Stefano Coronado
CPSC 240
Project 2 – Test Cases (Updated to complement submission)

In the increment of Project 1 – I lifted the interface from a commandline to a GUI in JavaFX. A graphical window should appear with six different tabs.

The spec asks for dynamic amount of warehouses and salesVans. When a warehouse is created, that will be an instance of a Warehouse object. In the other tabs, the amount of options in the combo boxes should increment with the name of the new warehouse.

This is now possible.

I should be able to sell a bike part by its part number. The way I do that is by entering the part number and selecting the warehouse to sell that from the combo box. Success of this means that the part number decrements and that the requested details of the sale are displayed to the user.

Entering a Bike Part is similar. The idea is to fill in the details about the BikePart and select the warehouse to drop it in. Clicking the “Enter” button will add it into the selected warehouse.

A longstanding bugfix inherited from Project 1 is that the program will remove the entry from the warehouse inventory if it already exists by name and number, then it will augment the quantity, eliminating redundant entries.

I want to be able to see BikeParts based on what warehouse they are in. The ComboBox allows Java to understand the difference between different warehouses. When I select a warehouse and click “Update List”, a different list should appear. This list will allow for sorting depending on the checked option selected (by Name or by number). There is also a option to be able to sort through the list by number, just click on the check box and tap the update button in the Display view.

Transfering parts between warehouses will identitify warehouse instances by Strings in the text file and move the respective item by name to the destination warehouse instance. If the part never existed there before, a new part is created. If the part is already there, the count is augmented. The user should be able to see the contents of the file on-screen to confirm that this has been done, they can also check the Display tab.

Importing a delivery file will import BikeParts from a text file into the warehouse instance chosen in the combobox. The user should be able to see the contents of that file in the to confirm that this has been done. They can also check the display tab to see if their inventory got updated.

The program is now able to save parts to to their respective __warehouseDB.txt files, the file named “warehouseDB.txt”. After much debugging during testing, the program will now be able to save all of inventories on exit.