Eng 3006 Software Engineering & Project

Shortest Path Algorithm for Material Transportation

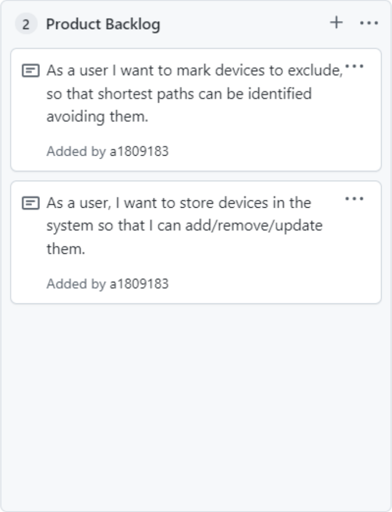
Sprint Retrospective 2 of Group Path 12

SEP Group Path 12 Members: Adam Rebes (a1793912), William Robinson (a1724943), Ayman Shaawi(a1799924), Gurvir Singh (a1796192), Stephanie Turner(a1833535), Nelson Peterson (a1801266), Murray Chahl (a1764549), Rukudzo Ndudzo (a1802422), Addy Dhingra (a1803893)

September 2023

Snapshots

Snapshot 2.1



2 Product Backlog

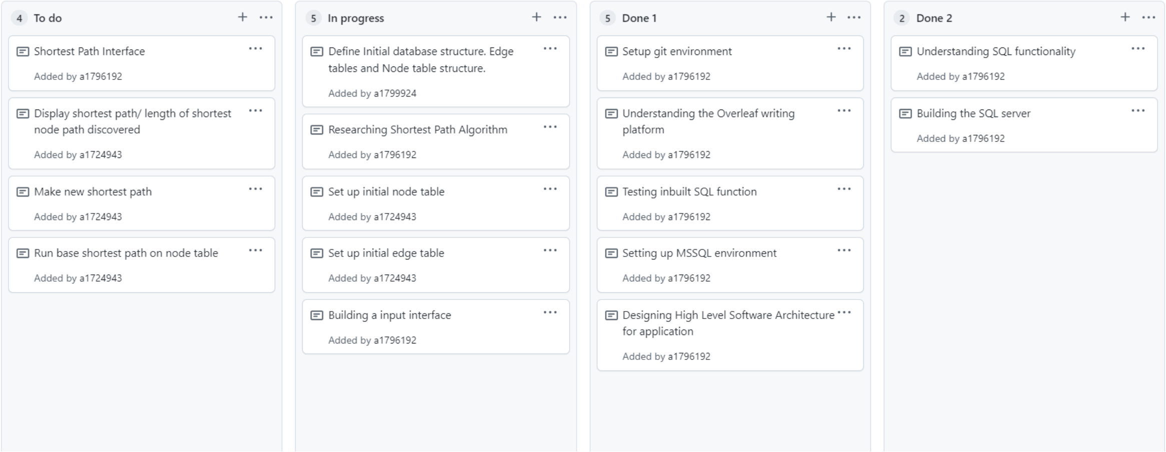
As a user I want to mark devices to exclude, so that shortest paths can be identified avoiding them.

Added by a1809183

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

Added by a1809183

Figure 1: Product Backlog



4 To do

Shortest Path Interface

Added by a1796192

Display shortest path/ length of shortest node path discovered

Added by a1724943

Make new shortest path

Added by a1724943

Run base shortest path on node table

Added by a1724943

5 In progress

Define Initial database structure. Edge tables and Node table structure.

