YuchenPeng_a1824982
Retrospective Sprint <2> of Group <Path5>
Shize Liu_a1844323 Yuze Li_a1848890 Ruoyu Xiong_a1847649
Yuchen Peng_a1824982 Yuejun Zhao_a1829813 Shijie Zhang_a1809881

1. What went well in the sprint (Individually Written)?

During this sprint, one standout quality has been our team's quick responsiveness and strong cohesion. These traits enable us to adapt swiftly to changes and remain composed in tense situations. For example, because we didn't check Slack messages frequently, we initially missed some updated requirements. It took us a while to notice, but once we did, everyone responded promptly. No one panicked; instead, we supported each other and communicated calmly. We quickly arranged an urgent meeting the following day, discussed the changes, and made necessary adjustments. This rapid response within our group prevents us from falling further behind in our progress and ensures we catch up quickly. Our group's strong cohesion also allows us to communicate peacefully even when facing unfavourable circumstances. These actions help mitigate the negative impacts on the software development process when unexpected challenges arise.

2. What could be improved (Individually Written)?

One area that could be improved is the way we use GitHub issues. Currently, it seems a bit disorganised, and the tasks are not well structured. From my point of view, the issue titles are often unclear and lacking in detail. The descriptions in the issue bodies could also be more comprehensive. This problem creates challenges for team members in terms of understanding the tasks. The lack of clarity can lead to confusion, as team members may not have a clear idea of progress and what needs to be addressed. This can result in misunderstandings and slow down the development process. More importantly, it can leads to insufficient collaboration among team members.

3. What will the group commit to improve in the next sprint?

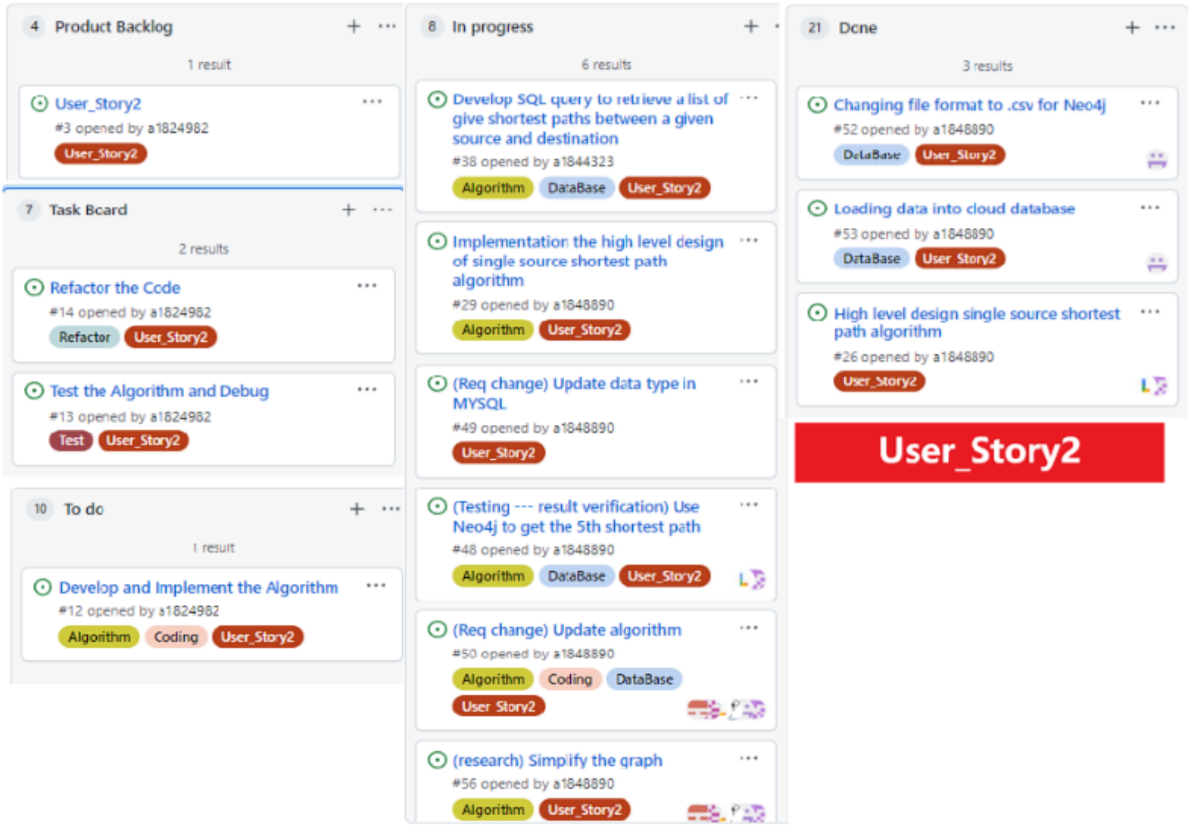
- More frequently check on Slack messages

In the upcoming sprint, we'll be increasing our frequency in checking Slack messages, ensuring that we receive updates in a timely manner. This approach is meant to improve the communication awareness with the product owner. By doing this, we'll be well-prepared to respond promptly to any new requirements that emerge, allowing us more time to carefully consider, discuss, and develop the right software solutions.

- Better organisation on GitHub issues

In the next sprint, we will fix the Github task boards. By providing clearer titles and more detailed issue descriptions will significantly enhance our software development process in terms of collaboration, communication and accountability. A well organised task board promotes a shared understanding of project goals which can reduce the likelihood of duplicated efforts or missed tasks and foster collaborations. Furthermore, detailed descriptions will make it easier to assign and track responsibilities. Each team members have a clear road map for their tasks, making it easier to meet deadlines and maintain accountability.

4. Comment on your progress this sprint (Individually Written)



4 Product Backlog
1 result
User_Story2
#3 opened by a1824982
User_Story2

7 Task Board
2 results
Refactor the Code
#14 opened by a1824982
Refactor User_Story2
Test the Algorithm and Debug
#13 opened by a1824982
Test User_Story2

10 To do
1 result
Develop and Implement the Algorithm
#12 opened by a1824982
Algorithm Coding User_Story2

8 In progress
6 results
Develop SQL query to retrieve a list of give shortest paths between a given source and destination
#38 opened by a1844323
Algorithm DataBase User_Story2
Implementation the high level design of single source shortest path algorithm
#29 opened by a1848890
Algorithm User_Story2
(Req change) Update data type in MYSQL
#49 opened by a1848890
User_Story2
(Testing --- result verification) Use Neo4j to get the 5th shortest path
#48 opened by a1848890
Algorithm DataBase User_Story2
(Req change) Update algorithm
#50 opened by a1848890
Algorithm Coding DataBase
User_Story2
(research) Simplify the graph
#56 opened by a1848890
Algorithm User_Story2

21 Done
3 results
Changing file format to .csv for Neo4j
#52 opened by a1848890
DataBase User_Story2
Loading data into cloud database
#53 opened by a1848890
DataBase User_Story2
High level design single source shortest path algorithm
#26 opened by a1848890
User_Story2

User_Story2

Written Snapshots

- Captured project milestones
- The complexity of this task is not high

Managed Github task board

- Managed progress updates
- Created new issues when changes/new requirements arise
- The complexity of this task is harder than I thought because it requires a deep understanding of the whole project. I need to have a complete roadmap of the project, considering many things such as technical issues, team member capabilities, etc. This is the part I did not do well, and I will improve in the next sprint

Organised meetings

- Contacted team members for attending meetings
- Our team responds quickly, so it is not difficult to organise a meeting

Researched how to generate the path diagram(graph)

- Researched methods to generate the graph rather than manually drawing it since the number of devices (nodes) and edges becomes very large
- Python and Neo4j are being taken into consideration
- It is a relatively challenging task because it takes a long time to learn how to automatically generate graphs using Python and Neo4j and also evaluate which method is more suitable for our situation.

