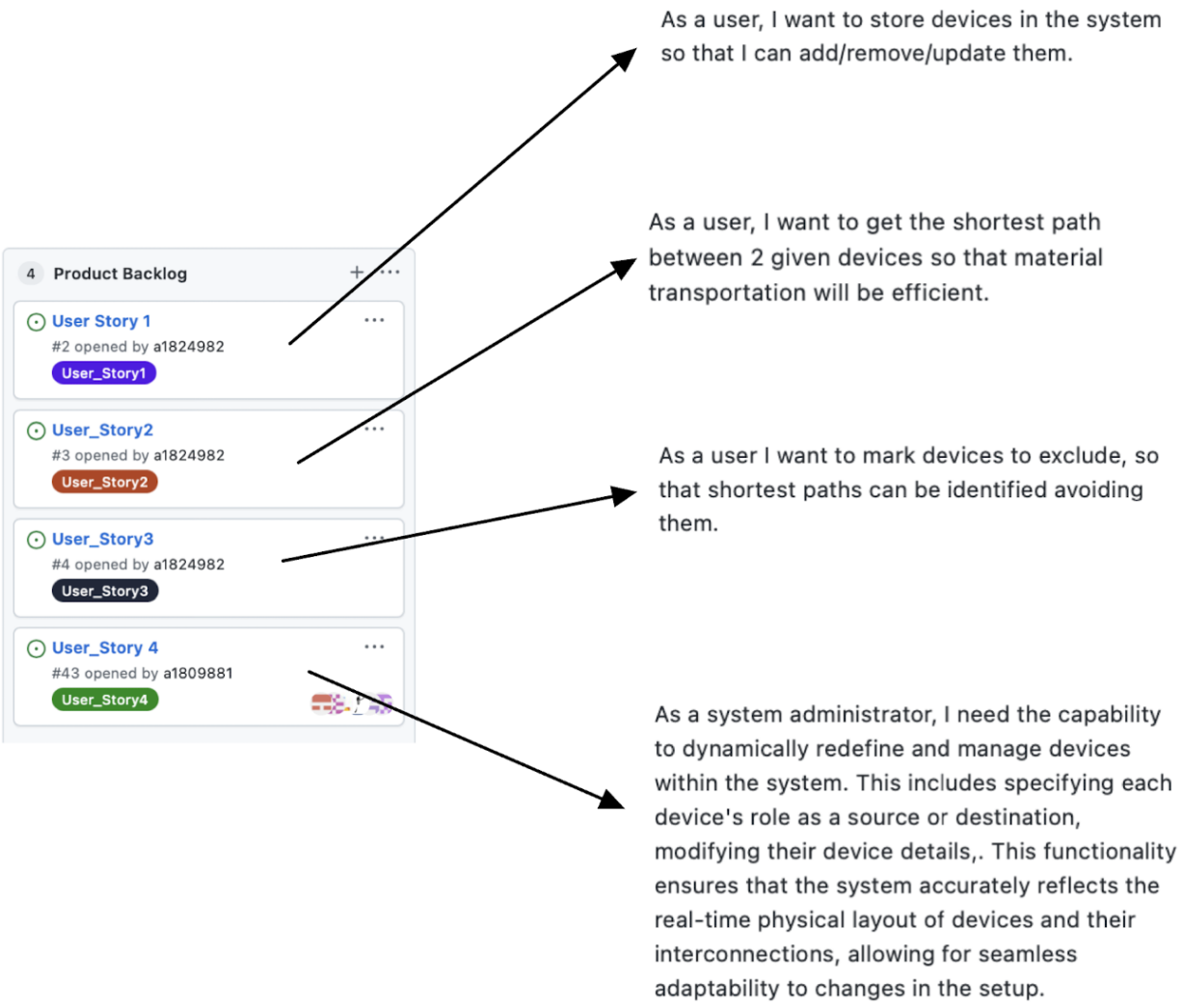
Snapshot Week <6> of Group <Path5>

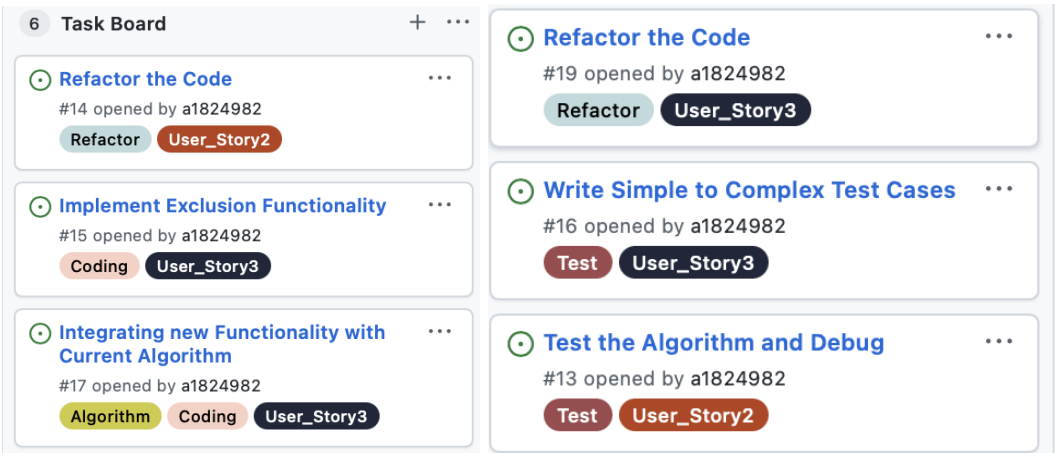
Project: ATSYS_Shortest Path Algorithm for Material Transportation

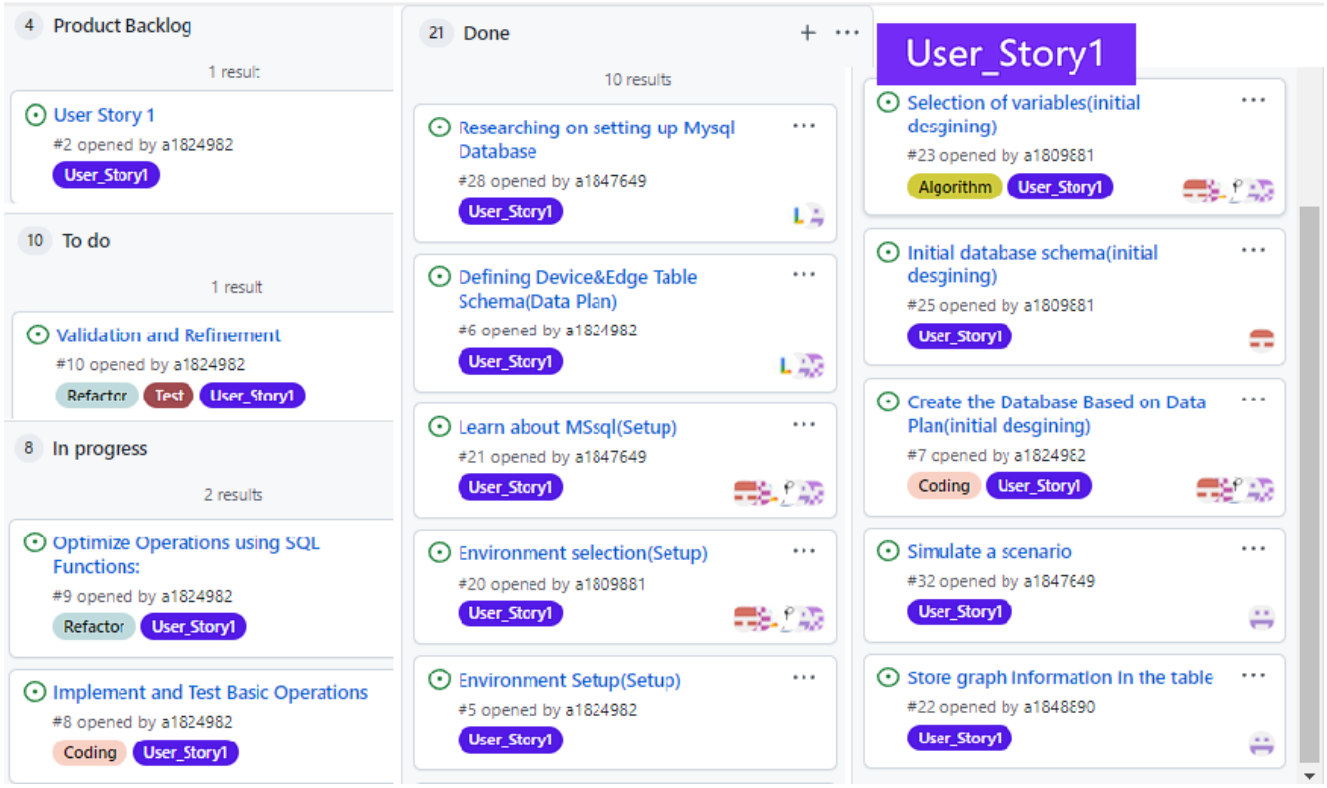
Members:

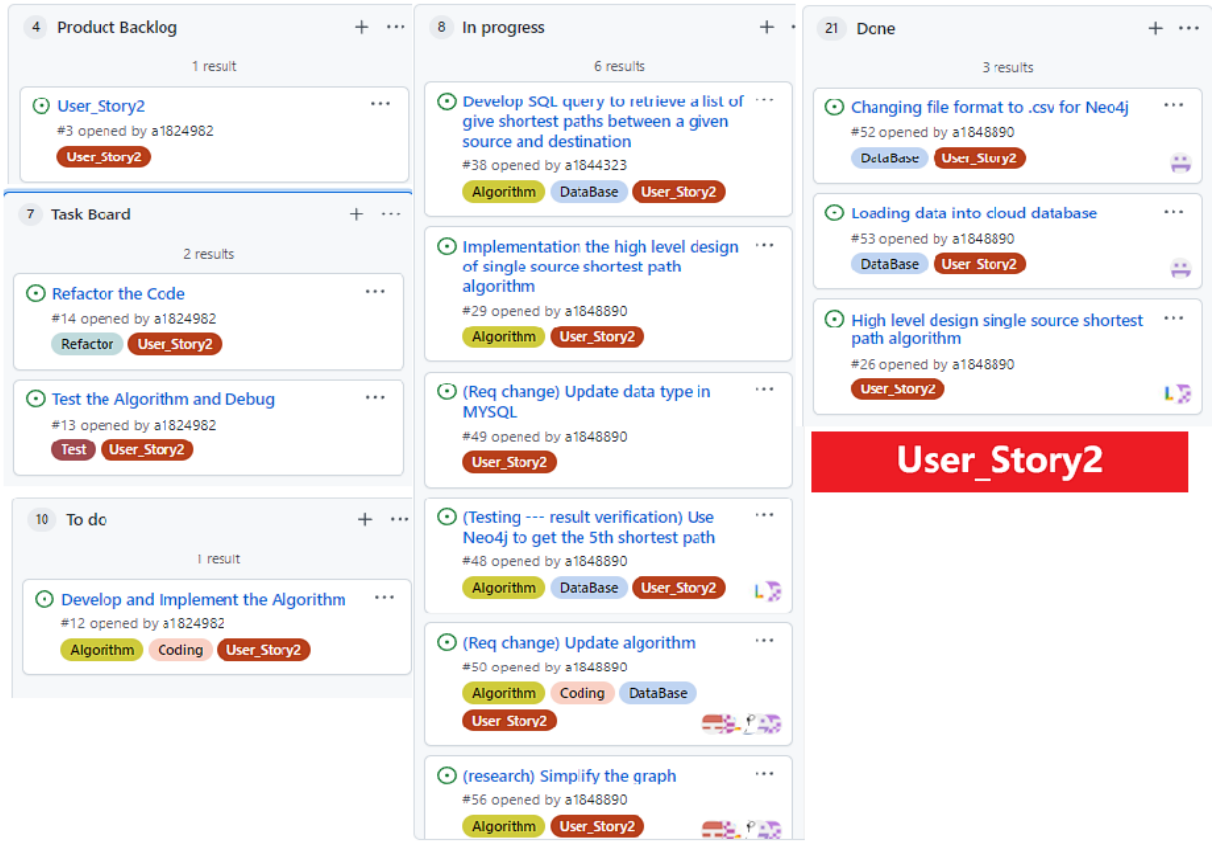