Added by a1799924

Researching Shortest Path Algorithm

Added by a1796192

Set up initial node table

Added by a1724943

Set up initial edge table

Added by a1724943

Building a input interface

Added by a1796192

5 Done 1

Setup git environment

Added by a1796192

Understanding the Overleaf writing platform

Added by a1796192

Testing inbuilt SQL function

Added by a1796192

Setting up MSSQL environment

Added by a1796192

Designing High Level Software Architecture for application

Added by a1796192

2 Done 2

Understanding SQL functionality

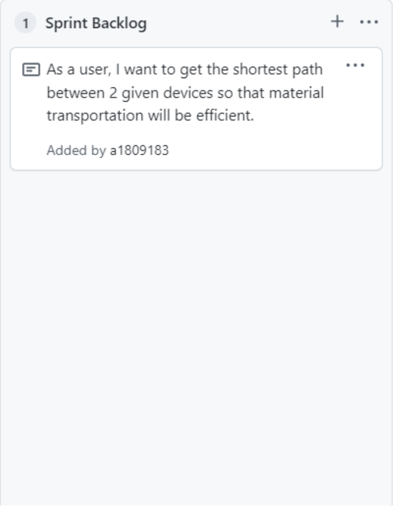
Added by a1796192

Building the SQL server

Added by a1796192

Figure 2: Task Board

Sprint Backlog and User Stories



1 Sprint Backlog

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

Added by a1809183

Figure 3: Sprint Backlog

After much deliberation, the user story had been changed and instead, the story to solve will be "I want to get the shortest path between 2 given devices so that material transportation will be efficient". We concluded that the previous user story would be much more difficult to implement if we did not have a grasp on the design of the node tables themselves hence the adding and removing would be irrelevant.

The sub tasks for the new user story that we have selected will direct the focus on building an early model of the problem. The user provided a set of data, hence, the team will focus on developing a method to integrate input data into MSSQL. In order to manipulate the given data to achieve the shortest path, the team will develop a method to implement node and edge tables in MSSQL. Finally, the team will delve into researching about the optimal shortest path algorithm to be used for the given problem.

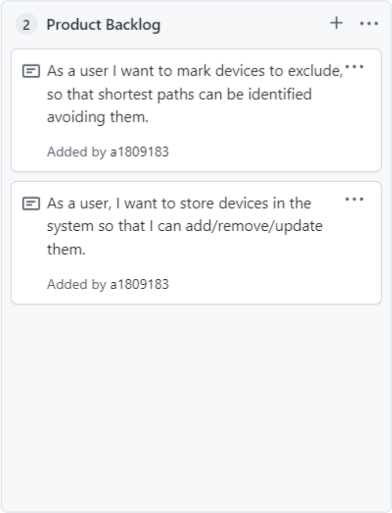
Definition of Done

- If code has been written, it must follow the coding standards defined in the initial report.
- If code has been written, it must be reviewed by members to ensure proper functionality and appropriate software design, as well as its relation to the product backlog.
- If code has been written, it must pass localization testing defined by the programmer, automated testing written by the team, and automated regression testing for any future alterations to ensure correct behaviour.
- If code has been written, it must ensure functionality continues to work on build server, and local device.
- The created sprint task must be linked towards the product backlog which in turn must be linked towards the user story.
- Any feedback provided by the client, product owner, and scrum team should be analysed for practicality/feasibility and implemented once deemed beneficial to the user.
- For any implementation of code or system architecture, sufficient documentation is required to justify its relation to the user story, and configuration changes must be well-documented.
- The documentation and code must be deemed acceptable by the criteria given by the product owner, as well as basic software/engineering standards, and be signed off by the product owner.
- Verification and validation of key scenarios are considered to ensure basic functionality before in-depth processes are considered for implementation.
- The final product must address the provided user stories to a reasonable standard, being able to at the very least produce a shortest path between two specified devices.

Summary of changes

Since the last snapshot, the user story to solve has been changed from "I want to store devices in the system so that I can add/remove/update them" to "I want to get the shortest path between 2 given devices so that material transportation will be efficient". The group has divided itself into smaller groups handling nodes and edge tables, integrating data into MSSQL, and starting research into the shortest path algorithm.

