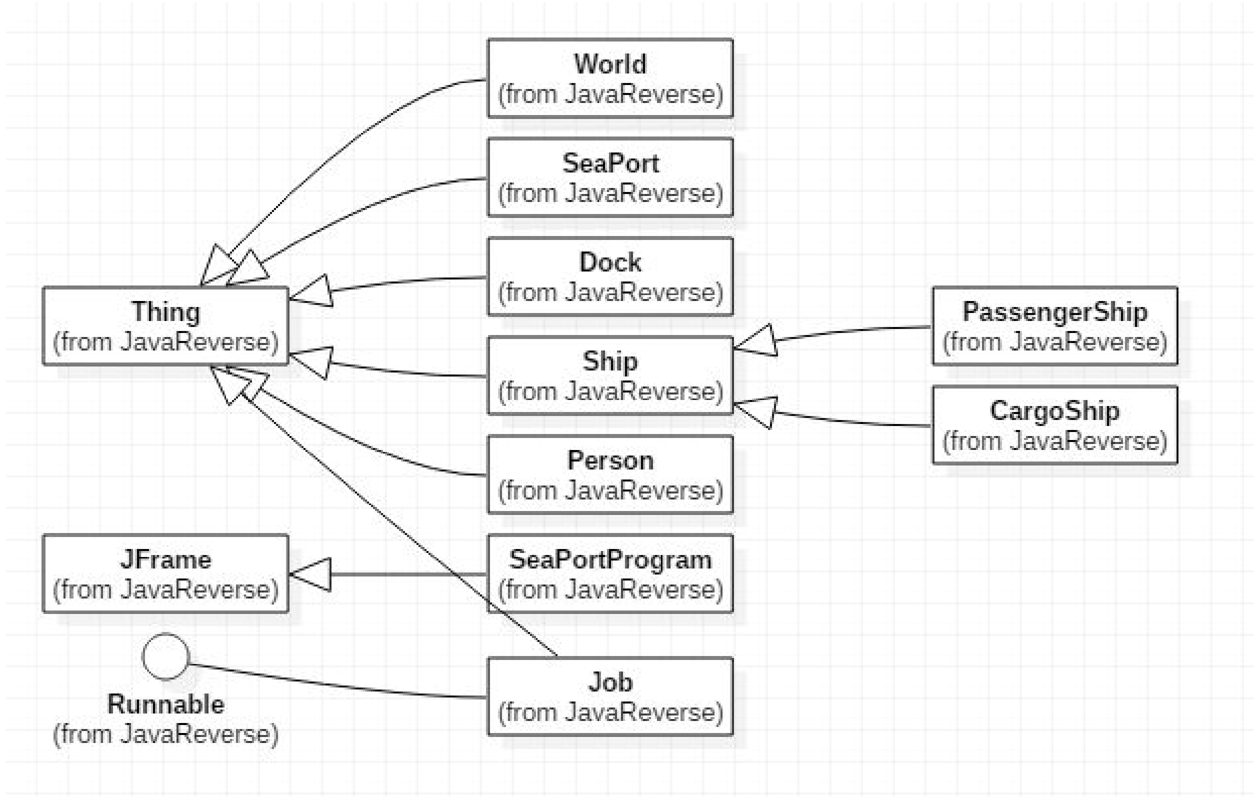
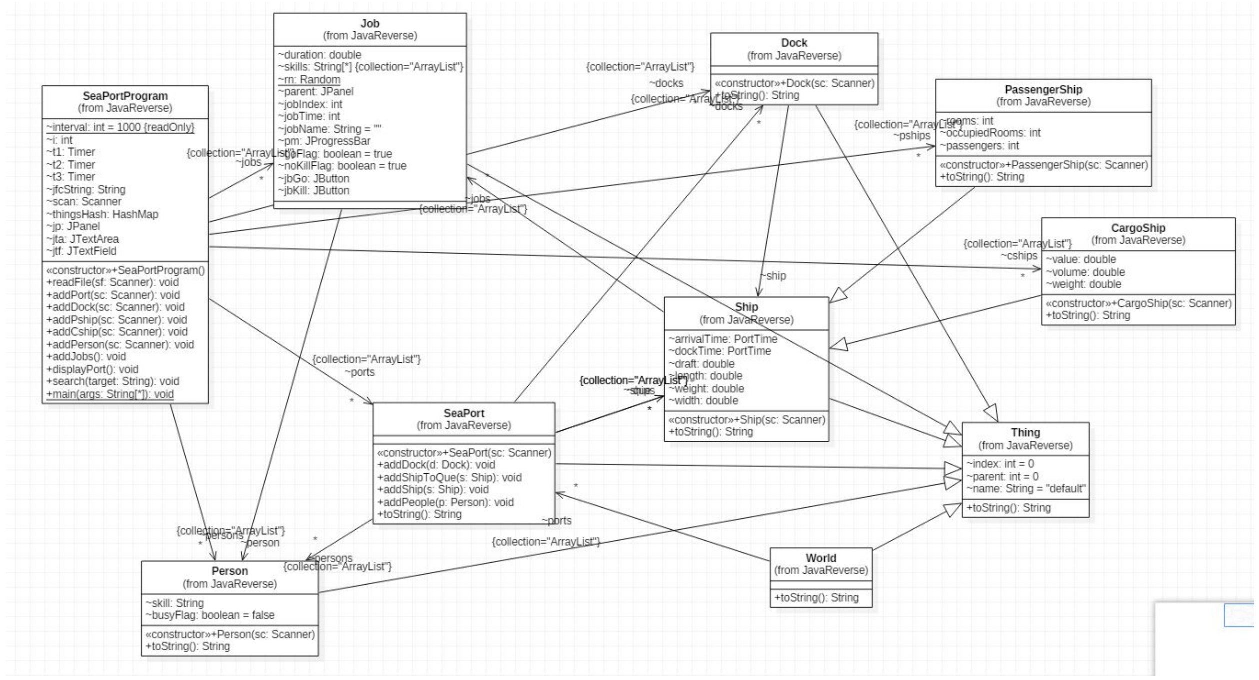
David Robbins July 30, 2017
CMSC 335 Project 3