I attended the sprint review/planning meeting on the 22nd of August, 5th of September with the tutor.

Snapshot Week <5> of Group <Path5>

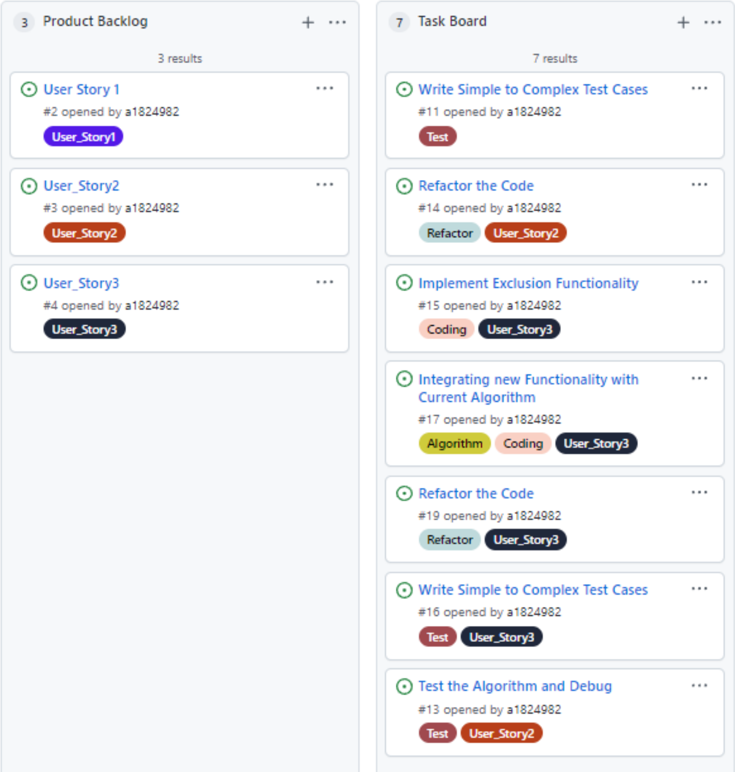
Project: ATSYS_Shortest Path Algorithm for Material Transportation

Members:

Shize Liu_a1844323
Yuze Li_a1848890
Ruoyu Xiong_a1847649
Yuchen Peng_a1824982
Yuejun Zhao_a1829813
Shijie Zhang_a1809881

Product Backlog and Task Board 3
Sprint Backlog and User Stories 5
Definition of Done 6
Summary of Changes 7

Product Backlog and Task Board



3 Product Backlog

3 results

User Story 1
#2 opened by a1824982
User_Story1

User_Story2
#3 opened by a1824982
User_Story2

User_Story3
#4 opened by a1824982
User_Story3

7 Task Board

7 results

Write Simple to Complex Test Cases
#11 opened by a1824982
Test

Refactor the Code
#14 opened by a1824982
Refactor User_Story2

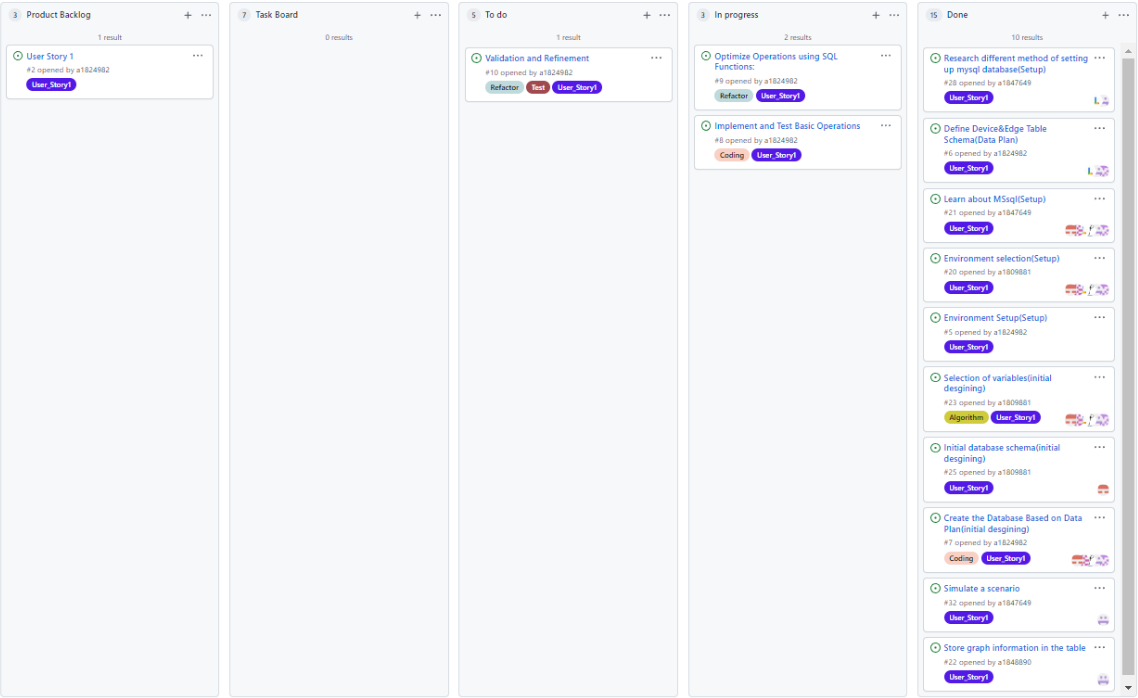
Implement Exclusion Functionality
#15 opened by a1824982
Coding User_Story3

Integrating new Functionality with Current Algorithm
#17 opened by a1824982
Algorithm Coding User_Story3

Refactor the Code
#19 opened by a1824982
Refactor User_Story3

Write Simple to Complex Test Cases
#16 opened by a1824982
Test User_Story3

Test the Algorithm and Debug
#13 opened by a1824982
Test User_Story2



3 Product Backlog

1 result

User Story 1
#2 opened by a1824982
User_Story1

7 Task Board

0 results

5 To do

1 result

Validation and Refinement
#10 opened by a1824982
Refactor Test User_Story1

3 In progress

2 results

Optimize Operations using SQL Functions:
#9 opened by a1824982
Refactor User_Story1

Implement and Test Basic Operations
#8 opened by a1824982
Coding User_Story1

15 Done

10 results

Research different method of setting up mysql database(Setup)
#28 opened by a1847649
User_Story1

