

Final Report

Title: Fianl Report of Group <Path5>

Project: ATSYS_Shortest Path Algorithm for Material Transportation

Mambers:

Yuchen Peng_a1824982

Project Vision

In summary, our initial project vision aimed to use an SQL-based algorithm to identify and calculate the top five shortest paths between given starting and ending points on a graph in order to make material transports in factories more efficient. While the fundamental objective of finding the shortest paths remains the same, as the project evolved, there has been some significant changes. The project shifted away from solely relying on SQL commands, instead, for easier graph generation and visualization we introduced Neo4j Graph Database and to make the interaction with Neo4j smoother, we employed a python script to send our graph query commands and parse received results into structured format. Beyond that, in response to new requirements, the final product also focuses on user experience by introducing a user friendly interface, structured output format, and reasonable execution time and these three factors were not considered in the initial project vision.

Customer Q&A

Question1: ‘Where do we give users the option to exclude devices? Before calculating or implementing a separate feature to update devices status?’

Answer: A separate feature that allows users to update devices status.

Question2: ‘When we add a device to the graph, do we just add it by itself or connect it to the graph with an edge?’

Answer: You can add the device by itself, but an adding edge feature is required, so users can not only connect the device to the graph but also create new edges.

Question3: ‘Are there any limitations in tools and algorithms used?’

Answer: Nah

Reflection:

The focus of the first question is to clarify how users want to interact with our program and the workflow of managing devices. By understanding when and where users want to have the option of excluding devices, we can develop a

program with better user experiences. Also, it helps to ensure that our software aligns with user needs and expectations.

Move on to the second question, the focus for this question is on the data flow in the software. Particularly, we want to understand what is the correct way to add a device to the graph, since this operation will change the database, it is necessary for us to know in order to calculate desired shortest paths and ensure the accuracy of the results.

In a similar way, both questions aim on the software's functionality and how it interacts with users. This information is essential for us to make correct decisions on how we build up our technology solution correctly to best satisfy users' needs.

Moving forward, the focus of our third question is to figure out the constraints or limitations in the software development process. We want to make sure that we do not violate user's restrictions. It is after finding out that there are no limitations and we are free to use Python, Neo4j. Therefore, we can choose the most suitable methods to design the project and the most appropriate technology for our team to implement, allowing us to efficiently develop a better program. Overall, these scenarios demonstrate that communications with the clients is a fundamental component for a successful software development. It ensures the software we developed not only meets user expectations but also in an effective manner.

Users and User Stories

1. Device and Path Manager

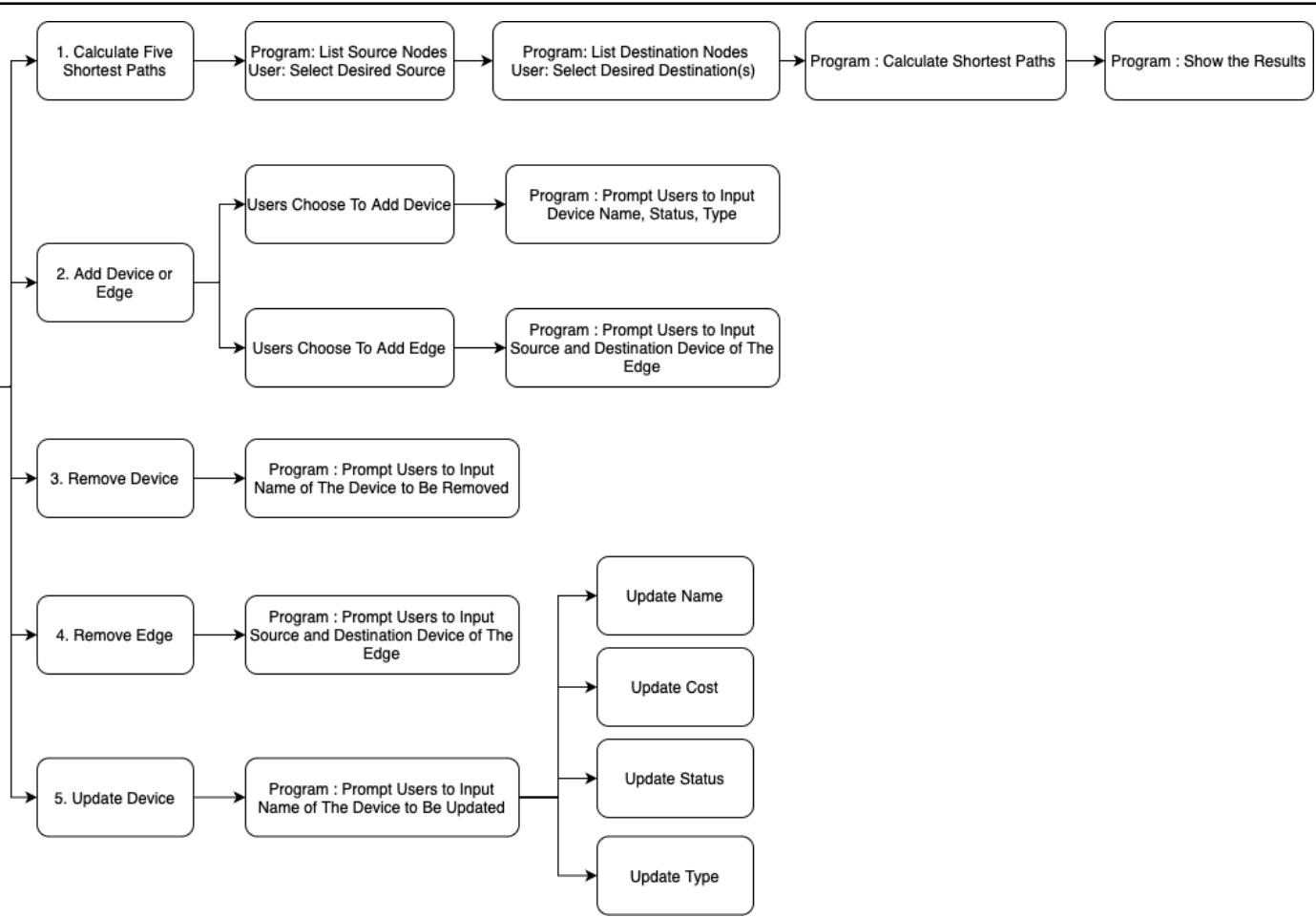
- Responsible for managing all the devices and edges in the database.
- Actions including:
 - Adding/Deleting a device to the database
 - Adding/Deleting an edge to the database
 - Updating device's name/cost/status/type
- Related user stories
 - User story1: Users should be able to *manage devices* in the system, including adding, removing, and updating them.
 - User story3: Users should be able to *mark devices as excluded* to ensure that the shortest paths are calculated without including them.