Design
Simple UML Flow diagram



More Detailed UML Diagram



Both of these PNG files are included in zip file in case they are hard to read

User's Guide

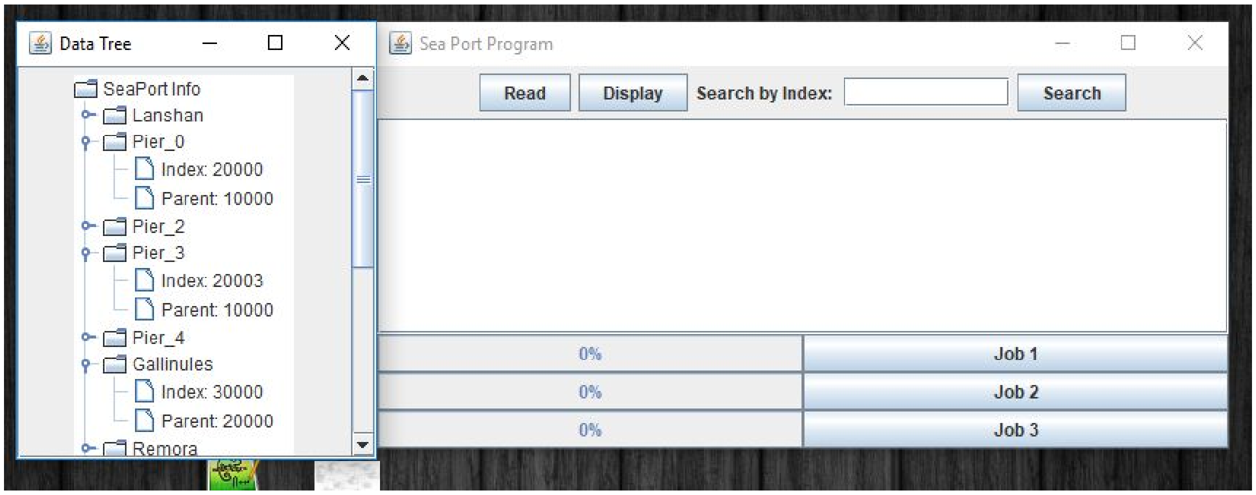
1. Open the JAR file named "Project" (this opens the program)
2. Click the "Read" button (this creates a separate GUI with the data in a sorted JTree)
3. Click the 'key' to drop down the needed object to view index and parent
4. Click 'Display' to view all of the SeaPort information
5. Once you have located the index of the object you want to know more about in the JTree, insert that index in the text field labelled 'Search by Index' then click 'Search' (this will clear the text area and fill in the information you want.
6. To view examples of mock jobs being done, click any of the Job buttons and a loading bar will fill at varying speeds. Once finished a job can be started over

