input Validation. If addresses are invalid, both Edit Listing and Map will be delayed.
- Multiple critical paths:
  Having three separate critical paths slows down each path's development and increases the risk of delay.
- Key features at the end of their paths:
  The sprint's biggest features rely on the completion of two other tasks. Any issues with these tasks will hinder the completion of the more important features.
- Broad scope:
  While the critical paths are relatively short, there is a large number of total tasks that need to be completed, whether they have dependencies or not. The independent tasks are at risk of being overlooked, so it is crucial to plan and track all tasks carefully.

What Went Wrong & Lessons Learned

What Went Wrong

- Underestimation of feature complexity:
  Some features, especially the map feature, were overlooked and had to be reworked from the initial oversimplified implementation. This caused delays in the final delivery.
- Misunderstandings of features:

Group members had different interpretations of features, leading to changes in feature requirements and an unplanned spike in development near the end of the sprint.

- Delayed completion of key features:
  While independent and non-critical tasks were completed early, key features Map and AI Integration were delayed to the end of the sprint due to mid-sprint changes in requirements.

Lessons Learned

- Enforce frequent check-ins for all features:
  All team members should demonstrate their progress at each standup in addition to providing updates in order to prevent miscommunication and misunderstandings of features.
- Narrow-down sprint scope:
  Focus the sprint on the main deliverables. All team members should be involved in ensuring the completion and quality of the key features.
- Consider dependencies in sprint planning:
  Plan to complete all known elements of each feature in its assigned sprint, to reduce carryover into future sprints.