2. Material Manager

- End users, whose job is to use our key feature: Calculating Top5 Shortest Paths
- Actions including:
 - Choosing a source device and a destination device(s)
 - Viewing our well structured results calculated by our program
- Related user stories
 - User story2: Users should be able to *find the shortest path between two devices.*
 - User story4: Users want to receive five efficient shortest paths *from a single source to multiple destinations, with a well-structured output format.*
 - User story5: Users expect the *execution time* for each operation is less than 4 seconds.

Benefit from the thorough coverage of the first three user stories, users identified in the initial report are already quite comprehensive. The only thing that has changed is we combined the roles of the Device Manager and Path Manager into a single role. It is because we consider these splitting roles to be redundant and unrealistic for a factory hiring two individuals for this position which goes against their goal of reducing operational costs and enhancing overall productivity. Thus, we have replaced them with the role of Device and Path Manager.

Software Architecture:



Justification:

The architecture we designed effectively addresses the user's requirements. In the provided architecture diagram, the first feature addresses the fundamental need of calculating shortest paths. The other functionalities focus on scalability. It is crucial because in a continuously operating factory, we can anticipate that there will be a growing demand for adding new devices and edges, as well as updating or deleting existing ones. During the software architecture design, we took these considerations into account and designed corresponding features to meet these needs. Ultimately, our software can perform a wide range of operations to meet the factory's evolving requirements, demonstrating the adaptability of our architecture to future changes.

Tech Stack and Standards

Tech Stack		
	Initial	Fianl
Back-end	SQL	Python Neo4j Cypher Query
Database	SQL	Neo4j Graph Database
Tools		
Communication	Slack Wechat	Slack Wechat Zoom
Development	Visual Studio Code	Visual Studio Code
Coding Standard		
<i>Formatting</i> - Each line is limited to 120 characters for better readability. Especially long SQL syntax is much harder to read.		
<i>Naming</i> - If required camelCase is applied to variable and function names & PascalCase is applied to class names. All the database reserved words need to be in uppercase.		
<i>Comment</i> - Clear comments required to explain algorithm and logic.		
<i>Code Duplication</i> - Code duplication should be minimized by using functions.		
<i>Testing</i> - Unit tests & Integration test for important features.		
<i>Version Control</i> - Use Git for version control with meaningful messages for each commit.		
<i>Code Reviews</i> - Codes need to pass peer review.		

In the backend aspect, we transitioned away from using SQL databases to Neo4j graph databases. It is because compared with SQL database extension in VS Code, Neo4j allows for easier reading of csv files and can generate corresponding graphs for visualization. Once we can visualize the graphs, it

becomes much easier to do testing because we can see whether the results of our operations are correct or not. Additionally, incorporating Neo4j with python scripts allows us to provide users a user-friendly terminal interface. We encapsulated complex Neo4j queries into Python functions, simplifying the user experience with our software. Instead of having to input lengthy commands in Neo4j, users can now enter simple numbers or keywords in our provided interface to perform their desired operations. Furthermore, using Python allows us to structure the format of our outputs, ensuring a more organized and user-friendly demonstration of the results. So, our initial idea of relying solely on SQL is impractical and could potentially hinder our ability to develop a better software solution.

Everything went well in terms of communication tools. Since WeChat is our most frequently used communication tool, we primarily used WeChat for communication. Additionally, we utilized Slack for communication with the tutor (PO) and conducted client meetings via Zoom.

Group Meetings and Team Member Roles:

1. How frequently and for how long did you meet? Did you manage to timebox the meetings?

Our group met twice a week at Hub Central. While we didn't strictly timebox the meetings, we maintained a meeting agenda that outlined all the issues to be discussed during each meeting and documented the group's decisions.

2. When did you schedule the sprint retrospective meetings?

Twice a week every Tuesday.

3. Did you arrange for additional feedback channels with the customer?

We mainly use Slack to communicate with PO.

4. Please name the Scrum Masters for each sprint (each team member can only be the Scrum Master for up to one sprint)

Sprint1: Yuejun Sprint2: Yuchen Sprint3: Shize

Sprint4: Yuze Sprint5: Shijie

5. Personal reflection on group meetings and my own role.

When it comes to group meetings, I consistently took on the role of leading the discussions and taking meeting notes. In retrospect, I think that I could have done better in time management. Sometimes, discussions deviated from the topic or took too long, and I believe these situations could have been better controlled if timeboxed meetings. So, in the future, I will pay more attention to time management during meetings.

During the development process, I focused on doing documentation and team management. Involving tasks such as writing all the snapshots, managing the Github Task Boards, maintaining the meeting agenda. Additionally, throughout the entire developing process, I kept track of any project progress and maintained a clear understanding of the project's future steps. It allows me to divide tasks appropriately based on the task complexity and assign them to the most suitable group members based on their capabilities, ensuring the team operates efficiently and smoothly. Moreover, I also engaged in coding tasks,

particularly testing. Looking back, during this development process, I practiced my skills in documentation and team management. I think I could have done better by involving myself more in coding tasks. This could expand my technical knowledge and skills and prepare me better for future projects.

Snapshot:

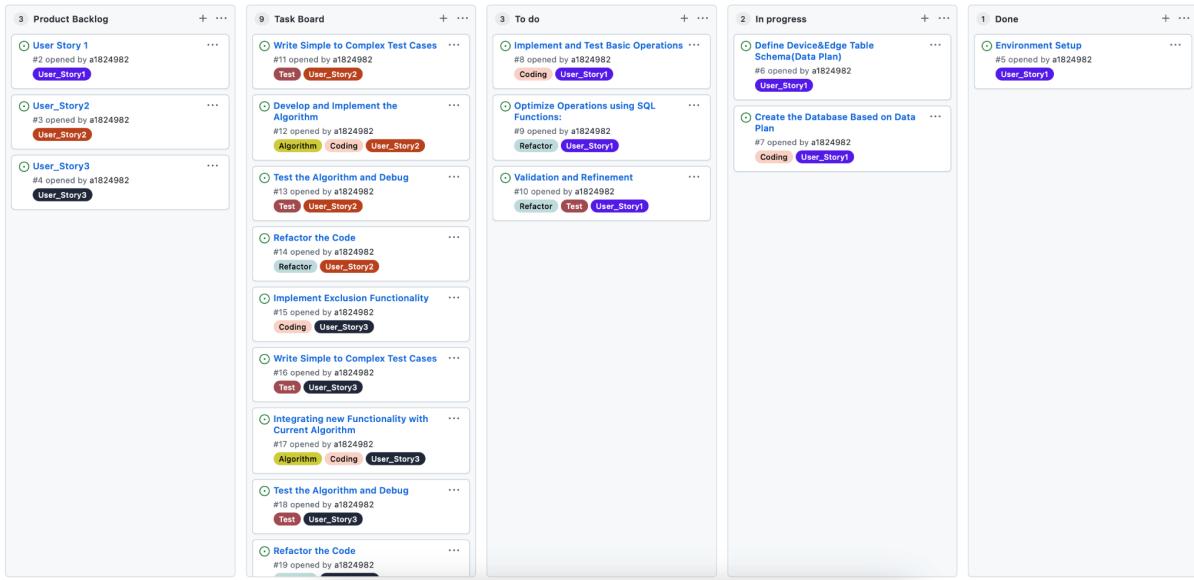
Snapshot Week <4> of Group <Path5>

Project: ATSYS_Shortest Path Algorithm for Material Transportation

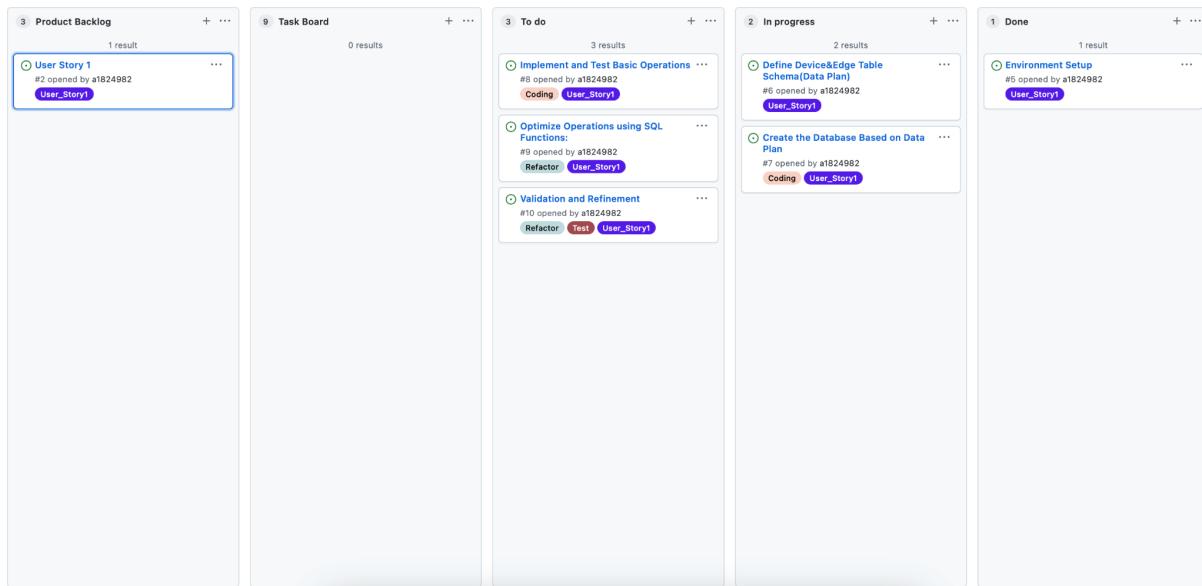
Members:

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

Product Backlog and Task Board



Sprint Backlog and User Stories



The initial user story we've selected to focus on is "***as a user, I want to store devices in the system so that I can add/remove/update them.***". This serves as the foundation of the project, the other user stories will need to be built on the structure and SQL script developed in the first user story.

For the development team, correctly storing device information and the graph structure is the fundamental requirement of implementing the other user stories (e.g. table manipulation, find the shortest path, etc...).

Definition of Done

- A coding task is done when the code is written aligned with coding standards mentioned in the report, reviewed, tested (unit and integration), refactored, successfully passes peer review, and gains unanimous approval from all team members.
- A non-coding task is done when it has been collectively considered, discussed, documented, and decided upon during a meeting, ensuring alignment among team members.

Summary of Changes

This is the first snapshot, no changes so far.

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	2
Sprint Backlog and User Stories	3
Definition of Done	4
Summary of Changes	4

Product Backlog and Task Board

Product Backlog

The diagram shows a 'Product Backlog' interface with three items:

- User Story 1:** As a user, I want to store devices in the system so that I can add/remove/update them.
- User Story 2:** As a user, I want to get the shortest path between 2 given devices so that material transportation will be efficient.
- User Story 3:** As a user I want to mark devices to exclude, so that shortest paths can be identified avoiding them.

Task Board for Sprint 2 Snapshot 2.1 (User Story2)

To Do	In Progress	Done
<p>5 To do + ...</p> <p>(2.6) Set up Cloud Neo4j Database and Load Files into it #53 opened by a1848890 DataBase User_Story2</p> <p>(2.7) Implement Functionality to Draw Graphs in Neo4j Based on Input Files #61 opened by a1824982 DataBase User_Story2</p> <p>(2.8) Update the Shortest Path Algorithm in Neo4j Based on Requirement Change #50 opened by a1848890 Algorithm Coding DataBase User_Story2</p> <p>(2.9) Test/Debug the Algorithm/Current Script #48 opened by a1848890 Algorithm DataBase User_Story2</p> <p>(2.10) Snapshot 2.1 and 2.2 #62 opened by a1824982 Document User_Story2</p>	<p>2 In progress + ...</p> <p>(2.4) Update Data Type in the Database Schema Based on Requirement Change #60 opened by a1824982 DataBase User_Story2</p> <p>(2.5) Research on Visualising the Graph with Large Inputs #52 opened by a1848890 DataBase User_Story2</p>	<p>14 Done + ...</p> <p>(2.3) Refactor the Code #14 opened by a1824982 Refactor User_Story2</p> <p>(2.2) Test/Debug the Algorithm #13 opened by a1824982 Test User_Story2</p> <p>(2.1) Implement Shortest Path Algorithm #12 opened by a1824982 Algorithm User_Story2</p>