Test Plan

My original plan for this project was to take what I had created and do my best to understand the material and implement it into my program. After about five days of failed attempts at this, I came to the realization that the way I was setting up my data structure, storing strings instead of objects in many instances and over all not following the guide provided to us via the Cave project example. This led me to a point where I said screw it and started over, basically making my way through project 1 and 2 all over again while this time using the Cave example as a sort of skeleton. After doing this, making a class to create the JTree GUI was simple. I tried many times to understand how to implement the Job object in a way that worked like is asked in the instructions but was unable to and ran out of time, so instead opted to show my understanding of how to use the JProgressBar through a few examples with buttons to activate them. While this did not fulfill the needs of the project, it at least showed some understanding.

Since I basically redid project 1 and 2, I did test some of the past projects functionality in my test cases, as well as the new functions with the JTree and JProgressBar objects.

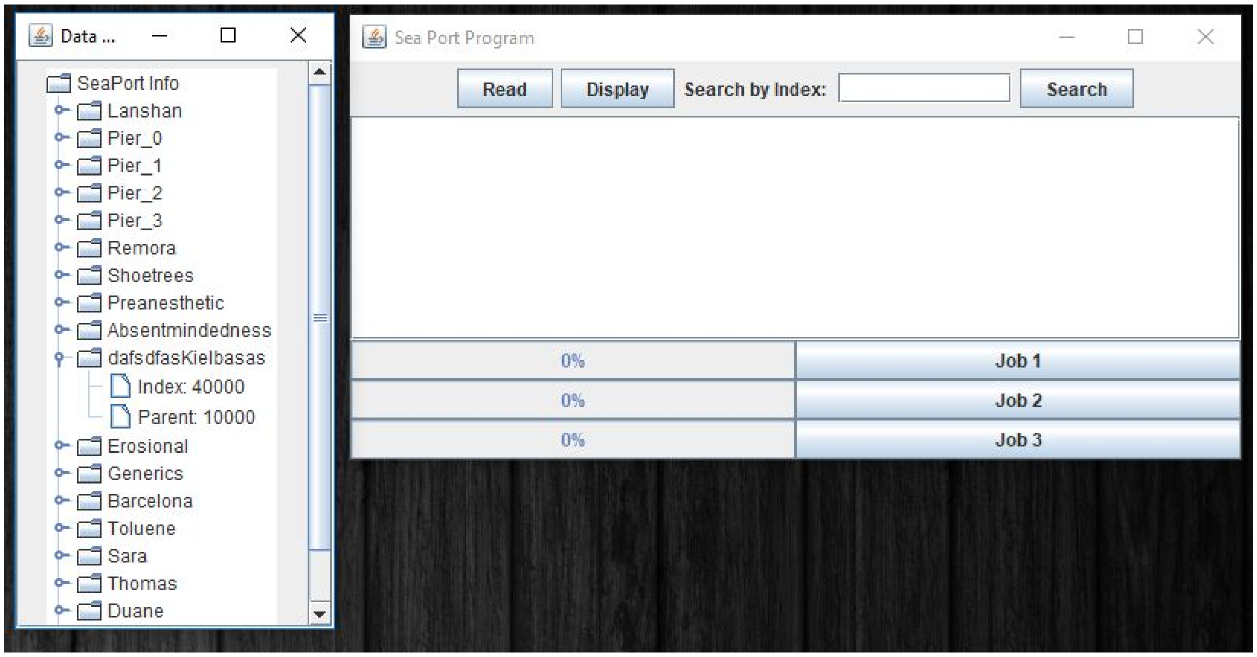
Test 1 - Success
Testing to see if the program successfully reads a text file with the correct format, and if so, if the JTree GUI is created correctly



