# Formalized Proposal

The goal is to solve a problem in the company I work for when it comes to deliveries. The hope was to create a program that would allow for tracking inventory items, assigning those inventory items to deliveries to be made by employees, and print out receipts for them to use in making these deliveries.

# Time/Change Logs

Initial time of project creation was 4/01/2023.

04/04/2023 - I had made the inventory class with constructors, getters, and setters.

04/09/2023 - I had made the delivery class with constructors, getters, and setters, and was in the process of figuring out how to create a schedule.

04/16/2023 - I had made the majority of interior logic for the program, this is where I made my first mistake of involving Qt.

04/23/2023 - I had figured out Qts UI creator, but I was struggling get things to connect properly to the meat of the program.

04/27/2023 - I had become fed up with Qt. I had spent the entire week troubleshooting the connect function. I made some progress, but eventually reached a point where the program would crash upon pressing any button. Wasted 10+ hours on Qt between set up and troubleshooting.

04/28/2023 – I scrapped Qt, took a half-day at work and spent 6+ hours creating a CLI in its place. I finished up some last minute logic and CSV parsing.

# Lessons Learned

The scope was to be a linked list full of inventory items, and a priority queue with deliveries that also had a linked list of inventory items meant for delivery. I kept the structure I made for the inventory list and eliminated the master inventory list, and instead kept it as a variable for the delivery priority queue.

A lesson I learned is that, though I may enjoy coding in C++ the most, its lack of a GUI system makes it ill suited for certain jobs/projects. I had first attempted to get GTK (A Gui System for C++) working for this project first. After 3 days of troubleshooting that, I switched to Qt. Things seemed to work initially, but began giving me odd errors, especially when I went to try parsing CSV files. The command to check for an end of file in C++ would never execute in Qt and result in an infinite loop, though works perfectly in VSCode.

Another lesson would be to check unknown variables earlier and have less tolerance for encountered problems. If I had scrapped Qt sooner, I might have been able to deliver a CLI that kept the inventory system intact and correctly debited it with more user input verification.

# Code

<https://github.com/trevor-tfe/Final-Project>

# User’s Manual

A screenshot of a computer

Description automatically generated with medium confidence

First, download the contents of the folder labelled FinalProjectCLI. Once you’ve done so and run “main.cpp” you should be greeted with the following information.

A screen shot of a computer program

Description automatically generated with low confidence

It is fairly self-explanatory, but I would like to add that entering q or Q functions the same, but only at the main menu or sub-menus. To enter a new delivery, enter “1”.

A screenshot of a computer

Description automatically generated with medium confidence

You will be asked if you would like to enter a new delivery. Selecting y or Y will send you through the delivery process. Entering n or N will return you to the main menu. Entering q or Q will immediately exit the program. After following the instructions, you will be asked if you would like to add any inventory items to the delivery. Entering n or N will skip this step.

A screenshot of a computer

Description automatically generated with medium confidence

You can continue to add items to the delivery in this manner until the entire delivery is set up. After which you will be asked if you would like to schedule the delivery. Entering n or N will skip that step.

A screenshot of a computer screen

Description automatically generated with low confidence

After entering your deliveries, you can return to the main menu to either View the next delivery by entering 2, or print the delivery receipt for employees to begin organizing the order and giving to the customer after delivery completion. Printing the receipt removes the delivery from the queue and prepares the next delivery.

A screenshot of a computer

Description automatically generated with medium confidence

A receipt text file should look like this after printing.

# Conclusion/Summary

Both Deliveries and Inventory practice encapsulation methods of constructors, getters, and setters for data. Loops are used sparingly, and nesting is kept to a minimum to keep time complexity to a minimum. However, in some cases it is impossible to avoid. Most user input is handled via characters or string to int conversion to prevent data mishmash. I’m sure it is possible to break it, but I made sure to allow spaces for it. Since a CLI is used, I made sure to instruct the user on what is expected at each step of input. In cases of readability, I made sure to properly indent, and comment as much as I could think to without bogging down the program reader with unnecessary details every line. Where I could, I also attempted to break chunks of code down into smaller segments for better readability and elegance.