Sprint Backlog and User Stories

Sprint Backlog

Product Backlog	To do	In progress	Done
3 results	5 results	2 results	3 results
User_Story2 #3 opened by a1824982 User_Story2	2.6) Set up Cloud Neo4j Database and Load Files into it #53 opened by a1848890 DataBase User_Story2 2.7) Implement Functionality to Draw Graphs in Neo4j Based on Input Files #61 opened by a1824982 DataBase User_Story2 2.8) Update the Shortest Path Algorithm in Neo4j Based on Requirement Change #50 opened by a1848890 Algorithm Coding DataBase User_Story2 2.9) Test/Debug the Algorithm/Current Script #48 opened by a1848890 Algorithm DataBase User_Story2 2.10) Snapshot 2.1 and 2.2 #62 opened by a1824982 Document User_Story2	2.4) Update Data Type in the Database Schema Based on Requirement Change #60 opened by a1824982 DataBase User_Story2 2.5) Research on Visualising the Graph with Large Inputs #52 opened by a1848890 DataBase User_Story2	2.3) Refactor the Code #14 opened by a1824982 Refactor User_Story2 2.2) Test/Debug the Algorithm #13 opened by a1824982 Test User_Story2 2.1) Implement Shortest Path Algorithm #12 opened by a1824982 Algorithm User_Story2

In the second sprint, the user story our group is working on is: '***As a user, I want to get the shortest path between 2 given devices so that material transportation will be efficient.***'

In this user story, users are requesting a method to determine the top 5 efficient routes between two given devices. This functionality will enable them to effectively and promptly manage the transportation between their plants. The purpose is to increase the overall productivity and cost efficiency.

An algorithm needs to be designed to accomplish this requirement. This algorithm should calculate the cost of paths between the selected devices. It will make a recursion approach to ensure that every device is visited, all routes are explored before calculating the cost and sorting them in order.

Definition of Done

- A coding task is considered to be completed when the code has been written in accordance with the coding standards outlined in the initial report, 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 is considered to be completed when it has been brainstormed, discussed, documented, reviewed and agreed upon by the team in a meeting to ensure everyone is aligned and informed about the task. Additionally, any specific problems that arose during the Sprint should be reported to the team in detail and converted to an issue on the GitHub task board.

Summary of Changes

During this week's snapshot, we have implemented several changes and updates to the project. These changes mainly focus on moving from user story 1 to user story 2, implementing the functionality of finding the shortest path as required in user story 2. Highlights include:

1. Finished the testing for codes/scripts written in user story1.
2. Implemented the algorithm to find the shortest path by using SQL commands.
3. Tested the scripts in the scenario stimulated in user story1.
4. Refactored the commands.
5. Updated everything on Github Taskboard.

Overall, these are the main changes our group made between the previous snapshot.

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	2, 3
Sprint Backlog and User Stories	4
Definition of Done	5
Summary of Changes	5

Product Backlog and Task Board

Product Backlog		
User Story 1	User Story 2	User Story 3
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.

Task Board for Sprint 2 Snapshot 2.2 (User Story2)	
To Do	In Progress
<p>1 To do + ...</p> <p>1 result</p> <p>2.10) Snapshot 2.1 and 2.2 #62 opened by a1824982</p> <p>Document User_Story2</p> 	<p>1 In progress + ...</p> <p>1 result</p> <p>2.9) Test/Debug the Algorithm/Current Script #48 opened by a1848890</p> <p>Algorithm DataBase User_Story2</p> 

Task Board for Sprint 2 Snapshot 2.2 (User Story2)

Done

<p>④ 2.4) Update Data Type in the Database Schema Based on Requirement Change #60 opened by a1824982 DataBase User_Story2</p> <p>④ 2.3) Refactor the Code #14 opened by a1824982 Refactor User_Story2</p> <p>④ 2.2) Test/Debug the Algorithm #13 opened by a1824982 Test User_Story2</p> <p>④ 2.1) Implement Shortest Path Algorithm #12 opened by a1824982 Algorithm User_Story2</p>	<p>④ 2.8) Update the Shortest Path Algorithm in Neo4j Based on Requirement Change #50 opened by a1848890 Algorithm Coding DataBase User_Story2</p> <p>④ 2.7) Implement Functionality to Draw Graphs in Neo4j Based on Input Files #61 opened by a1824982 DataBase User_Story2</p> <p>④ 2.6) Set up Cloud Neo4j Database and Load Files into it #53 opened by a1848890 DataBase User_Story2</p> <p>④ 2.5) Research on Visualising the Graph with Large Inputs #52 opened by a1848890 DataBase User_Story2</p>
---	--

Sprint Backlog and User Stories

Sprint Backlog

The image shows a digital sprint backlog board with the following structure:

- Product Backlog:** Contains 1 result, one item: **User_Story2** (#3 opened by a1824982), with tags **User_Story2**.
- To do:** Contains 1 result, one item: **2.10) Snapshot 2.1 and 2.2** (#62 opened by a1824982), with tags **Document** and **User_Story2**.
- In progress:** Contains 1 result, one item: **2.9) Test/Debug the Algorithm/Current Script** (#48 opened by a1848890), with tags **Algorithm**, **DataBase**, and **User_Story2**.
- Done:** Contains 8 results, including:
 - 2.8) Update the Shortest Path Algorithm in Neo4j Based on Requirement Change** (#50 opened by a1848890), with tags **Algorithm**, **Coding**, **DataBase**, **User_Story2**.
 - 2.7) Implement Functionality to Draw Graphs in Neo4j Based on Input Files** (#61 opened by a1824982), with tags **DataBase** and **User_Story2**.
 - 2.6) Set up Cloud Neo4j Database and Load Files into it** (#53 opened by a1848890), with tags **DataBase** and **User_Story2**.
 - 2.5) Research on Visualising the Graph with Large Inputs** (#52 opened by a1848890), with tags **DataBase** and **User_Story2**.
 - 2.4) Update Data Type in the Database Schema Based on Requirement Change** (#60 opened by a1824982), with tags **DataBase** and **User_Story2**.
 - 2.3) Refactor the Code** (#14 opened by a1824982), with tags **Refactor** and **User_Story2**.
 - 2.2) Test/Debug the Algorithm** (#13 opened by a1824982), with tags **Test** and **User_Story2**.
 - 2.1) Implement Shortest Path Algorithm** (#12 opened by a1824982), with tags **Algorithm** and **User_Story2**.

In the second sprint, the user story our group is working on is: '***As a user, I want to get the shortest path between 2 given devices so that material transportation will be efficient.***'

In this user story, users are requesting a method to determine the top 5 efficient routes between two given devices. This functionality will enable them to effectively and promptly manage the transportation between their plants. The purpose is to increase the overall productivity and cost efficiency.

An algorithm needs to be designed to accomplish this requirement. This algorithm should calculate the cost of paths between the selected devices. It will make a

recursion approach to ensure that every device is visited, all routes are explored before calculating the cost and sorting them in order.

Definition of Done

- A coding task is considered to be completed when the code has been written in accordance with the coding standards outlined in the initial report, 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 is considered to be completed when it has been brainstormed, discussed, documented, reviewed and agreed upon by the team in a meeting to ensure everyone is aligned and informed about the task. Additionally, any specific problems that arose during the Sprint should be reported to the team in detail and converted to an issue on the GitHub task board.