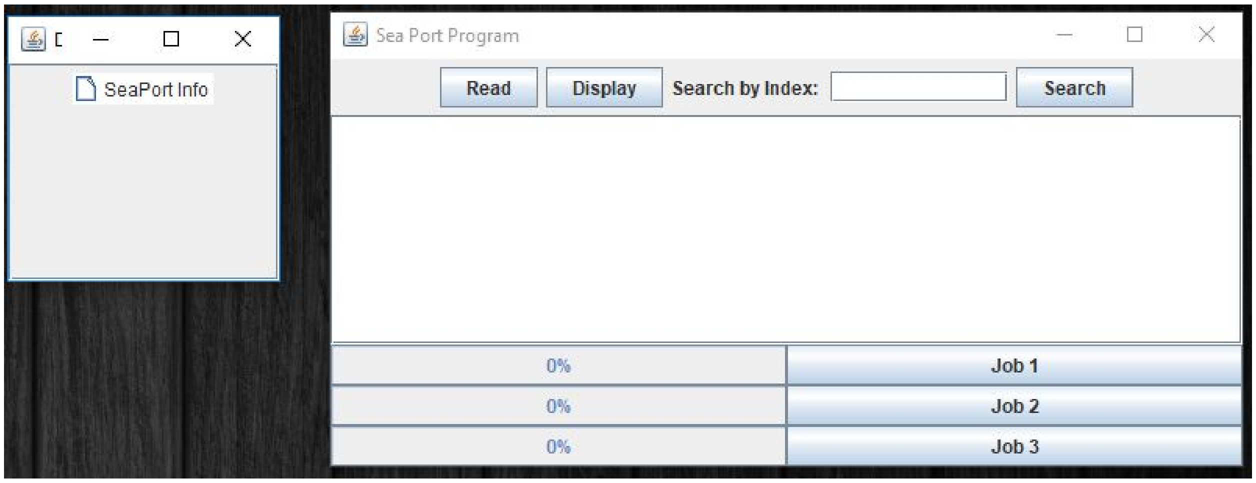
```
SeaPort Info
  Lanshan
  Pier_0
    Index: 20000
    Parent: 10000
  Pier_2
  Pier_3
    Index: 20003
    Parent: 10000
  Pier_4
  Gallinules
    Index: 30000
    Parent: 20000
  Remora
```

Test 2 - Success

Testing to see what happens when the program reads a file that is not a text file, or is the incorrect format. When it is not a text file, the JTree populates nothing, when it is a text file with the wrong format, the program still pulls information where it can find it and ignores parts that are formatted incorrectly, so it is resilient to errors in some ways

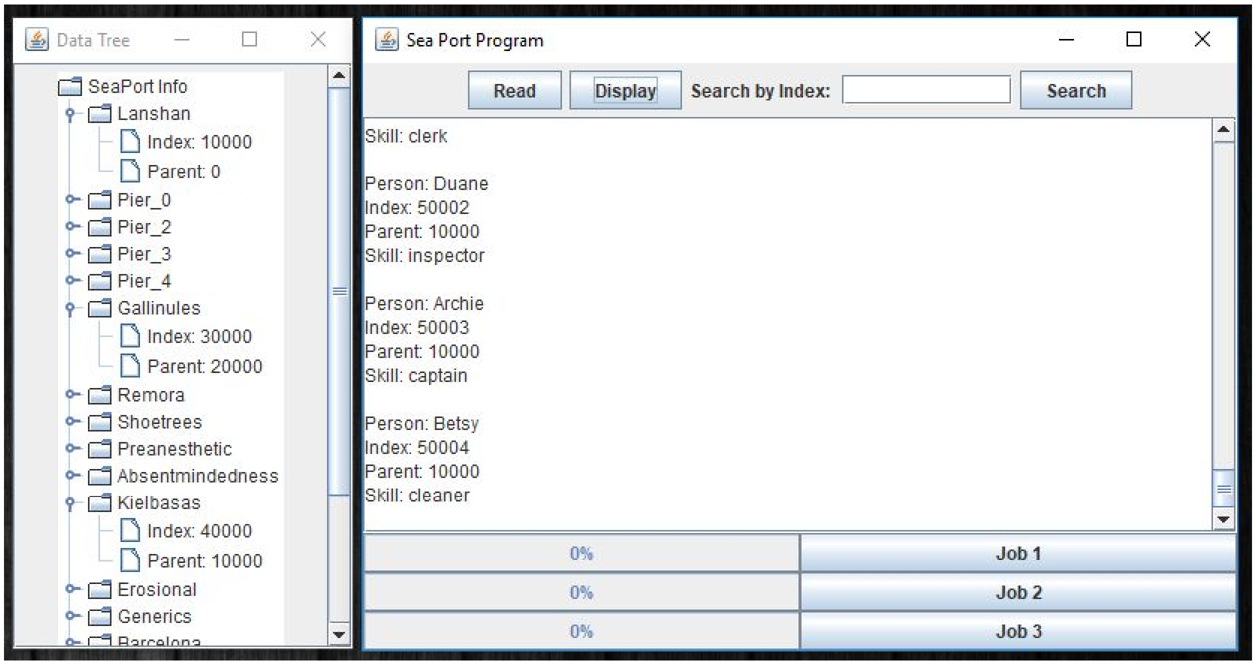


```
SeaPort Info
  Lanshan
  Pier_0
  Pier_1
  Pier_2
  Pier_3
  Remora
  Shoetrees
  Preanesthetic
  Absentmindedness
  dafsdfasKielbasas
    Index: 40000
    Parent: 10000
  Erosional
  Generics
  Barcelona
  Toluene
  Sara
  Thomas
  Duane
```

