**User Story: Community Connection on Shared Issues**

*“As a student, I want to connect with others who've shared similar feedback or faced similar issues, so we can collaborate on solutions or offer mutual support.”*

**Story Tasks**

1. **Research and Requirement Gathering**

* Conduct focus groups or surveys with students to gauge the desire and preferred methods of connecting with peers over shared issues.
* Examine existing platforms to understand how they facilitate community connections on shared topics.

1. **Design**

* Design an interface where students can view feedback or issues similar to theirs.
* Propose features that allow students to initiate discussions or chats based on shared feedback.
* Incorporate user profiles or anonymity features based on student preference.

1. **Backend Development**

* Structure a database to categorize and match similar feedback or issues.
* Develop API endpoints to fetch related feedback for a student and facilitate connections.

1. **Frontend Development**

* Implement the community connection interface per the approved design.
* Ensure students can smoothly navigate, view related feedback, and initiate discussions.

1. **Testing**

* Conduct unit tests for backend functionalities.
* Perform integration tests to confirm frontend and backend coherence.
* Gather a group of students for user acceptance testing of the community connection feature.

1. **Documentation**

* Create or update user guides detailing the community connection functionalities.

1. **Deployment**

* Outline a deployment plan for introducing the community connection feature.
* Monitor after deployment for issues and user feedback.

**Functional Requirements**

1. **Feedback/Issue Matching**

* The system must identify and categorize feedback or issues to match similar ones.
* Students should be presented with a list of feedback or issues akin to their own.

1. **Discussion Initiation**

* Upon identifying related feedback, students should be able to initiate or join discussions.
* The system must notify students when new discussions related to their feedback start.

1. **User Profiles & Anonymity**

* Based on preferences, students can choose to either showcase their profiles or remain anonymous in discussions.
* The system should ensure privacy settings are adhered to consistently.

1. **Search and Filters**

* Students must have the option to search for specific issues or feedback.
* Filters should be available for students to refine the list of feedback/issues based on different criteria (e.g., date, popularity, category).

**Non-functional Requirements**

1. **Usability**

* The community connection interface should offer a user-friendly experience, allowing for easy navigation and feedback exploration.

1. **Performance**

* Feedback matching and discussion initiation should occur promptly, avoiding unnecessary delays.

1. **Security**

* Personal data and feedback details must remain secure, especially if a student chooses to be anonymous.
* Proper authentication measures should be in place to safeguard student data.

1. **Scalability**

* As the student community grows, the system should scale to accommodate more feedback and connections without performance degradation.

1. **Reliability**

* The platform should offer consistent uptime, ensuring students can connect over shared issues whenever needed.

1. **Maintainability**

* The community connection feature should allow for updates or changes with minimal disruption.

1. **Accessibility**

* The interface should adhere to accessibility standards, ensuring all students, including those with disabilities, can utilize the feature.