Shize Liu_a1844323
Yuze Li_a1848890
Ruoyu Xiong_a1847649
Yuchen Peng_a1824982
Yuejun Zhao_a1829813
Shijie Zhang_a1809881

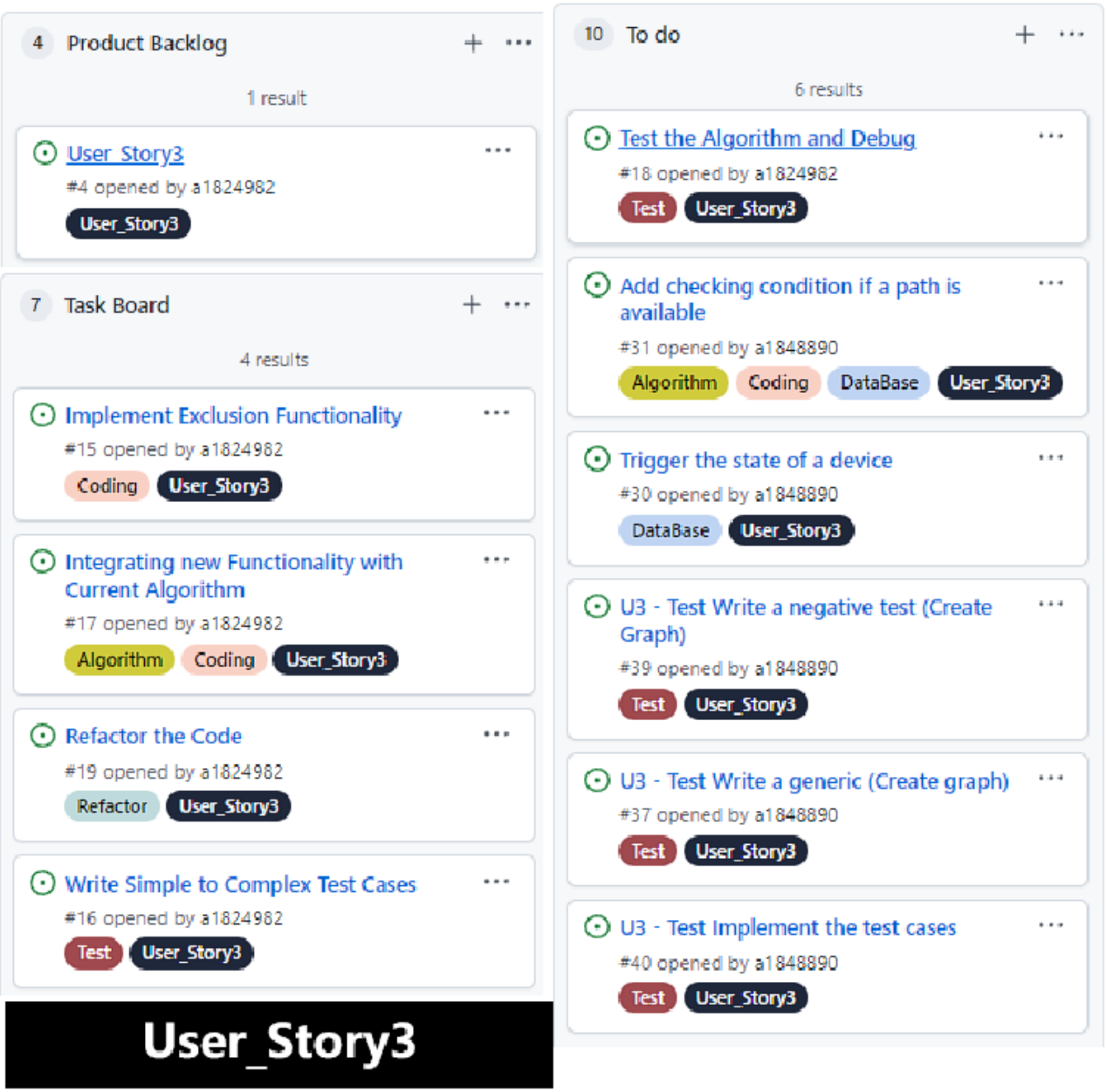
Product Backlog and Task Board





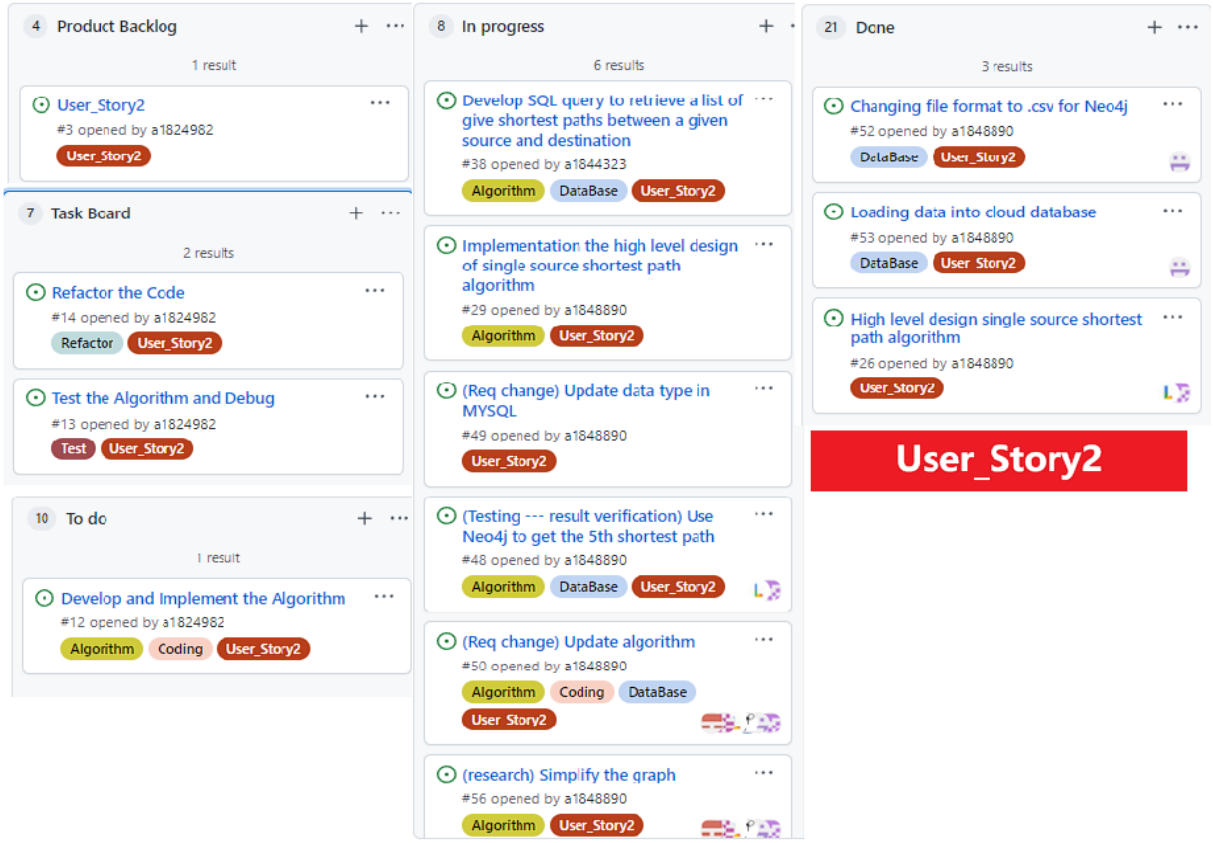

