Criterion E

E1: Evaluation against Success Criteria

Success Criteria	Met	Explanation
The program should have a hierarchy of login credentials: Managers and drivers using a dropdown menu with Netbeans.	✓	- The program login page provides a dropdown menu where the login user type can be chosen. - This links to the login credentials and user types stored in the MySQL database connected to NetBeans
The program validates the user input against the stored login credentials	✓	- Through the use of the MySQL database and IF/ELSE loops, this success criterion is achieved as it repeatedly compares the user's input to the stored login credentials to validate the login. - If the comparison proves the user input to be incorrect, the login access is denied and the user is notified.
The program displays a dialogue box for correct and incorrect log-ins.	✓	- A dialogue box is displayed to the user informing them of the correct or incorrect login input
The program provides two panels depending on the user's credentials: Manager or Driver.	✓	- After the login credentials are validated against the credentials in the MySQL database, the program displays different panels depending on the user type provided by the user.
The program allows the driver user to input several inputs such as; Driver Name, Truck ID, Trailer ID, Nearest city and Journey updates	✓	- As demonstrated in Criteria D, the GUI allows for inputs of different data types from the user and has a clear function to improve the program's usability.
The program handles incorrect input like incorrect data types for the driver's input and notifies the user of the incorrect data type.	✓	- A dialogue box is displayed informing the user of the incorrect data type input.

The program stores all of the driver user’s inputs in the MySQL database.	✓	- This success criteria is achieved through the connection to the database and the use of the “INSERT INTO” method to store the driver’s inputs in the appropriate column in the MySQL database.
The program allows the manager user to select the criteria for the data displayed through buttons in the GUI. The criteria are; Total capacity, List of active drivers and trucks, Drivers and their nearest location and Drivers and their truck and trailer ID.	✓	- This success criteria is achieved through creating a menu using Netbeans in the Manager panel, which allows them to select which criteria they want to view. This determines which data is fetched from the MySQL database and displayed in tabular form to the user
The program should allow the manager to view the data using a table displayed in Netbeans.	✓	- Use of JTable to display data fetched from MySQL database.
The program displays a pie chart, showing the total available capacity for active trucks.	✓	- applying JFreeChart - The program uses a count to tally the number of active and inactive trucks, which is then used to calculate the pie charts' parts and the associated values.
The program allows the manager to refresh the table displayed to show new driver inputs.	✓	- When the refresh button is clicked, the try-catch statement fetching the data from the MySQL database is run again, updating the JTable in Netbeans.

E2: Client feedback

After providing the client with the program for a week, I scheduled a meeting with him where he gave me feedback on his experience with the program and whether it met his user requirements (Appendix 2). Overall, the client was pleased with the functionality of the program and stated that it met his user requirements. More specifically, the client found the Pie Chart display of the total capacity to be his favourite feature due to its utility in day-to-day operations. In addition, the client was quite satisfied with the program’s ability to display the inputs from the drivers in tabular form as it allowed for faster consolidation of data and faster decision-making and coordination of drivers. According to the client, the display of journey updates allows him to stay informed on his driver's situation and also allows him to address any issues that may arise without calling or texting them as they drive. (Appendix 2)

Despite the fact the client stated that the program met all his requirements, he suggested possible improvements that would improve the functionality and usability. One notable piece of feedback was on the refreshing of the tables and how manually refreshing still limits him as he often forgets to refresh the table to check for new driver inputs. He also mentioned how time-consuming looking through the tables for specific locations was despite being drastically easier than his previous system.

E3: Future recommendations

- To further improve the functionality, the client would like automatic real-time updates to be implemented as he believes it would be easier to view the data without the need for manual refreshing, which would enhance the user experience (Appendix 2). Although the current program updates the tables by the user clicking the refresh button, the client argues that automatic updates would improve the accuracy of the tables when users look at the data displayed. To implement this improvement, I would add a scheduled task in the background of the program - which periodically queries the SQL database updating the displayed table. By programming the scheduled task to execute SQL queries, the program would automatically update the tables without the user's input.

- Another improvement the client discussed with me is the implementation of additional data visualizations such as bar charts or line graphs to help visualize the number of drivers in a specific location which would be used in department and company meetings (Appendix 2). To implement this improvement, I would use the JFreeChart library on Net beans coupled with specific SQL queries to allow additional data visualizations such as bar and line graphs to help improve the visualization and functionality of the program. Additionally, including the real-time updates discussed before and the ability to tailor the visualizations of the charts would also ensure clarity and accuracy when displaying the data to the managers.

- Lastly, the client would also like additional documentation and help resources within the application to assist new users, such as newly employed managers, in understanding the application and addressing any potential issues (Appendix 2). To implement this improvement, I would implement contextual help across the program by integrating information buttons that would provide the user with an explanation or guidance as the manager or driver hovers over a specific function of the program. Additionally, the addition of a user guide at the beginning of the program would provide both drivers and managers an adequate source of information to aid the navigation of the program.