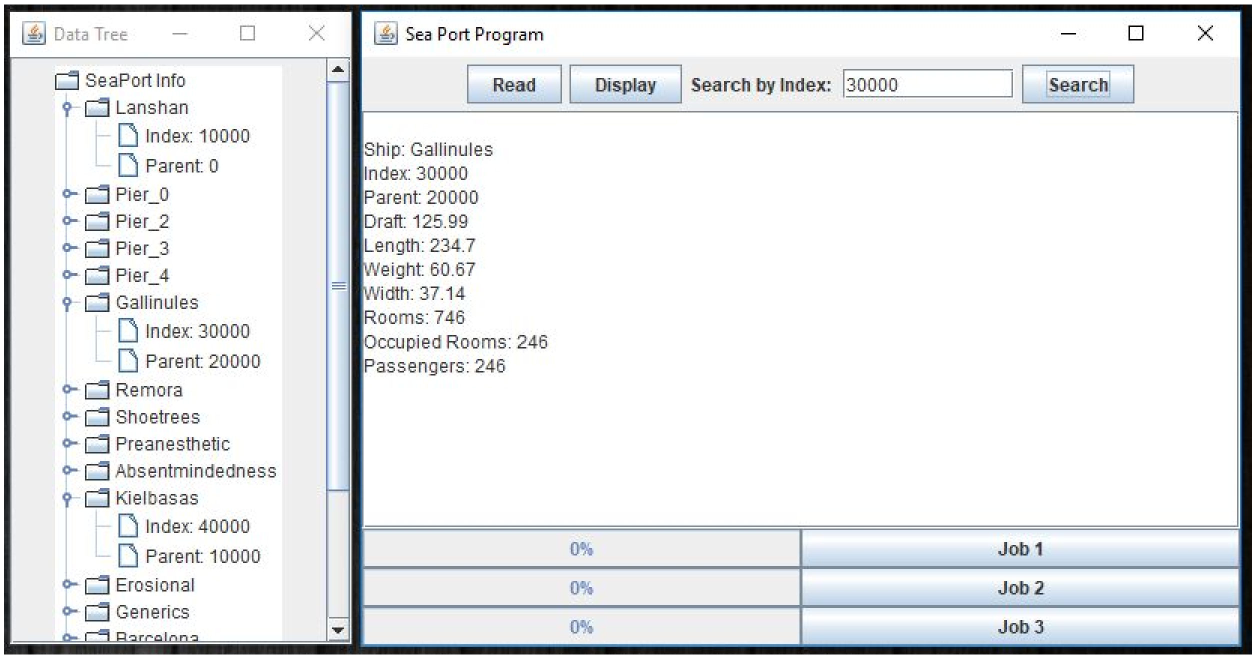


Test 3 - Success

Testing to see if the Display and Search buttons work as they should



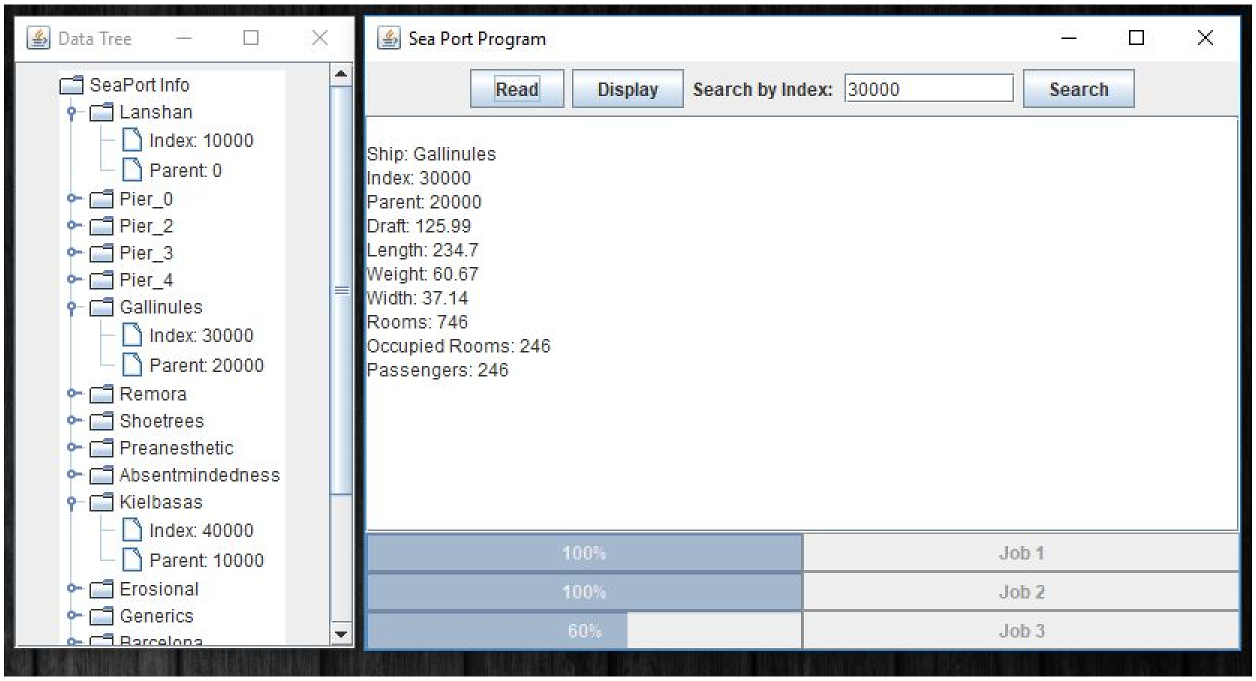
```
Skill: clerk

Person: Duane
Index: 50002
Parent: 10000
Skill: inspector

Person: Archie
Index: 50003
Parent: 10000
Skill: captain

Person: Betsy
Index: 50004
Parent: 10000
Skill: cleaner
```



```
Ship: Gallinules
Index: 30000
Parent: 20000
Draft: 125.99
Length: 234.7
Weight: 60.67
Width: 37.14
Rooms: 746
Occupied Rooms: 246
Passengers: 246
```

Test 4 - Success

Testing to see if the Job buttons activate the progress bars



```
Ship: Gallinules
Index: 30000
Parent: 20000
Draft: 125.99
Length: 234.7
Weight: 60.67
Width: 37.14
Rooms: 746
Occupied Rooms: 246
Passengers: 246
```

Lessons Learned -

This whole class has been quite the learning experience and this project was no exception. Most of what I learned is fairly easy to glean from my Test Plan. The fact that it wasn't so much a plan as much as a try this, try that, fail and fail again until I became so frustrated I started all over two days before it was due, gives a good idea of how I felt about this project. I spent a great deal of time just in creating the JTree, and it was while creating this with my previous Project 2's work that I came to realize the way my data structure was set up would never work for what was being asked for. That is what caused me to finally decide to turn back and start over from the beginning. I think my program now in its current state will be much easier to implement the Job threads, although I wasn't able to complete that. My take away from this week's project was learning how to use and implement the JTree as well as the JProgessBar, but I am still very fuzzy on how to use the synchronize directive as well as how to use Threads in a way that matches with what was asked for in the project instruction. I would appreciate any feedback on how I can adjust my current code to implement the Job class as it is asked. I'd like to have this project ready going into Project 4 as my biggest issue with this week's project was I didn't feel that my code in Project 2 was conducive to success going into Project 3.