Snapshot 2.2

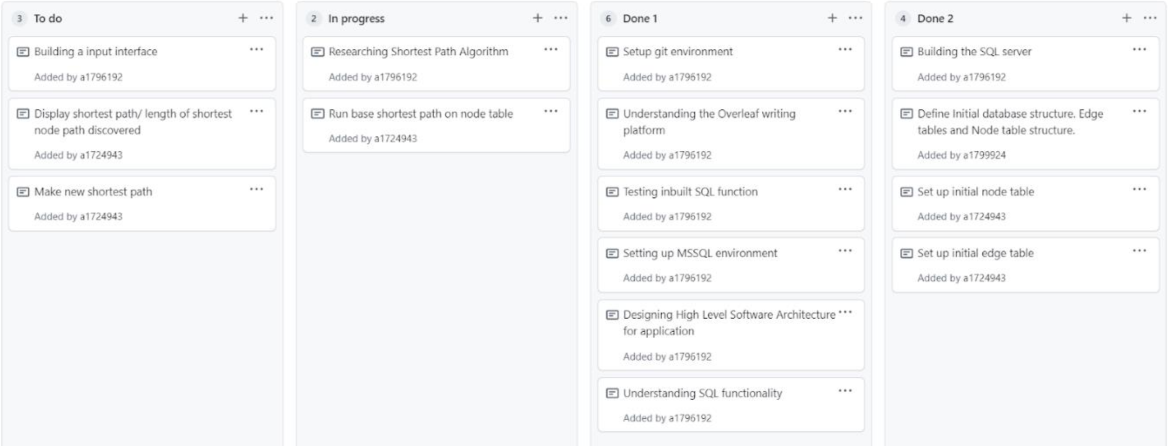


2 Product Backlog

As a user I want to mark devices to exclude, so that shortest paths can be identified avoiding them.
Added by a1809183

As a user, I want to store devices in the system so that I can add/remove/update them.
Added by a1809183

Figure 1: Product Backlog



3 To do

Building a input interface
Added by a1796192

Display shortest path/ length of shortest node path discovered
Added by a1724943

Make new shortest path
Added by a1724943

2 In progress

Researching Shortest Path Algorithm
Added by a1796192

Run base shortest path on node table
Added by a1724943

6 Done 1

Setup git environment
Added by a1796192

Understanding the Overleaf writing platform
Added by a1796192

Testing inbuilt SQL function
Added by a1796192

Setting up MSSQL environment
Added by a1796192

Designing High Level Software Architecture for application
Added by a1796192

Understanding SQL functionality
Added by a1796192

4 Done 2

Building the SQL server
Added by a1796192

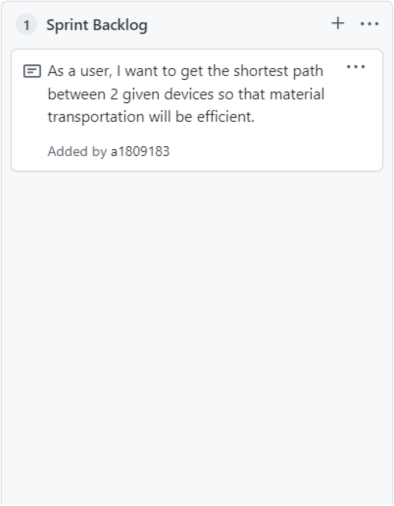
Define Initial database structure. Edge tables and Node table structure.
Added by a1799924

Set up initial node table
Added by a1724943

Set up initial edge table
Added by a1724943

Figure 2: Task Board

Sprint Backlog and User Stories



1 Sprint Backlog

As a user, I want to get the shortest path between 2 given devices so that material transportation will be efficient.
Added by a1809183

Figure 3: Sprint Backlog

After much deliberation, the user story had been changed and instead, the story to solve will be “I want to get the shortest path between 2 given devices so that material transportation will be efficient”. We concluded that the previous user story would be much more difficult to implement if we did not have a grasp on the design of the node tables themselves hence the adding and removing would be irrelevant.