Define Device&Edge Table Schema(Data Plan)
#6 opened by a1824982
User_Story1

Learn about MSsql(Setup)
#21 opened by a1847649
User_Story1

Environment selection(Setup)
#20 opened by a1809881
User_Story1

Environment Setup(Setup)
#5 opened by a1824982
User_Story1

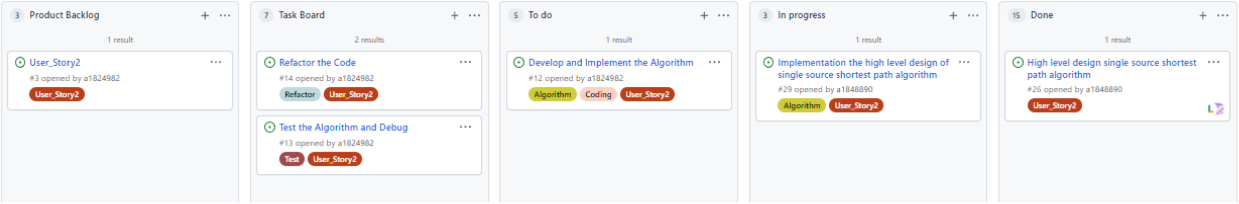
Selection of variables(initial desgining)
#23 opened by a1809881
Algorithm User_Story1

Initial database schema(initial desgining)
#25 opened by a1809881
User_Story1

Create the Database Based on Data Plan(initial desgining)
#7 opened by a1824982
Coding User_Story1

Simulate a scenario
#32 opened by a1847649
User_Story1

Store graph information in the table
#22 opened by a1848890
User_Story1



3 Product Backlog

1 result

User_Story2
#3 opened by a1824982
User_Story2

7 Task Board

2 results

Refactor the Code
#14 opened by a1824982
Refactor User_Story2

Test the Algorithm and Debug
#13 opened by a1824982
Test User_Story2

5 To do

1 result

Develop and Implement the Algorithm
#12 opened by a1824982
Algorithm Coding User_Story2

3 In progress

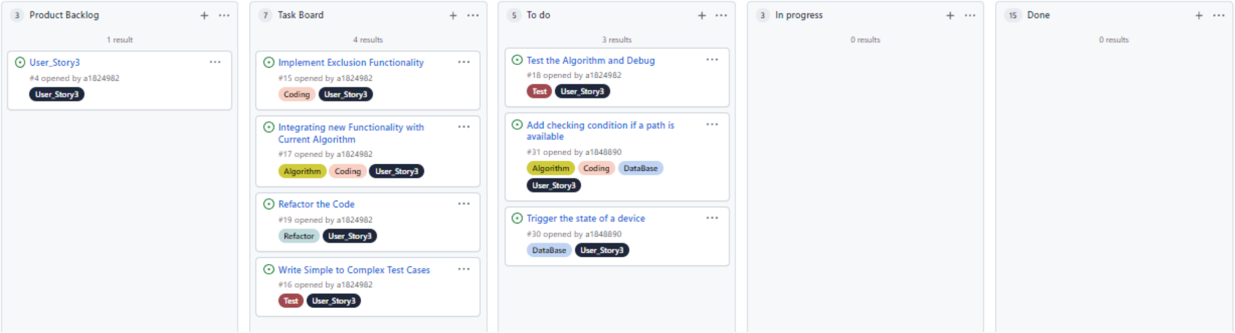
1 result

Implementation the high level design of single source shortest path algorithm
#29 opened by a1848890
Algorithm User_Story2

15 Done

1 result

High level design single source shortest path algorithm
#26 opened by a1848890
User_Story2



3 Product Backlog

1 result

User_Story3
#4 opened by a1824982
User_Story3

7 Task Board

4 results

Implement Exclusion Functionality
#15 opened by a1824982
Coding User_Story3

Integrating new Functionality with Current Algorithm
#17 opened by a1824982
Algorithm Coding User_Story3

Refactor the Code
#19 opened by a1824982
Refactor User_Story3

Write Simple to Complex Test Cases
#16 opened by a1824982
Test User_Story3

5 To do

3 results

Test the Algorithm and Debug
#18 opened by a1824982
Test User_Story3

Add checking condition if a path is available
#31 opened by a1848890
Algorithm Coding DataBase User_Story3

Trigger the state of a device
#30 opened by a1848890
DataBase User_Story3

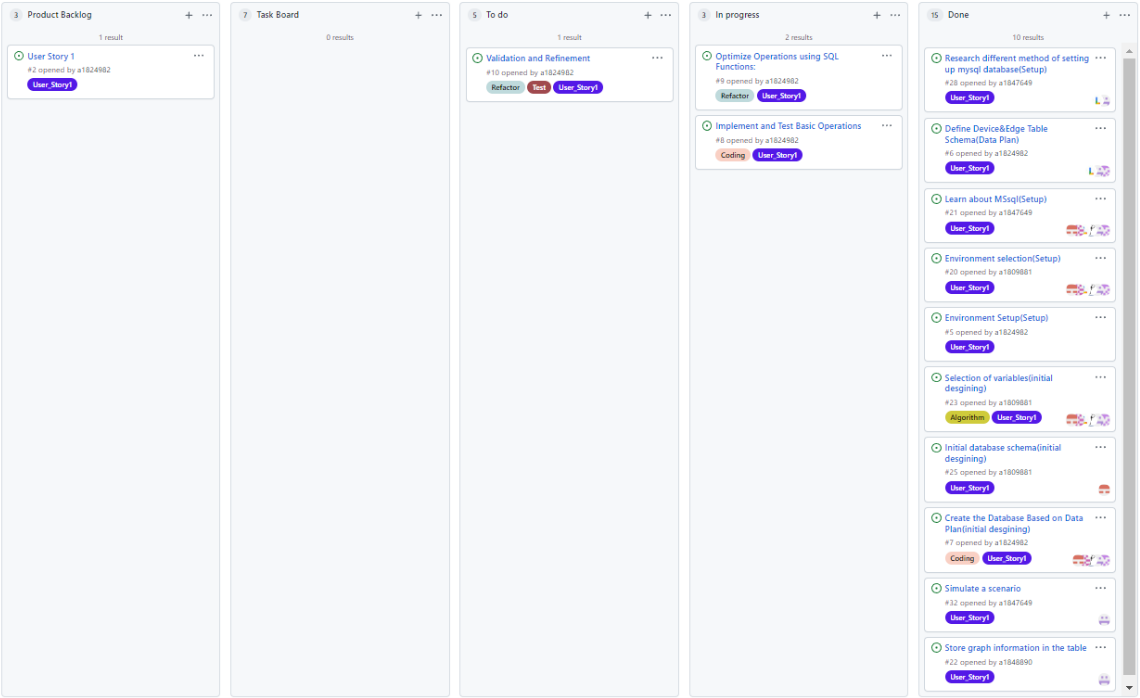
3 In progress

0 results

15 Done

0 results

Sprint Backlog and User Stories



3 Product Backlog

1 result

User Story 1
#2 opened by a1824982
User_Story1

7 Task Board

0 results

5 To do

1 result

Validation and Refinement
#10 opened by a1824982
Refactor Test User_Story1

3 In progress

2 results

Optimize Operations using SQL Functions:
#9 opened by a1824982
Refactor User_Story1