4 Product Backlog
1 result
User Story 1
#2 opened by a1824982
User_Story1
10 To do
1 result
Validation and Refinement
#10 opened by a1824982
Refactor
Test
User_Story1
8 In progress
2 results
Optimize Operations using SQL Functions:
#9 opened by a1824982
Refactor
User_Story1
Implement and Test Basic Operations
#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
User_Story1
Selection of variables(initial desgining)
#23 opened by a1809681
Algorithm
User_Story1
Initial database schema(initial desgining)
#25 opened by a1809681
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."***



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.*

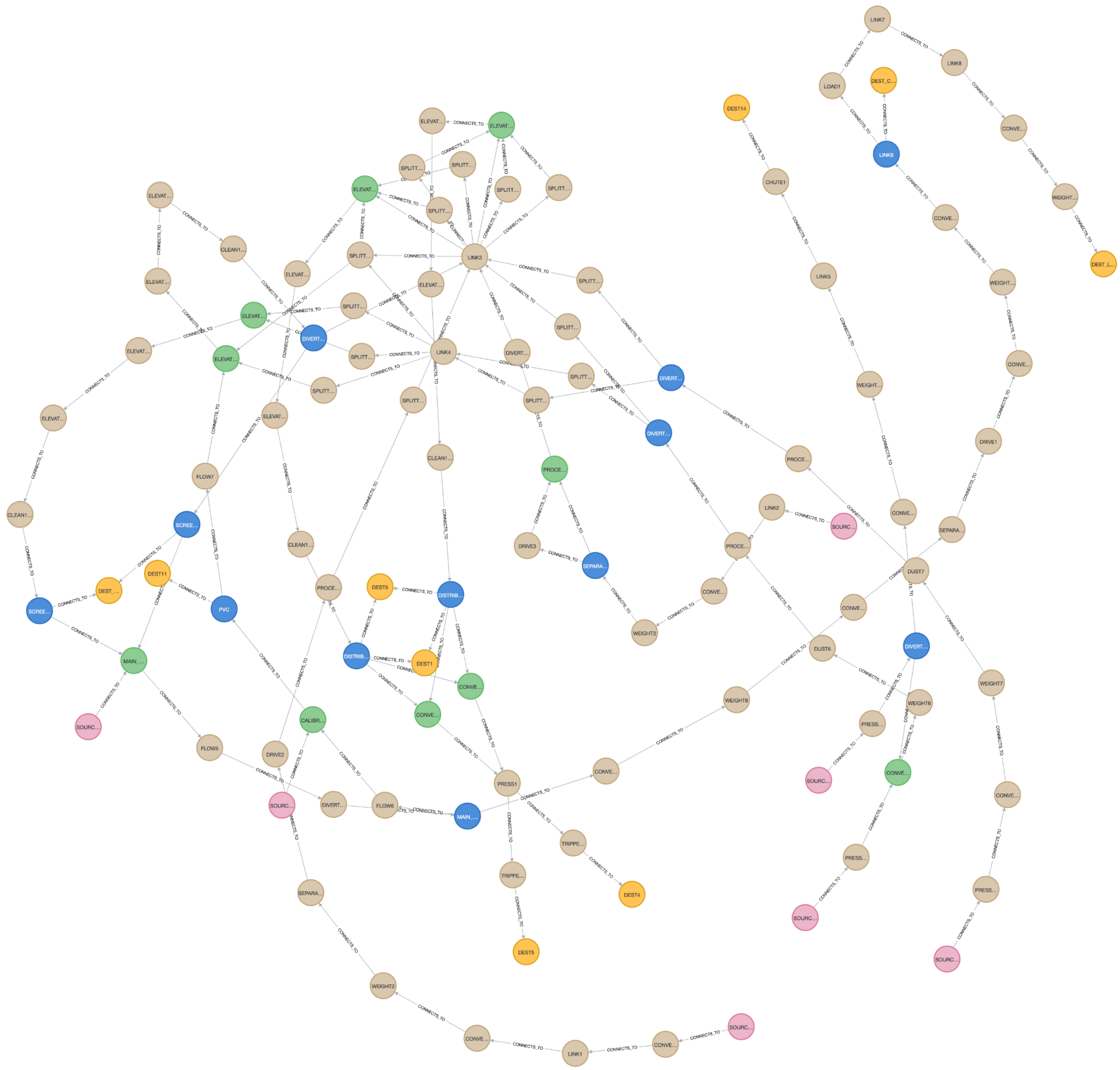
We involve the system administrator to grant them the capability of making modifications, within the system and updating data information. The administrator possesses privileges compared to users, which impacts both the security aspect and the implementation aspect of the coding process.

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