Summary of Changes

During this week's snapshot, we have implemented many changes and updates to the project. Our primary focus was addressing the requirements change from the product owner. Highlights include:

1. Updated screen shots in Product Backlog and Sprint Backlog to make it more clearer to see and easier to understand.
2. Updated everything on Github Taskboard.
3. Moved from Mysql Database to Neo4j Cloud Database.
 - a. Implemented commands to read in csv files.
 - b. Able to generate path diagrams based on inputs provided by PO(tutor).
 - c. More convenient when inputs are large.
4. Updated the Mixed attribute in our device table.
5. Updated the Cost attribute in our edge table.

Overall, these are the main changes our group made between the previous snapshot.

Snapshot Week <7> of Group <Path5>

Snapshot 3.1

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	2
Sprint Backlog and User Stories	3,4
Definition of Done	4
Summary of Changes	4, 5

Product Backlog and Task Board

Product Backlog		
User Story 1	User Story 2	User Story 3
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.

Task Board for Sprint 3 Snapshot 3.1 (User Story3)	
To Do	In Progress
<p>2 To do + ...</p> <p>2 results</p> <ul style="list-style-type: none"> 3.4) Integrate Exclusion Functionality ... #15 opened by a1824982 Coding User_Story3 3.5) Test/Debug the Algorithm ... #18 opened by a1824982 Test User_Story3 	<p>2 In progress + ...</p> <p>2 results</p> <ul style="list-style-type: none"> 3.2) Implement the Functionality to Mark Devices as Excluded #63 opened by a1824982 Coding User_Story3 3.3) Update the Algorithm/Script in Neo4j to Take 'Status' into Consideration #31 opened by a1848890 Algorithm DataBase User_Story3

Done
<p>22 Done + ...</p> <p>1 result</p> <ul style="list-style-type: none"> 3.1) Update 'Status' Attribute of Nodes in Device Table ... #30 opened by a1848890 DataBase User_Story3

Sprint Backlog and User Stories

Sprint Backlog

The Sprint Backlog board displays the following columns:

- Product Backlog:** 3 items, 1 result. One item is "User_Story3" with details: #4 opened by a1824982, User_Story3.
- To do:** 2 results. Two items are listed: "3.4) Integrate Exclusion Functionality" and "3.5) Test/Debug the Algorithm". Both have details: #15 and #18 opened by a1824982, and tags Coding, User_Story3.

The Sprint Backlog board displays the following columns:

- In progress:** 2 results. Two items are listed: "3.2) Implement the Functionality to Mark Devices as Excluded" and "3.3) Update the Algorithm/Script in Neo4j to Take 'Status' into Consideration". Both have details: #63 and #31 opened by a1824982, and tags Coding, User_Story3.
- Done:** 22 items, 1 result. One item is "3.1) Update 'Status' Attribute of Nodes in Device Table" with details: #30 opened by a1848890, and tags DataBase, User_Story3.

In the third sprint, we have moved on to the third user story: ***“As a user I want to mark devices to exclude, so that shortest paths can be identified avoiding them.”***

In this user story, users want to be able to exclude specific devices when calculating the shortest path between two given devices. Realistically, this functionality is important to implement because devices in the factory can be broken and including them in the calculation is not practical.

In order to achieve this, the database schema needs to be updated. To be specific, an additional attribute ‘Status’ will be added to our Device table to keep track of the condition of each device/node. Next, a logic condition will be added to our algorithm to avoid including unwanted devices into calculation.

Definition of Done

- A coding task is considered to be completed when the code has been written in accordance with the coding standards outlined in the initial report, 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 task is considered to be completed when it has been brainstormed, discussed, documented, reviewed and agreed upon by the team in a meeting to ensure everyone is aligned and informed about the task. Additionally, any specific problems that arose during the Sprint should be reported to the team in detail and converted to an issue on the GitHub task board.

Summary of Changes

During this week’s snapshot, several important changes and updates have been made. These changes mainly focus on adding a new functionality of marking devices as excluded when executing the shortest path algorithm described in user story 3. Highlights include:

1. **New ‘Status’ Attribute for Nodes:** We introduced a new attribute called “status” for nodes, which has three values: ‘Active’, ‘In Use’ and ‘Fault’.
2. **New Logic Condition In Shortest Path Algorithm:** Now, when calculating for the shortest paths between two given devices, if the path contains ‘Fault’

or ‘In Use’ devices, it will be considered as invalid and will not be shown as a result candidate.

3. **New Devices Exclusion Functionality:** We have developed a script containing commands that can enable users to mark specific devices as excluded when calculating for the shortest path.
4. **Updated** everything on Github Taskboard.
5. **Improved** Visibility on Screenshots taken.

Overall, these are the main changes our group made between the previous snapshot.

Snapshot Week <8> of Group <Path5>

Snapshot 3.2

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	2, 3
Sprint Backlog and User Stories	3, 4
Definition of Done	5
Summary of Changes	5

Product Backlog and Task Board

Product Backlog		
User Story 1	User Story 2	User Story 3
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.

Task Board for Sprint 3 Snapshot 3.2 (User Story3)	
To Do	In Progress
<p>1 Task Board + ...</p> <p>0 results</p>	<p>2 In progress + ...</p> <p>2 results</p> <ul style="list-style-type: none"> ● 3.6) Investigate on Optimising Operations for Users ... <p>#65 opened by a1824982</p> <p>Refactor User_Story3</p> <ul style="list-style-type: none"> ● 3.5) Test/Debug the Algorithm ... <p>#18 opened by a1824982</p> <p>Test User_Story3</p>

Done

25 Done + ...

4 results

- 3.4) Integrate Exclusion Functionality** ...

#15 opened by a1824982

Coding User_Story3
- 3.3) Update the Algorithm/Script in Neo4j to Take 'Status' into Consideration** ...

#31 opened by a1848890

Algorithm DataBase User_Story3
- 3.2) Implement the Functionality to Mark Devices as Excluded** ...

#63 opened by a1824982

Coding User_Story3
- 3.1) Update 'Status' Attribute of Nodes in Device Table** ...

#30 opened by a1848890

DataBase User_Story3

Sprint Backlog and User Stories

Sprint Backlog

Product Backlog	To do	In progress
3 Product Backlog + ... 1 result User_Story3 ... <p>#4 opened by a1824982</p> <p>User_Story3</p>	0 To do + ... 0 results	2 In progress + ... 2 results 3.6) Investigate on Optimising Operations for Users ... <p>#65 opened by a1824982</p> <p>Refactor User_Story3</p> 3.5) Test/Debug the Algorithm ... <p>#18 opened by a1824982</p> <p>Test User_Story3</p>

25 Done + ...

4 results

- 3.4) Integrate Exclusion Functionality** ...