Implement and Test Basic Operations
#8 opened by a1824982
Coding User_Story1

15 Done

10 results

Research different method of setting up mysql database(Setup)
#28 opened by a1847649
User_Story1

Define Device&Edge Table Schema(Data Plan)
#6 opened by a1824982
User_Story1

Learn about MSsql(Setup)
#21 opened by a1847649
User_Story1

Environment selection(Setup)
#20 opened by a1809881
User_Story1

Environment Setup(Setup)
#5 opened by a1824982
User_Story1

Selection of variables(initial desgining)
#23 opened by a1809881
Algorithm User_Story1

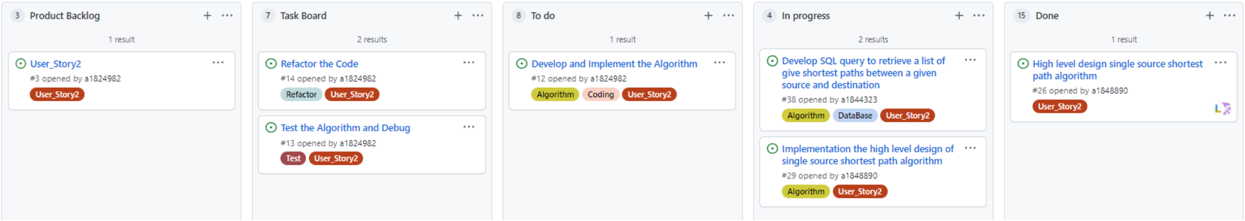
Initial database schema(initial desgining)
#25 opened by a1809881
User_Story1

Create the Database Based on Data Plan(initial desgining)
#7 opened by a1824982
Coding User_Story1

Simulate a scenario
#32 opened by a1847649
User_Story1

Store graph information in the table
#22 opened by a1848890
User_Story1

User Story(2): ***"As a user, I want to get the shortest path between 2 given devices so that material transportation will be efficient."***



3 Product Backlog

1 result

User_Story2
#3 opened by a1824982
User_Story2

7 Task Board

2 results

Refactor the Code
#14 opened by a1824982
Refactor User_Story2

Test the Algorithm and Debug
#13 opened by a1824982
Test User_Story2

8 To do

1 result

Develop and Implement the Algorithm
#12 opened by a1824982
Algorithm Coding User_Story2

4 In progress

2 results

Develop SQL query to retrieve a list of give shortest paths between a given source and destination
#38 opened by a1844323
Algorithm DataBase User_Story2

Implementation the high level design of single source shortest path algorithm
#29 opened by a1848890
Algorithm User_Story2

15 Done

1 result

High level design single source shortest path algorithm
#26 opened by a1848890
User_Story2

https://github.cs.adelaide.edu.au/SEP23S2PATH/PATH_5/projects/1?card_filter_query=label%3A+label%3Auser_story2

In this user story, users are requesting a method to determine the top 5 efficient routes between two specified devices. This feature will assist them in effectively and promptly managing their plants resulting in increased productivity and cost efficiency.

To fulfill this requirement we have implemented an algorithm that utilizes a MySQL database. The algorithm calculates the cost of the path between the selected devices. It uses recursion to ensure that every device is visited and explores all routes before sorting them in ascending order. The algorithm looks at where the devices are, if devices are in use(not available) or not, and other stuff to find the most cost efficient way.

The user specifies the starting and ending node and our implemented algorithm in MySQL returns the 5 paths that're cost efficient. Eventually this application could provide users with the ability to manage devices and plants in a cost manner.

Definition of Done

- A coding task is considered complete when the code has been written in accordance with the coding standards outlined in the report reviewed, tested (both unit and integration) refactored as needed, successfully passed peer review and obtained approval from all members of the team.
- A non-coding task assignment is considered complete when it has been reviewed, discussed, documented and agreed upon by the team in a meeting to ensure everyone is on the page. Additionally any specific problems should be reported in detail using our project page, on Github.

Summary of Changes

Github task board wise:
We reviewed the existing user stories. Included additional detailed tasks, in the “To do” section. We also went through and added more specific tasks. Additionally we completed some tasks in the “In progress” and moved them to the "Done" section.

Code wise:

- We plan to implement the shortest path algorithm in SQL, the algorithm is expected to return the shortest path between the start Node and the destination Node. In the coming sprints we plan to combine the SQL recursive CTE method and this algorithm to implement the shortest path algorithm that returns the top-5 shortest path.

- We plan to implement the test for user stories 2. This involves creating graphs, designing tables and conducting algorithm testing.

Snapshot Week <6> of Group <Path5>

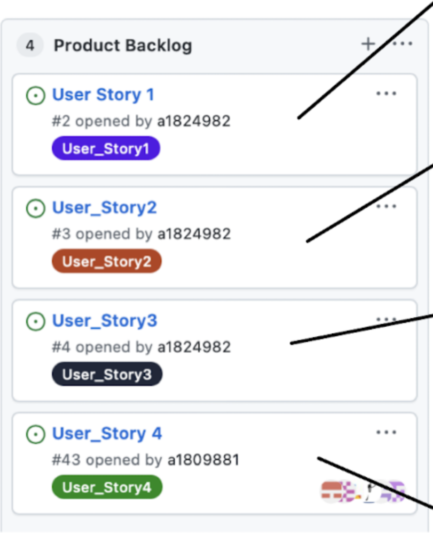
Project: ATSYS_Shortest Path Algorithm for Material Transportation

Members:

Shize Liu_a1844323
Yuze Li_a1848890
Ruoyu Xiong_a1847649
Yuchen Peng_a1824982
Yuejun Zhao_a1829813
Shijie Zhang_a1809881

Product Backlog and Task Board 3
Sprint Backlog and User Stories 6
Definition of Done 7
Summary of Changes 8

Product Backlog and Task Board



4 Product Backlog

User Story 1
#2 opened by a1824982
User_Story1

User_Story2
#3 opened by a1824982
User_Story2

User_Story3
#4 opened by a1824982
User_Story3

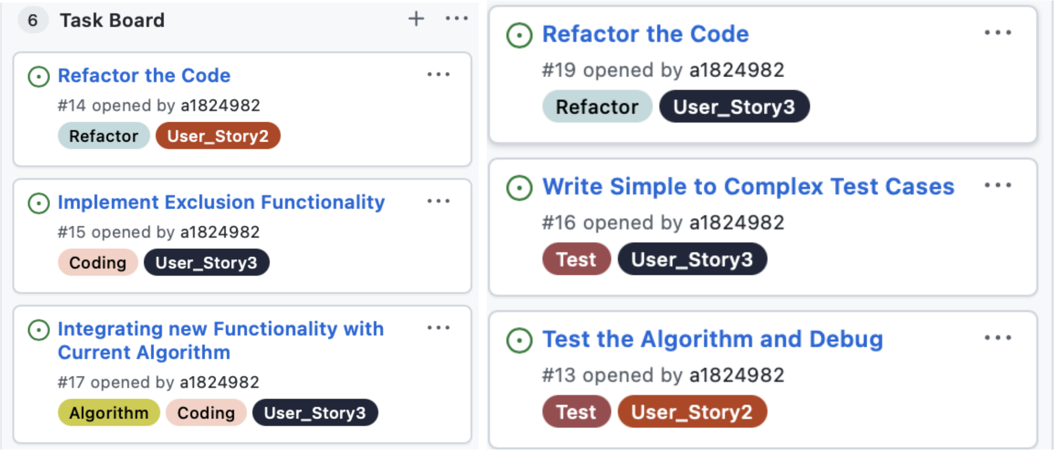
User_Story 4
#43 opened by a1809881
User_Story4

As a user, I want to store devices in the system so that I can add/remove/update them.