The sub tasks for the new user story that we have selected will direct the focus on building an early model of the problem. The user provided a set of data, hence, the team will focus on developing a method to integrate input data into MSSQL. In order to manipulate the given data to achieve the shortest path, the team will develop a method to implement node and edge tables in MSSQL. Currently the selected user story is the shortest path algorithm. The current task is to conduct research on shortest path techniques in SQL and what mathematical equation to implement. Just a simple model needs to be made to read the minimum information required to find the shortest path. The shortest path algorithm must get data from the node and edge table and hence those data platforms must be built.

Definition of Done

- If code has been written, it must follow the coding standards defined in the initial report.
- If code has been written, it must be reviewed by members to ensure proper functionality and appropriate software design, as well as its relation to the product backlog.
- If code has been written, it must pass localization testing defined by the programmer, automated testing written by the team, and automated regression testing for any future alterations to ensure correct behaviour.
- If code has been written, it must ensure functionality continues to work on build server, and local device.
- The created sprint task must be linked towards the product backlog which in turn must be linked towards the user story.
- Any feedback provided by the client, product owner, and scrum team should be analysed for practicality/feasibility and implemented once deemed beneficial to the user.
- For any implementation of code or system architecture, sufficient documentation is required to justify its relation to the user story, and configuration changes must be well-documented.
- The documentation and code must be deemed acceptable by the criteria given by the product owner, as well as basic software/engineering standards, and be signed off by the product owner.
- Verification and validation of key scenarios are considered to ensure basic functionality before in-depth processes are considered for implementation.
- The final product must address the provided user stories to a reasonable standard, being able to at the very least produce a shortest path between two specified devices.

Summary of changes

Since the last snapshot on 28/08/2023, members have been divided into teams of 2 and 3 and are assigned to work on the sprint tasks. Current focus in previous snapshot 2.1 was to focus on research on shortest path algorithms and SQL environments. Currently we have implemented a basic form of shortest path algorithm implemented by the SQL software. Given basic information such as node names and data, the system can find the shortest path. Furthermore, given the example inputs given, we can manipulate the data and filter the data for the desired outputs. However, data could only be changed manually through the server entries and not the SQL framework. We constructed edge and node tables by creating specific csv files with the appropriate headings. This was done to test the initial shortest path algorithm.

"I attended the sprint review/planning meeting on Thursday the 7th of September with the tutor Dileepa Pitawela"

Sprint

What went well?

In this sprint we were more efficient with identifying and splitting tasks and workload among members of the group. The product owner gave us some data in an excel file which allowed us to proceed further than we could in the previous sprint. This included tasks such as getting the data into MSSQL, setting up initial node and edge tables, researching shortest path algorithms, and organising snapshots. In the first group meeting of the sprint, we split these tasks among 2–3-man teams depending on the difficulty. We were able to get the data into MSSQL fairly quickly by converting it into a csv file and importing that, which allowed other group members to get started on initial node and edge tables. Since we're currently being told to assume a node and edge cost of 1, we decided that for now it would be best to use the in-built MSSQL shortest path function on our tables and data. By the second week of the sprint, we had a working demo which runs Shortest Path taking advantage of created node and edge tables, returning one shortest path between two devices.

What could be improved?