#15 opened by a1824982

Coding User_Story3
- 3.3) Update the Algorithm/Script in Neo4j to Take 'Status' into Consideration** ...

#31 opened by a1848890

Algorithm DataBase User_Story3
- 3.2) Implement the Functionality to Mark Devices as Excluded** ...

#63 opened by a1824982

Coding User_Story3
- 3.1) Update 'Status' Attribute of Nodes in Device Table** ...

#30 opened by a1848890

DataBase User_Story3

In the third sprint, we have moved on to the third user story: ***“As a user I want to mark devices to exclude, so that shortest paths can be identified avoiding them.”***

In this user story, users want to be able to exclude specific devices when calculating the shortest path between two given devices. Realistically, this functionality is important to implement because devices in the factory can be broken and including them in the calculation is not practical.

In order to achieve this, the database schema needs to be updated. To be specific, an additional attribute ‘Status’ will be added to our Device table to keep track of the condition of each device/node. Next, a logic condition will be added to our algorithm to avoid including unwanted devices into calculation.

Definition of Done

- A coding task is considered to be completed when the code has been written in accordance with the coding standards outlined in the initial report, 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 is considered to be completed when it has been brainstormed, discussed, documented, reviewed and agreed upon by the team in a meeting to ensure everyone is aligned and informed about the task. Additionally, any specific problems that arose during the Sprint should be reported to the team in detail and converted to an issue on the GitHub task board.

Summary of Changes

During this week's snapshot, few changes and updates have been made. These changes mainly focus on integrating all components from 3 user stories, testing the functionality and investigating on optimizing the process of how users interact with our software. Highlights include:

1. **New Functionality ‘Exclusion’:** The Exclusion functionality is now integrated to our current algorithm.
2. **Updated Scripts:** Now, SQL commands/queries will be generated from a python script.
 - a. This can avoid complex and redundant inputs when users want to perform an operation
3. **Updated** everything on Github Taskboard.
4. **Improved** Visibility on Screenshots taken.

Overall, these are the main changes our group made between the previous snapshot.

Snapshot Week <9> of Group <Path5>

Snapshot 4.1

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	2, 3
Sprint Backlog and User Stories	4
Definition of Done	5
Summary of Changes	5

Product Backlog and Task Board

Product Backlog

User Story 1	User Story 2	User Story 3
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.
User Story 4	User Story 5	
As a user, I want to get 5 shortest paths given a single source and multiple destinations, so that material distribution will be efficient.	As a user, I want the execution time of each operation to be optimised as possible and visualise the output (the 5 shortest paths) as a table or as a console output ordered by the path cost, so that user experience aspect will be improved.	

Task Board	
Tasks For User Story 4	Tasks For User Story 5
<p>4.1) Implement Add Device/Edge Function and Test it ...</p> <p>#73 opened by a1824982</p> <p>Algorithm Test User_Story4</p>	<p>5.1) Add the 'Timer Functionality' ...</p> <p>#79 opened by a1824982</p> <p>Coding User_Story5</p>
<p>4.2) Design User Interface / Interaction Process ...</p> <p>#74 opened by a1824982</p> <p>Document User_Story4</p>	<p>5.2) Test if '1-to-1 scenarios' Reach Acceptance Criteria ...</p> <p>#80 opened by a1824982</p> <p>Test User_Story5</p>
<p>4.3) Implement the User Interface ...</p> <p>#75 opened by a1824982</p> <p>Coding User_Story4</p>	<p>5.3) Test if '1-to-N scenarios' Reach Acceptance Criteria ...</p> <p>#81 opened by a1824982</p> <p>Test User_Story5</p>
<p>4.4) Modify the Calculate Cost Algorithm ...</p> <p>#77 opened by a1824982</p> <p>Algorithm User_Story4</p>	<p>5.4) Optimise the Program if Needed ...</p> <p>#82 opened by a1824982</p> <p>Refactor User_Story5</p>
<p>4.5) Integrate the Interface and Functionality ...</p> <p>#76 opened by a1824982</p> <p>Coding User_Story4</p>	
<p>4.5) Testing the Entire Program ...</p> <p>#78 opened by a1824982</p> <p>Test User_Story4</p>	

Sprint Backlog and User Stories

Sprint Backlog		
ToDo	In Progress	Done
<p>2 To do + ...</p> <p>2 results</p> <ul style="list-style-type: none"> 4.5) Integrate the Interface and Functionality #76 opened by a1824982 Coding User_Story4 4.5) Testing the Entire Program #78 opened by a1824982 Test User_Story4 	<p>3 In progress + ...</p> <p>3 results</p> <ul style="list-style-type: none"> 4.1) Implement Add Device/Edge Function and Test it #73 opened by a1824982 Algorithm Test User_Story4 4.3) Implement the User Interface #75 opened by a1824982 Coding User_Story4 4.4) Modify the Calculate Cost Algorithm #77 opened by a1824982 Algorithm User_Story4 	<p>28 Done + ...</p> <p>1 result</p> <ul style="list-style-type: none"> 4.2) Design User Interface / Interaction Process #74 opened by a1824982 Document User_Story4

In the fourth sprint, we started working on a new user story: “**As a user, I want to get 5 shortest paths given a single source and multiple destinations, so that material distribution will be efficient**”. In this user story, users not only want to find out the distance from one starting point to a single destination but also the cost from one starting point to multiple destinations. This will increase the efficiency of material transportation. Additionally, users have provided the expected output. We need to modify our current algorithm to calculate any overlap in the paths in ‘1 to N’ scenarios and add a user-friendly interface for users to perform desired operations and display the requested results.

Definition of Done

- A coding task is considered to be completed when the code has been written in accordance with the coding standards outlined in the initial report, 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 is considered to be completed when it has been brainstormed, discussed, documented, reviewed and agreed upon by the team in a meeting to ensure everyone is aligned and informed about the task. Additionally, any specific problems that arose during the Sprint should be reported to the team in detail and converted to an issue on the GitHub task board.

Summary of Changes

During this week's snapshot, several changes, updates and decisions have been made. These changes mainly focus on: 1) realizing the functionality of computing the path from one source to multiple destinations with the lowest cost 2) designing and implementing a user-friendly interface for users to perform required operation easily. Highlights include:

1. **Broke** two new user stories down into several small tasks.
2. **Carefully designed** a new user interface to streamline all the operations.
3. **Successfully connected** python scripts with Neo4j database.
4. **Updated** the algorithm to calculate the shortest path, testing will be done next week.
5. **Updated** everything on Github Taskboard.