As a user, I want to get the shortest path between 2 given devices so that material transportation will be efficient.

As a user I want to mark devices to exclude, so that shortest paths can be identified avoiding them.

As a system administrator, I need the capability to dynamically redefine and manage devices within the system. This includes specifying each device's role as a source or destination, modifying their device details,. This functionality ensures that the system accurately reflects the real-time physical layout of devices and their interconnections, allowing for seamless adaptability to changes in the setup.



6 Task Board

Refactor the Code
#14 opened by a1824982
Refactor User_Story2

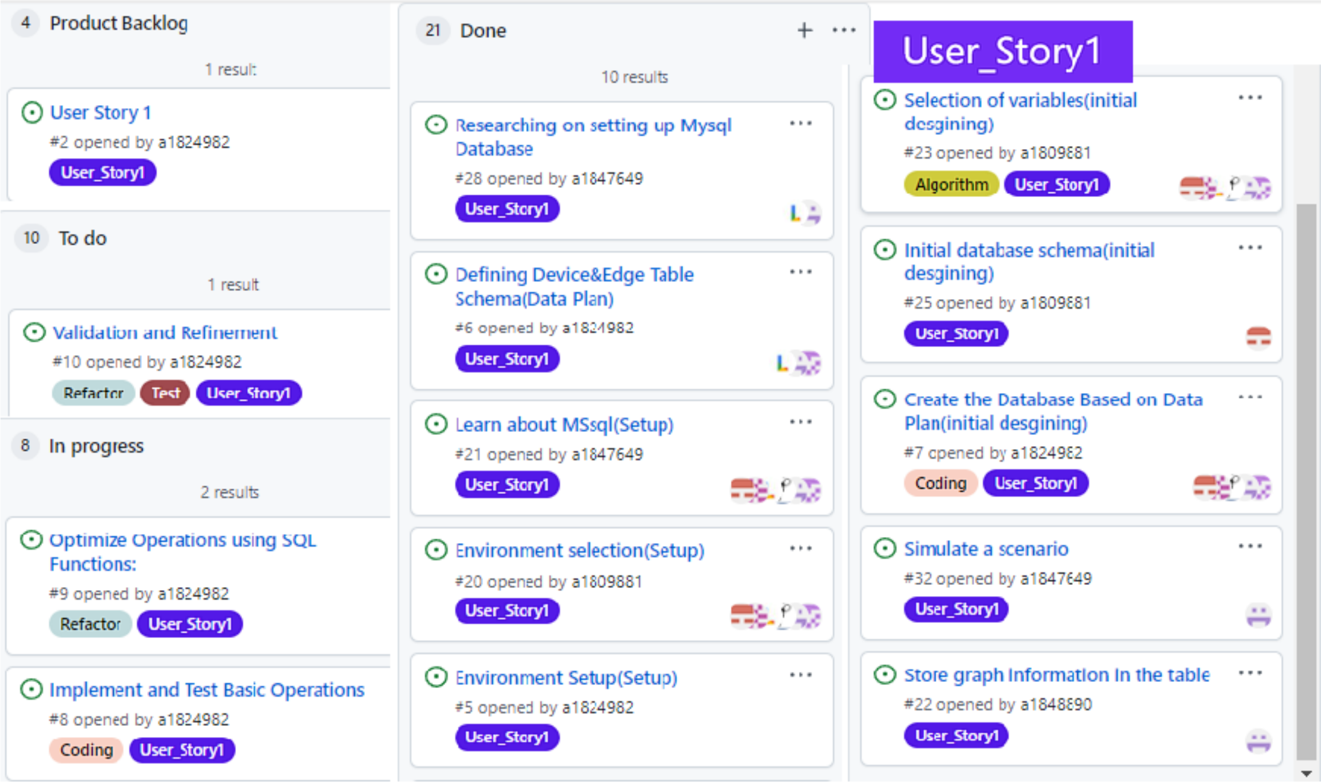
Implement Exclusion Functionality
#15 opened by a1824982
Coding User_Story3

Integrating new Functionality with Current Algorithm
#17 opened by a1824982
Algorithm Coding User_Story3

Refactor the Code
#19 opened by a1824982
Refactor User_Story3

Write Simple to Complex Test Cases
#16 opened by a1824982
Test User_Story3

Test the Algorithm and Debug
#13 opened by a1824982
Test User_Story2



User_Story1

4 Product Backlog

1 result

User Story 1
#2 opened by a1824982
User_Story1

10 To do

1 result

Validation and Refinement
#10 opened by a1824982
Refactor Test User_Story1

8 In progress

2 results

Optimize Operations using SQL Functions:
#9 opened by a1824982
Refactor User_Story1