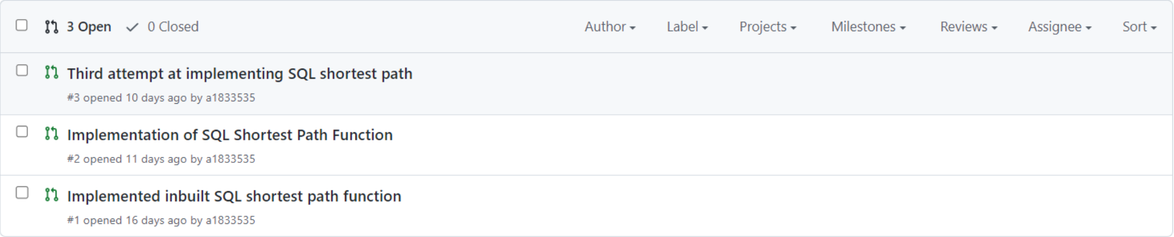
After we had split into our separate teams to work on tasks, there wasn't as much consolidation as there could have been. We became focused on our own tasks and made good progress, however, although the tasks were fairly linked, the final demo wasn't a culmination of all we had done within the sprint. For example, once we had imported the data from the product owner into MSSQL and created tables from it, we could have tried to coordinate better so that those tables and data were in the final demo instead of new ones created from scratch. Another possible issue that arises from our segmented approach is that some team members might get very familiar with one area such as shortest path algorithms, SQL, or the factory layout, instead of having some understanding across all areas which may make it harder to communicate about what has been done and decrease the chance of the team completely understanding the final product.

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

The goal we set during the sprint planning meeting was that in the next sprint we would try to take our current demo and have it work with updated values. This means that we need to be able to change the layout of the factory given new data. As of now the product owner has said a user interface is not necessary and instead we will demonstrate the new functionality by manually changing the values in the tables and re-running the algorithm. We should add more attributes so that our current demo tables more-closely resemble those in the excel tables given by the product owner, such as attributes denoting whether a plant item is a source or a destination, and the cost of nodes and edges, and exploring the possibility of writing a script or code to do this automatically. We should also continue to look into shortest path algorithms and whether the in-built MSSQL will be sufficient for the cases of varying node costs and for returning five shortest paths, potentially looking to python and MySQL for this and to explore handling user input.

Sprint Progress

The only pull requests our group currently has on the GitHub was around the upload of the demo using the in-built MSSQL shortest path function on simple node and edge tables.

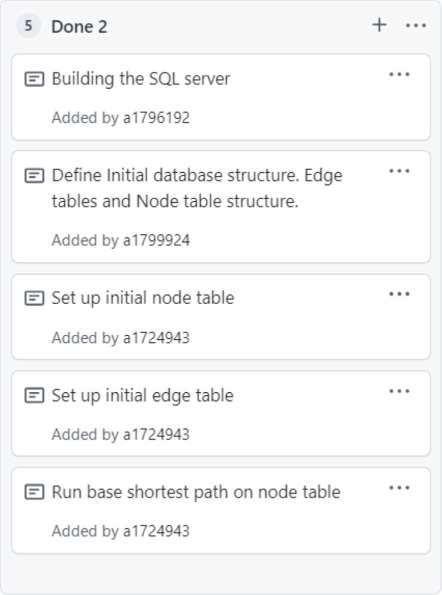


□ Third attempt at implementing SQL shortest path
#3 opened 10 days ago by a1833535

□ Implementation of SQL Shortest Path Function
#2 opened 11 days ago by a1833535

□ Implemented inbuilt SQL shortest path function
#1 opened 16 days ago by a1833535

I participated in discussion about improving workflow and efficiency by splitting tasks better and assigning people based on difficulty. This sprint I was responsible for taking the data from the product owner, importing it into the MSSQL environment. MSSQL wasn't accepting the excel file, so it had to be converted into a csv file before being imported as a flat file. I researched more into SQL and the functionalities provided by MSSQL and modified the imported data into a form more suitable of that for determining shortest paths. This included tasks such as adding attributes which would be needed such as node ids and modifying attributes such as Plant Item with entries such as "Conveyer12" or "Diverter6" to instead be defined by ids and just retain "Conveyer" or "Diverter" as the type of plant item they were. I also worked on the second snapshot with two other group members.



Building the SQL server
Added by a1796192

Define Initial database structure. Edge tables and Node table structure.
Added by a1799924

Set up initial node table
Added by a1724943

Set up initial edge table
Added by a1724943

Run base shortest path on node table
Added by a1724943