Overall, these are the main updates made between the previous snapshot.

Snapshot Week <10> of Group <Path5>

Snapshot 4.2

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	2, 3
Sprint Backlog and User Stories	4
Definition of Done	5
Summary of Changes	5

Product Backlog and Task Board

Product Backlog

User Story 1	User Story 2	User Story 3
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.
User Story 4	User Story 5	
As a user, I want to get 5 shortest paths given a single source and multiple destinations, so that material distribution will be efficient.	As a user, I want the execution time of each operation to be optimised as possible and visualise the output (the 5 shortest paths) as a table or as a console output ordered by the path cost, so that user experience aspect will be improved.	

Task Board	
Tasks For User Story 4	Tasks For User Story 5
<p>4.1) Implement Add Device/Edge Function and Test it ...</p> <p>#73 opened by a1824982</p> <p>Algorithm Test User_Story4</p>	<p>5.1) Add the 'Timer Functionality' ...</p> <p>#79 opened by a1824982</p> <p>Coding User_Story5</p>
<p>4.2) Design User Interface / Interaction Process ...</p> <p>#74 opened by a1824982</p> <p>Document User_Story4</p>	<p>5.2) Test if '1-to-1 scenarios' Reach Acceptance Criteria ...</p> <p>#80 opened by a1824982</p> <p>Test User_Story5</p>
<p>4.3) Implement the User Interface ...</p> <p>#75 opened by a1824982</p> <p>Coding User_Story4</p>	<p>5.3) Test if '1-to-N scenarios' Reach Acceptance Criteria ...</p> <p>#81 opened by a1824982</p> <p>Test User_Story5</p>
<p>4.4) Modify the Calculate Cost Algorithm ...</p> <p>#77 opened by a1824982</p> <p>Algorithm User_Story4</p>	<p>5.4) Optimise the Program if Needed ...</p> <p>#82 opened by a1824982</p> <p>Refactor User_Story5</p>
<p>4.5) Integrate the Interface and Functionality ...</p> <p>#76 opened by a1824982</p> <p>Coding User_Story4</p>	
<p>4.5) Testing the Entire Program ...</p> <p>#78 opened by a1824982</p> <p>Test User_Story4</p>	

Sprint Backlog and User Stories

Sprint Backlog		
ToDo	In Progress	Done
<p>0 To do</p> <p>0 results</p>	<p>1 In progress</p> <p>1 result</p> <p>4.5) Testing the Entire Program #78 opened by a1824982</p> <p>Test User_Story4</p>	<p>32 Done</p> <p>5 results</p> <p>4.1) Implement Add Device/Edge Function and Test it #73 opened by a1824982</p> <p>Algorithm Test User_Story4</p> <p>4.2) Design User Interface / Interaction Process #74 opened by a1824982</p> <p>Document User_Story4</p> <p>4.3) Implement the User Interface #75 opened by a1824982</p> <p>Coding User_Story4</p> <p>4.4) Modify the Calculate Cost Algorithm #77 opened by a1824982</p> <p>Algorithm User_Story4</p> <p>4.5) Integrate the Interface and Functionality #76 opened by a1824982</p> <p>Coding User_Story4</p>

In the fourth sprint, we started working on a new user story: **“As a user, I want to get 5 shortest paths given a single source and multiple destinations, so that material distribution will be efficient”**. In this user story, users not only want to find out the distance from one starting point to a single destination but also the cost from one starting point to multiple destinations. This will increase the efficiency of material transportation. Additionally, users have provided the expected output. We need to modify our current algorithm to calculate any overlap in the paths in ‘1 to N’ scenarios and add a user-friendly interface for users to perform desired operations and display the requested results.

Definition of Done

- A coding task is considered to be completed when the code has been written in accordance with the coding standards outlined in the initial report, 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 is considered to be completed when it has been brainstormed, discussed, documented, reviewed and agreed upon by the team in a meeting to ensure everyone is aligned and informed about the task. Additionally, any specific problems that arose during the Sprint should be reported to the team in detail and converted to an issue on the GitHub task board.

Summary of Changes

During this week's snapshot, several changes, updates have been made. These changes mainly focus on: 1) implementing the user interfaces 2) integrating the interface with implemented functionality. Highlights include:

1. **Implemented** several versions of the user interface.
2. **Updated** the algorithm to implement 1-to-N scenarios.
3. **Successfully integrated** the user interface with implemented functionality.
4. **Updated** everything on Github Taskboard.

Overall, these are the main updates made between the previous snapshot. Next week, we will finish the testing and move to user story 5.

Snapshot Week <11> of Group <Path5>

Snapshot 5.1

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	2, 3
Sprint Backlog and User Stories	4
Definition of Done	5
Summary of Changes	5

Product Backlog and Task Board

Product Backlog

User Story 1	User Story 2	User Story 3
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.
User Story 4	User Story 5	
As a user, I want to get 5 shortest paths given a single source and multiple destinations, so that material distribution will be efficient.	As a user, I want the execution time of each operation to be optimised as possible and visualise the output (the 5 shortest paths) as a table or as a console output ordered by the path cost, so that user experience aspect will be improved.	

Task Board	
Tasks For User Story 4	Tasks For User Story 5
<p>4.1) Implement Add Device/Edge Function and Test it ...</p> <p>#73 opened by a1824982</p> <p>Algorithm Test User_Story4</p>	<p>5.1) Add the 'Timer Functionality' ...</p> <p>#79 opened by a1824982</p> <p>Coding User_Story5</p>
<p>4.2) Design User Interface / Interaction Process ...</p> <p>#74 opened by a1824982</p> <p>Document User_Story4</p>	<p>5.2) Test if '1-to-1 scenarios' Reach Acceptance Criteria ...</p> <p>#80 opened by a1824982</p> <p>Test User_Story5</p>
<p>4.3) Implement the User Interface ...</p> <p>#75 opened by a1824982</p> <p>Coding User_Story4</p>	<p>5.3) Test if '1-to-N scenarios' Reach Acceptance Criteria ...</p> <p>#81 opened by a1824982</p> <p>Test User_Story5</p>
<p>4.4) Modify the Calculate Cost Algorithm ...</p> <p>#77 opened by a1824982</p> <p>Algorithm User_Story4</p>	<p>5.4) Optimise the Program if Needed ...</p> <p>#82 opened by a1824982</p> <p>Refactor User_Story5</p>
<p>4.5) Integrate the Interface and Functionality ...</p> <p>#76 opened by a1824982</p> <p>Coding User_Story4</p>	
<p>4.5) Testing the Entire Program ...</p> <p>#78 opened by a1824982</p> <p>Test User_Story4</p>	