Implement and Test Basic Operations
#8 opened by a1824982
Coding User_Story1

21 Done

10 results

Researching on setting up Mysql Database
#28 opened by a1847649
User_Story1

Defining Device&Edge Table Schema(Data Plan)
#6 opened by a1824982
User_Story1

Learn about MSsql(Setup)
#21 opened by a1847649
User_Story1

Environment selection(Setup)
#20 opened by a1809881
User_Story1

Environment Setup(Setup)
#5 opened by a1824982
User_Story1

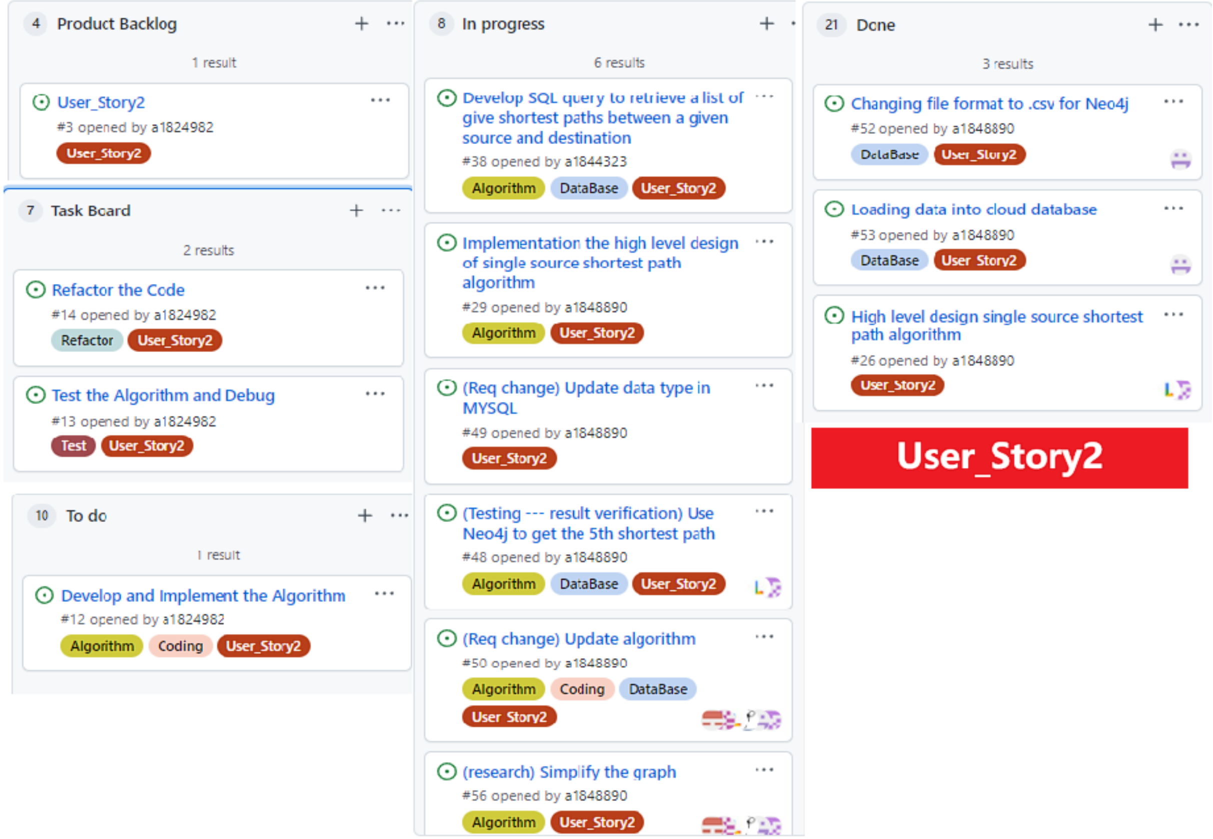
Selection of variables(initial desgining)
#23 opened by a1809881
Algorithm User_Story1

Initial database schema(initial desgining)
#25 opened by a1809881
User_Story1

Create the Database Based on Data Plan(initial desgining)
#7 opened by a1824982
Coding User_Story1

Simulate a scenario
#32 opened by a1847649
User_Story1

Store graph Information In the table
#22 opened by a1848890
User_Story1



4 Product Backlog

1 result

User_Story2
#3 opened by a1824982
User_Story2

7 Task Board

2 results

Refactor the Code
#14 opened by a1824982
Refactor User_Story2

Test the Algorithm and Debug
#13 opened by a1824982
Test User_Story2

10 To do

1 result

Develop and Implement the Algorithm
#12 opened by a1824982
Algorithm Coding User_Story2

8 In progress

6 results

Develop SQL query to retrieve a list of give shortest paths between a given source and destination
#38 opened by a1844323
Algorithm DataBase User_Story2

Implementation the high level design of single source shortest path algorithm
#29 opened by a1848890
Algorithm User_Story2

(Req change) Update data type in MYSQL
#49 opened by a1848890
User_Story2

(Testing --- result verification) Use Neo4j to get the 5th shortest path
#48 opened by a1848890
Algorithm DataBase User_Story2

(Req change) Update algorithm
#50 opened by a1848890
Algorithm Coding DataBase
User_Story2

(research) Simplify the graph
#56 opened by a1848890
Algorithm User_Story2

21 Done

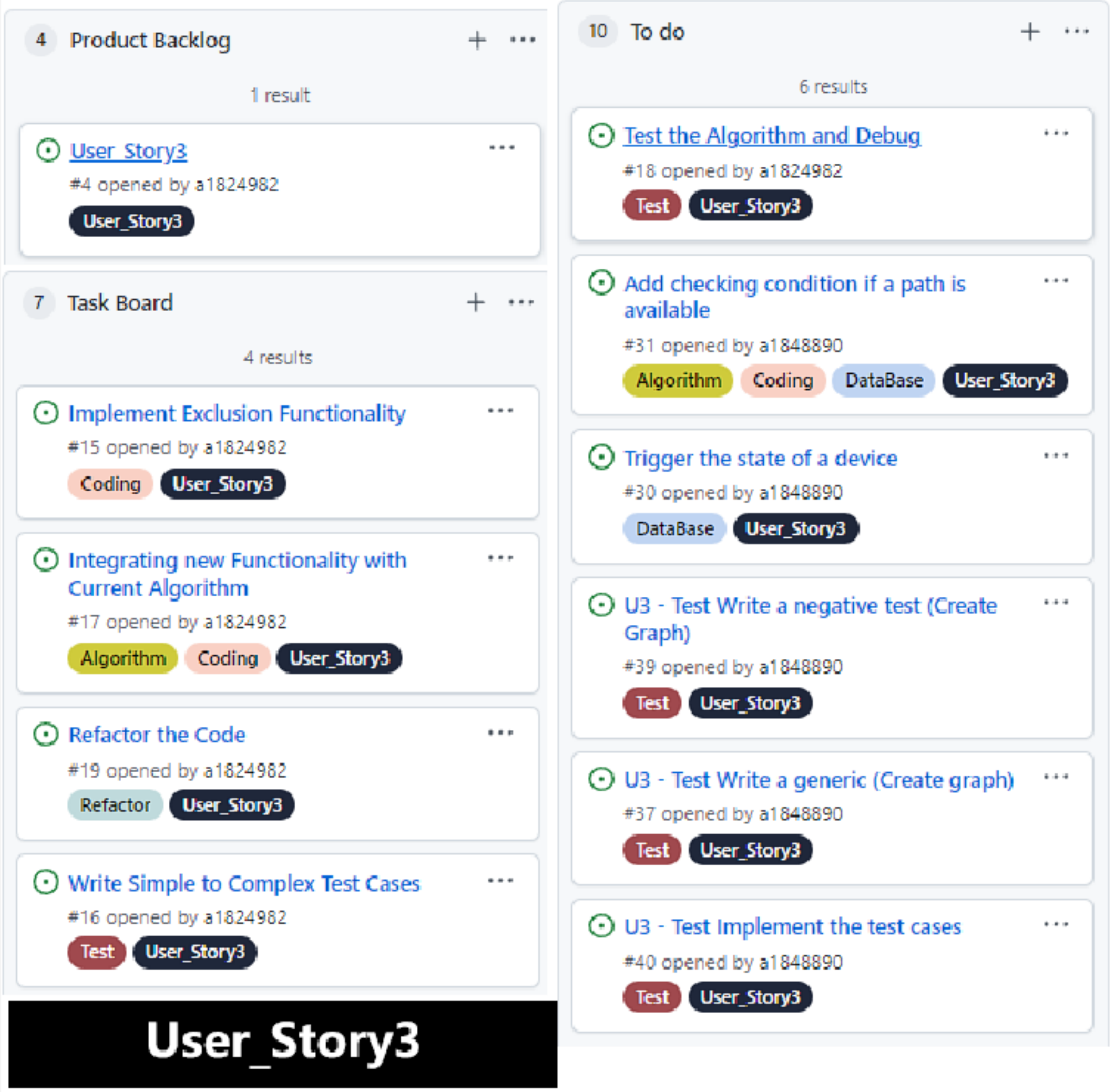
3 results

Changing file format to .csv for Neo4j
#52 opened by a1848890
DataBase User_Story2

