Story: Driv- 2
Sub Task: Driv-37
Determine search criteria

Story: Driv- 2
Sub Task: Driv-39
Develop backend logic to return relevant search results

Story: Driv- 4
Sub Task: Driv-46
Update the endpoint for accepting friend requests in server.js

Story: Driv- 4
Sub Task: Driv-48
Update Notifications.js to fetch and display incoming friend requests

Story: Driv- 2
Sub Task: Driv-38
Develop frontend user interface

Story: Driv- 3
Sub Task: Driv-33
Develop an endpoint for backend to handle the friend request submissions

Story: Driv- 4
Sub Task: Driv-47 Add an endpoint for declining friend requests in server.js

Story: Driv- 4
Sub Task: Driv-49
Modify backend logic for friend request handling.

Story: Driv- 3
Sub Task: Driv-34
Create validation checks for existing friend requests

Story: Driv- 3
Sub Task: Driv-35
Integrate authentication to ensure request

Story: Driv- 3
Sub Task: Driv-36
Create front end in search user list for "Add Friend"

Story: Driv- 11
Sub Task: Driv-43
Implement backend to search and find users with the same interests

Story: Driv- 11
Sub Task: Driv-44
Implement frontend to display suggested friends

Story: Driv- 14
Sub Task: Driv-40/41
Develop frontend user interface for users to see friends in a scrollable list format

Story: Driv- 11
Sub Task: Driv-45
Integrate Backend and Frontend

Legend

B → A

B depends on A

User Story Driv-2: Search for other students

1. **Driv-37: Determine search criteria**
    - **Reason:** This is the first need as we need to be able to search for users before adding them as friends.
2. **Driv-38 Subtask 2: Develop frontend user interface**
    - **Depends on Driv-37**
    - **Reason:** The UI design and functionality depend on knowing what the search criteria are. Without this, it's impossible to create an interface that allows users to input the correct search parameters.
3. **Driv-39: Develop backend logic to return relevant search results**
    - **Depends on Driv-37**
    - **Reason:** The backend needs to know what criteria it will be searching for in the database. The search criteria defined in Driv-2 Subtask 1 inform the queries and data retrieval logic.

User Story Driv-3: Send connection requests

1. **Driv-33: Develop an endpoint for backend to handle the friend request submissions**
    - **Depends on Driv-39**
    - **Reason:** The endpoint for submitting friend requests relies on the user data returned from the search functionality. The search results are needed to identify which user to send the request to.
2. **Driv-34: Create validation checks for existing friend requests**
    - **Depends on Driv-33**
    - **Reason:** Validation logic needs an endpoint to process and check requests against. Without the submission endpoint, there is no way to validate friend requests.
3. **Driv-35: Integrate authentication to ensure request**
    - **Depends on Driv-33**
    - **Reason:** Authentication is necessary to ensure that the user sending the friend request is authorized and legitimate. This task can be developed in parallel but is essential before Driv-3 Subtask 4 to ensure secure requests.
4. **Driv-36: Create front end in search user list for "Add Friend"**
    - **Depends on Driv-35**
    - **Reason:** The frontend interface to add friends must integrate with the search UI and the backend submission endpoint. It also needs validation checks to prevent duplicate requests.

User Story Driv-4: Accept or decline connection requests

1. **Driv-46: Update the endpoint for accepting friend requests in server.js**
    - **Depends on Driv-33**
    - **Reason:** To accept friend requests, the system must first have a way to submit and track these requests, which is provided by Driv-33.
2. **Driv-47: Add an endpoint for declining friend requests in server.js**
    - **Depends on Driv-33**
    - **Reason:** Similar to accepting requests, the ability to decline requests requires a functioning request submission system.
3. **Driv-48: Update Notifications.js to fetch and display incoming friend requests**
    - **Depends on Driv-46 and Driv-47**
    - **Reason:** Notifications for incoming requests rely on the endpoints for accepting and declining requests to determine the state of each request.
4. **Driv-49: Modify backend logic for friend request handling**
    - **Depends on Driv-46 and Driv-47**
    - **Reason:** Comprehensive backend logic is required to handle all possible states of a friend request (e.g., accepted, declined, pending), which builds on the endpoints created in the previous subtasks.

User Story Driv-11: Get connection recommendations

1. **Driv-44: Implement frontend to display suggested friends**
    - **Reason:** The frontend for displaying recommendations can be developed independently but needs backend support to fetch and display meaningful data.
2. **Driv-43: Implement backend to search and find users with the same interests**
    - **Depends on Driv-39**
    - **Reason:** Finding users with similar interests is a backend process that involves complex querying of user data, similar to the search functionality.
3. **Driv-45: Integrate Backend and Frontend**
    - **Depends on Driv-44 and Driv-43**
    - **Reason:** Integration is necessary to connect the data processed by the backend with the frontend interface. Both must be fully developed before they can be integrated.

User Story Driv-14: See a list of my connections

1. **Driv-40/41: Develop frontend user interface for users to see friends in a scrollable list format**
    - **Depends on Driv-33**
    - **Reason:** The UI for displaying friends relies on having the ability to accept friend requests since the friends list is derived from accepted requests.

Critical Path:

Driv-37: Determine search criteria

Driv-39: Develop backend logic to return relevant search results

Driv-33: Develop an endpoint for backend to handle the friend request submissions

Driv-34: Create validation checks for existing friend requests

Driv-35: Integrate authentication to ensure request

Driv-36: Create front end in search user list for "Add Friend"

Driv-46: Update the endpoint for accepting friend requests in server.js

Driv-47: Add an endpoint for declining friend requests in server.js

Driv-49: Modify backend logic for friend request handling

Driv-48: Update Notifications.js to fetch and display incoming friend requests

Driv-43: Implement backend to search and find users with the same interests

Driv-45: Integrate Backend and Frontend

Driv-40/41: Develop frontend user interface for users to see friends in a scrollable list format

Explain what you do to keep your sprint in schedule

We have ensured that the tasks that have other stories/subtasks depending on it, to be prioritize first and identifying these as these tasks affect the timeline for our Sprint 2, so when we focus on them we ensure we spend our time on those to minimize delays.

We also used frequent standups during our development cycle to help us review our progress and our blockers and to adjust story point priorities as we needed when we realized other tasks that are in the critical path. These meetings helped us focus on key updates to each other.

We also ensured during the initial planning of the user stories and sprint 2 in general that we broke down the user story into manageable tasks to work up to the completion. This helped with our initial estimation of the user story points and also gave us a good idea of the task dependencies.

We also tried to identify risks and then the following contingency plans.This lead us to effectively communicating, involving maintaining open communication of the development and encourage the team when they are stuck.We used instagram mostly.