Sprint Backlog and User Stories

Sprint Backlog		
ToDo	In Progress	Done
<p>1 To do + ...</p> <p>1 result</p> <p>5.4) Optimise the Program if Needed ... #82 opened by a1824982 Refactor User_Story5</p>	<p>2 In progress + ...</p> <p>2 results</p> <p>5.2) Test if '1-to-1 scenarios' Reach Acceptance Criteria ... #80 opened by a1824982 Test User_Story5</p> <p>5.3) Test if '1-to-N scenarios' Reach Acceptance Criteria ... #81 opened by a1824982 Test User_Story5</p>	<p>34 Done + ...</p> <p>1 result</p> <p>5.1) Add the 'Timer Functionality' ... #79 opened by a1824982 Coding User_Story5</p>

In the fifth sprint, we started working on a new user story: “**As a user, I want the execution time of each operation to be optimized as possible and visualize the output (the 5 shortest paths) as a table or as a console output ordered by the path cost, so that user experience aspect will be improved**”. In this user story, users want to enhance efficiency by reducing waiting times for calculating the top five shortest paths. This optimisation on the program aims to make the user experience smoother and more responsive. Beyond that, the users ask to visualize the output in a structured format, so they can quickly identify the five shortest paths which improves user experience. We will 1) implement a timer function to keep track of the time taken for each operation 2) optimize our algorithm to meet the criteria and requirements from users 3) structure our outputs.

Definition of Done

- A coding task is considered to be completed when the code has been written in accordance with the coding standards outlined in the initial report, 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 is considered to be completed when it has been brainstormed, discussed, documented, reviewed and agreed upon by the team in a meeting to ensure everyone is aligned and informed about the task. Additionally, any specific problems that arose during the Sprint should be reported to the team in detail and converted to an issue on the GitHub task board.

Summary of Changes

During this week's snapshot, several changes, updates have been made. These changes mainly focus on: 1) implementing the timer functionality 2) testing our software under time condition.

Highlights include:

1. **Implemented** timer functionality.
2. **Integrated** the new feature to our program so now we can see the time for each operation.
3. **Finished** all testing that is left from the previous sprint.
4. **Updated** everything on Github Taskboard.

Overall, these are the main updates made between the previous snapshot.

Snapshot Week <12> of Group <Path5>

Snapshot 5.2

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	2, 3
Sprint Backlog and User Stories	4
Definition of Done	5
Summary of Changes	5

Product Backlog and Task Board

Product Backlog

User Story 1	User Story 2	User Story 3
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.
User Story 4	User Story 5	
As a user, I want to get 5 shortest paths given a single source and multiple destinations, so that material distribution will be efficient.	As a user, I want the execution time of each operation to be optimised as possible and visualise the output (the 5 shortest paths) as a table or as a console output ordered by the path cost, so that user experience aspect will be improved.	

Task Board	
Tasks For User Story 4	Tasks For User Story 5
<p>4.1) Implement Add Device/Edge Function and Test it ...</p> <p>#73 opened by a1824982</p> <p>Algorithm Test User_Story4</p>	<p>5.1) Add the 'Timer Functionality' ...</p> <p>#79 opened by a1824982</p> <p>Coding User_Story5</p>
<p>4.2) Design User Interface / Interaction Process ...</p> <p>#74 opened by a1824982</p> <p>Document User_Story4</p>	<p>5.2) Test if '1-to-1 scenarios' Reach Acceptance Criteria ...</p> <p>#80 opened by a1824982</p> <p>Test User_Story5</p>
<p>4.3) Implement the User Interface ...</p> <p>#75 opened by a1824982</p> <p>Coding User_Story4</p>	<p>5.3) Test if '1-to-N scenarios' Reach Acceptance Criteria ...</p> <p>#81 opened by a1824982</p> <p>Test User_Story5</p>
<p>4.4) Modify the Calculate Cost Algorithm ...</p> <p>#77 opened by a1824982</p> <p>Algorithm User_Story4</p>	<p>5.4) Optimise the Program if Needed ...</p> <p>#82 opened by a1824982</p> <p>Refactor User_Story5</p>
<p>4.5) Integrate the Interface and Functionality ...</p> <p>#76 opened by a1824982</p> <p>Coding User_Story4</p>	
<p>4.5) Testing the Entire Program ...</p> <p>#78 opened by a1824982</p> <p>Test User_Story4</p>	

Sprint Backlog and User Stories

Sprint Backlog		
ToDo	In Progress	Done
<p>0 To do + ... 0 results</p>	<p>0 In progress + ... 0 results</p>	<p>37 Done + ... 4 results</p> <ul style="list-style-type: none"> (●) 5.4) Optimise the Program if Needed ... #82 opened by a1824982 Refactor User_Story5 (●) 5.3) Test if '1-to-N scenarios' Reach Acceptance Criteria ... #81 opened by a1824982 Test User_Story5 (●) 5.2) Test if '1-to-1 scenarios' Reach Acceptance Criteria ... #80 opened by a1824982 Test User_Story5 (●) 5.1) Add the 'Timer Functionality' ... #79 opened by a1824982 Coding User_Story5

In the fifth sprint, we started working on a new user story: “**As a user, I want the execution time of each operation to be optimized as possible and visualize the output (the 5 shortest paths) as a table or as a console output ordered by the path cost, so that user experience aspect will be improved**”. In this user story, users want to enhance efficiency by reducing waiting times for calculating the top five shortest paths. This optimisation on the program aims to make the user experience smoother and more responsive. Beyond that, the users ask to visualize the output in a structured format, so they can quickly identify the five shortest paths which improves user experience. We will 1) implement a timer function to keep track of the time taken for each operation 2) optimize our algorithm to meet the criteria and requirements from users 3) structure our outputs.

Definition of Done

- A coding task is considered to be completed when the code has been written in accordance with the coding standards outlined in the initial report, 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 is considered to be completed when it has been brainstormed, discussed, documented, reviewed and agreed upon by the team in a meeting to ensure everyone is aligned and informed about the task. Additionally, any specific problems that arose during the Sprint should be reported to the team in detail and converted to an issue on the GitHub task board.

Summary of Changes

During this week's snapshot, several changes, updates have been made. These changes mainly focus on: 1) Optimizing the execution time for each operation Highlights include:

1. **Testing** our program to make sure each operation takes reasonable time.
2. **Optimizing** our program's '1-to-N Scenarios'.
3. **Updated** everything on Github Taskboard.

Overall, these are the main updates made between the previous snapshot.