Loading data into cloud database
#53 opened by a1848890
DataBase User_Story2

High level design single source shortest path algorithm
#26 opened by a1848890
User_Story2

User_Story2



4 Product Backlog

1 result

User_Story3
#4 opened by a1824982
User_Story3

7 Task Board

4 results

Implement Exclusion Functionality
#15 opened by a1824982
Coding User_Story3

Integrating new Functionality with Current Algorithm
#17 opened by a1824982
Algorithm Coding User_Story3

Refactor the Code
#19 opened by a1824982
Refactor User_Story3

Write Simple to Complex Test Cases
#16 opened by a1824982
Test User_Story3

10 To do

6 results

Test the Algorithm and Debug
#18 opened by a1824982
Test User_Story3

Add checking condition if a path is available
#31 opened by a1848890
Algorithm Coding DataBase User_Story3

Trigger the state of a device
#30 opened by a1848890
DataBase User_Story3

U3 - Test Write a negative test (Create Graph)
#39 opened by a1848890
Test User_Story3

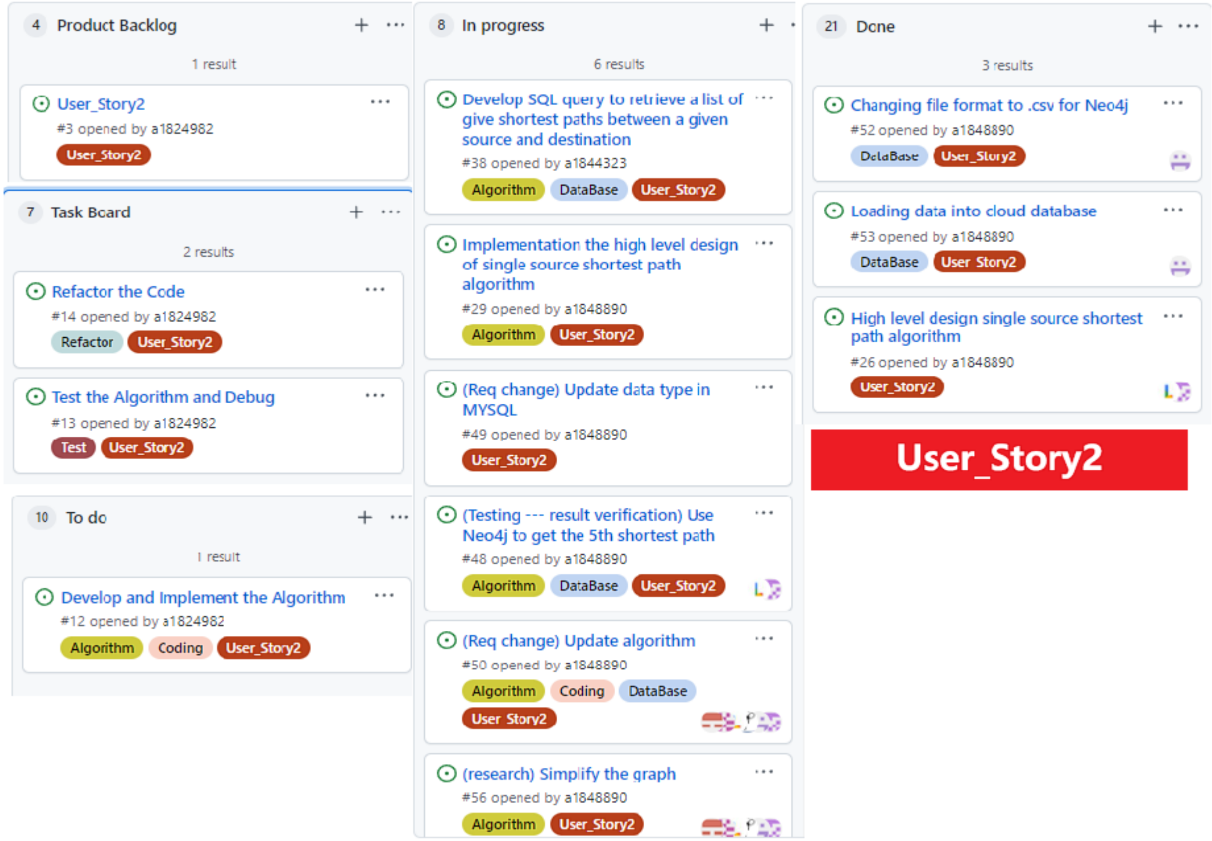
U3 - Test Write a generic (Create graph)
#37 opened by a1848890
Test User_Story3

U3 - Test Implement the test cases
#40 opened by a1848890
Test User_Story3

User_Story3

Sprint Backlog and User Stories

User Story(2): ***"As a user, I want to get the shortest path between 2 given devices so that material transportation will be efficient."***



4 Product Backlog

1 result

User_Story2
#3 opened by a1824982
User_Story2

7 Task Board

2 results

Refactor the Code
#14 opened by a1824982
Refactor User_Story2

Test the Algorithm and Debug
#13 opened by a1824982
Test User_Story2

10 To do

1 result

Develop and Implement the Algorithm
#12 opened by a1824982
Algorithm Coding User_Story2

8 In progress

6 results

Develop SQL query to retrieve a list of give shortest paths between a given source and destination
#38 opened by a1844323
Algorithm DataBase User_Story2

Implementation the high level design of single source shortest path algorithm
#29 opened by a1848890
Algorithm User_Story2

(Req change) Update data type in MYSQL
#49 opened by a1848890
User_Story2

(Testing --- result verification) Use Neo4j to get the 5th shortest path
#48 opened by a1848890
Algorithm DataBase User_Story2

(Req change) Update algorithm
#50 opened by a1848890
Algorithm Coding DataBase
User_Story2

(research) Simplify the graph
#56 opened by a1848890
Algorithm User_Story2

21 Done

3 results

Changing file format to .csv for Neo4j
#52 opened by a1848890
DataBase User_Story2

Loading data into cloud database
#53 opened by a1848890
DataBase User_Story2

High level design single source shortest path algorithm
#26 opened by a1848890
User_Story2

User_Story2

https://github.cs.adelaide.edu.au/SEP23S2PATH/PATH_5/projects/1?card_filter_query=label%3A+label%3Auser_story2

In this user story, users are requesting a method to determine the top 5 efficient routes between two specified devices. This feature will assist them in effectively and promptly managing their plants resulting in increased productivity and cost efficiency.

To fulfill this requirement we have implemented an algorithm that utilizes a MySQL database. The algorithm calculates the cost of the path between the selected devices. It uses recursion to ensure that every device is visited and explores all routes before sorting them in ascending order. The algorithm looks at where the devices are, if devices are in use(not available) or not, and other stuff to find the most cost efficient way.

The user specifies the starting and ending node and our implemented algorithm in MySQL returns the 5 paths that're cost efficient. Eventually this application could provide users with the ability to manage devices and plants in a cost manner.

Definition of Done

- A coding task is considered complete when the code has been written in accordance with the coding standards outlined in the report reviewed, tested (both unit and integration) refactored as needed, successfully passed peer review and obtained approval from all members of the team.
- A non-coding task assignment is considered complete when it has been reviewed, discussed, documented and agreed upon by the team in a meeting to ensure everyone is on the page. Additionally any specific problems should be reported in detail using our project page, on Github.

