Grocery Store Manager

Final Report

**Nicholas Board**

**CSC316: Data Structures for Computer Scientists**

**ndboard@ncsu.edu**

**North Carolina State University**

**Department of Computer Science**

**<Date: February 3rd, 2016>**

# Challenges

Briefly describe 2 major challenges you encountered during this project and how you responded to each challenge. (1-2 paragraphs)

For example:

* Did your build time-out in Jenkins and force you to find a more efficient solution?
* Did you set aside enough time to work on the project?
* Did you have trouble with file input/output?
* Did you have trouble writing JUnit test cases?

With this project, I discovered just how dependent I was on the TS design in 216. When I saw the UML for this project and saw that it only listed suggested classes and not the state and methods of the class, it took me a long time to be able to start properly. I did the same thing I did with the 216 projects and started with the most basic classes (Customer and Product) and worked my way up. Also, after doing a lot of not OO programming in C, I had a hard time going back to OO. I had to relearn some of the basics (especially when dealing with input), and just generally get used to Java again.

I did not start on this project as soon as I should. I was assigned a project in this class and one in C at roughly the same time, and I prioritized the C one because I thought it would be harder. Turns out it was leagues easier than this project, so I barely had my project started three days before it was due, and only got it functional the night before. Nonetheless, I was surprised at how much I DID end up getting done in this short amount of time.

# Data Structures & Algorithms

Did you use the same data structures and/or algorithm(s) you selected in your proposal? Why or why not? (2-3 sentences)

No, in the proposal we used selection sort, mainly just because it was the easiest one to analyze the pseudocode of for part one. In the actual project, I’m using the quicksort because it’s, well, quicker. I was also considering merge sort. I ended up using a non-recursive version of quicksort based off an algorithm I found, because it intrigued me and seemed easier to actually implement.

# Improving Efficiency

Describe at least one way you could further improve the efficiency of your project software, based on topics we have discussed in class. (2-3 sentences)

I don’t really know how to make my getProduct method more efficient, even though it’s failing some of the larger input tests. To get the frequency, it must access every item in the array to check if it’s a matching product. So, no matter what it’s going to be O(n). Only thing I can think of is reducing the number of constant operations, as well as somehow maybe reducing the access of each object’s brand and description for comparison.

# Moving Forward

Briefly describe one thing you will change or improve upon as you work on the next project. (2-3 sentences)

I want to start earlier, plan my program with my own UML before starting to code. Document better while coding so I don’t have to go back. I spent most of my time on this project just warming back up to the Java basics. Now that I’m back in the flow of it, next time I want to spend more time thinking about algorithms, data structures, and improving efficiency rather than just madly trying to get the thing to work.