Summary of Changes

Github task board wise:

Adding the new user_story4 to fulfill the task board:

> *As a system administrator, I need the capability to dynamically redefine and manage devices within the system. This includes specifying each device's role as a source or destination, modifying their device details,. This functionality ensures that the system accurately reflects the real-time physical layout of devices and their interconnections, allowing for seamless adaptability to changes in the setup.*

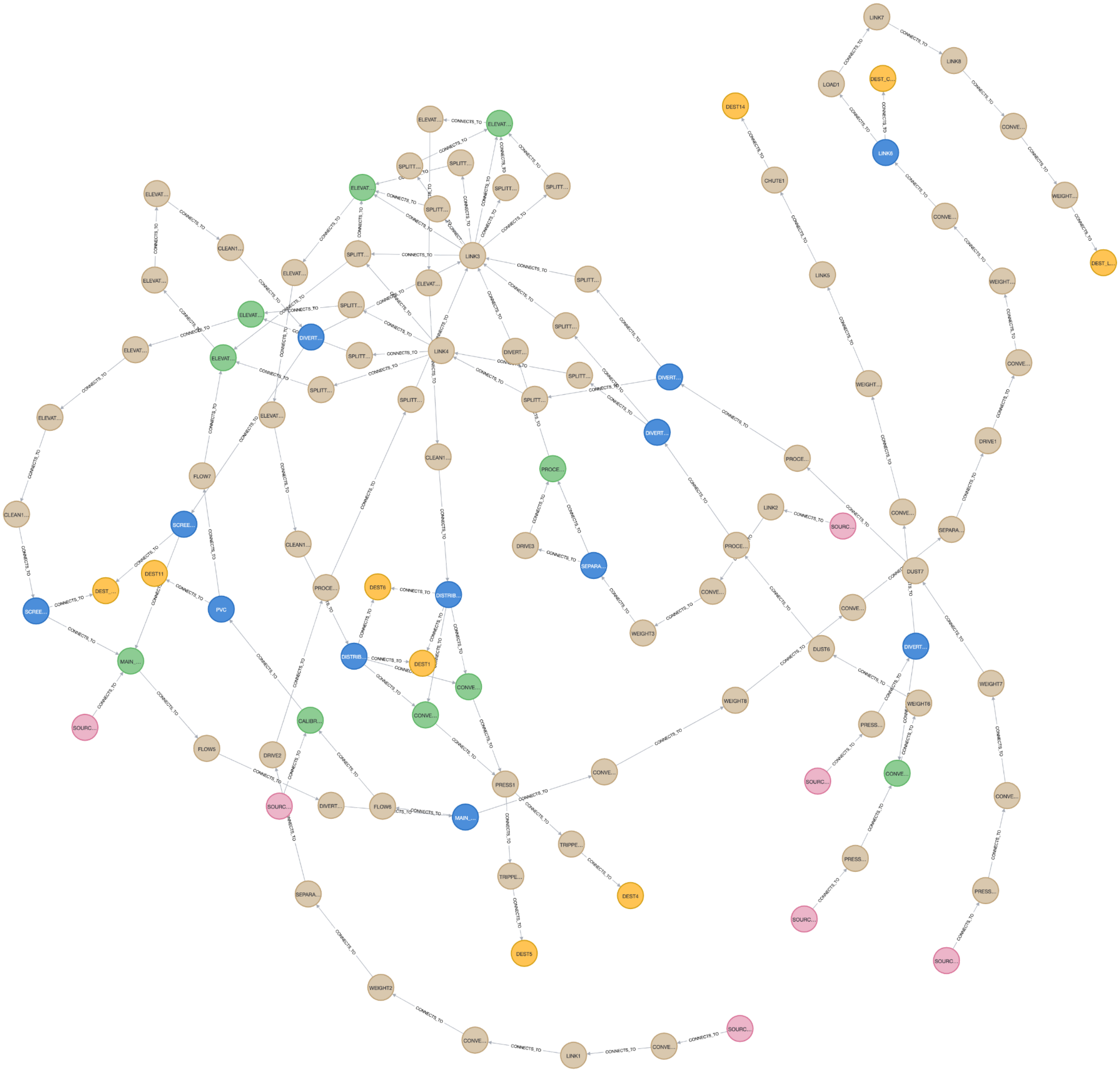
We involve the system administrator to grant them the capability of making modifications, within the system and updating data information. The administrator possesses privileges compared to users, which impacts both the security aspect and the implementation aspect of the coding process.

New Given Data:

The coding team took the data from(Device list.xlsx) the Excel table provided by our tutor. To visualize the data it was exported in CSV format. After consideration the coding team decided to use either Neo4j or Python for visualization purposes. Ultimately they opted for Neo4j as their preferred choice. Neo4j is more for node edges and python is more suited to general data. Neo4j uses a graph database model, where data is represented as nodes (entities) and relationships (connections between entities). It is effective for handling the data from the tutor.

Code wise:

- Adding a logical assessment to determine the type of devices because the device list provided does not indicate the type of devices, we need to figure it out when calculating the shortest path.
- Attempting to use python scripts to illustrate path diagrams but the performance is awful because data we have does not contain x-y coordinates for devices and python is not our best choice in this situation.
- We also added some code related to neo4j to display the entire path graph and test the built-in function of neo4j for finding the shortest path.


LINK7
LINK8
LOAD1
DEST_C...
DEST14
CONVE...
LINK6
CHUTE1
WEIGHT...
DEST_L...
CONVE...
ELEVAT...
ELEVAT...
SPLITT...
SPLITT...
SPLITT...
SPLITT...
ELEVAT...
SPLITT...
ELEVAT...
CLEAN1...
LINK3
ELEVAT...
SPLITT...
LINK5
WEIGHT...
ELEVAT...
ELEVAT...
SPLITT...
SPLITT...
DIVERT...
SPLITT...
LINK4
DIVERT...
ELEVAT...
ELEVAT...
SPLITT...
SPLITT...
DIVERT...
WEIGHT...
CONVE...
SPLITT...
SPLITT...
ELEVAT...
DIVERT...
DRIVE1
ELEVAT...
CLEAN1...
PROCE...
FLOW7
PROCE...
LINK2
CONVE...
CLEAN1...
SOURC...
SCREE...
SEPARA...
DRIVE3
CLEAN1...
PROCE...
SEPARA...
DEST11
DEST_...
DEST6
DISTRIB...
DUST7
SCREE...
PVC
CONVE...
CONVE...
WEIGHT3
DISTRIB...
DEST1
DUST6
DIVERT...
MAIN_...
CONVE...
WEIGHT7
WEIGHT8
CONVE...
CALIBR...
PRESS...
WEIGHT6
SOURC...
FLOW5
DRIVE2
CONVE...
SOURC...
CONVE...
PRESS1
SOURC...
DIVERT...
FLOW6
MAIN_...
TRIPPE...
PRESS...
CONVE...
TRIPPE...
DEST4
PRESS...
SEPARA...
SOURC...
DEST5
SOURC...
WEIGHT2
SOURC...
CONVE...
LINK1
CONVE...
CONNECTS_TO