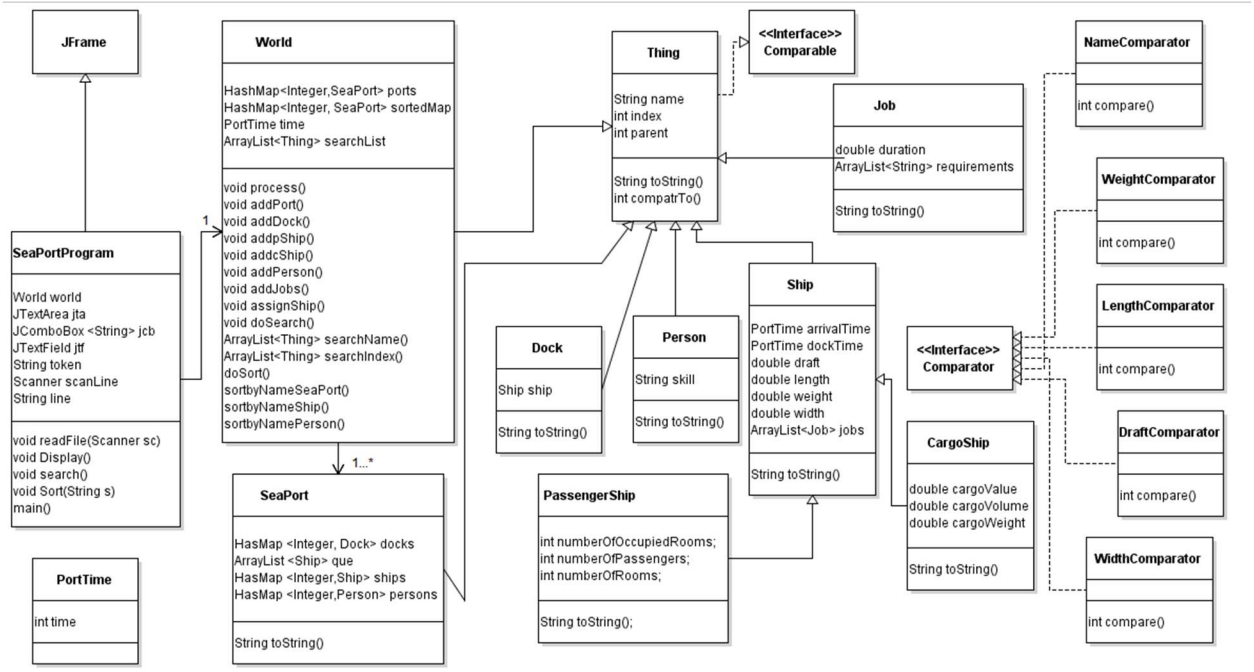
Project 2 Documentation

Name: Pooja Patel

Date: April 14th 2019

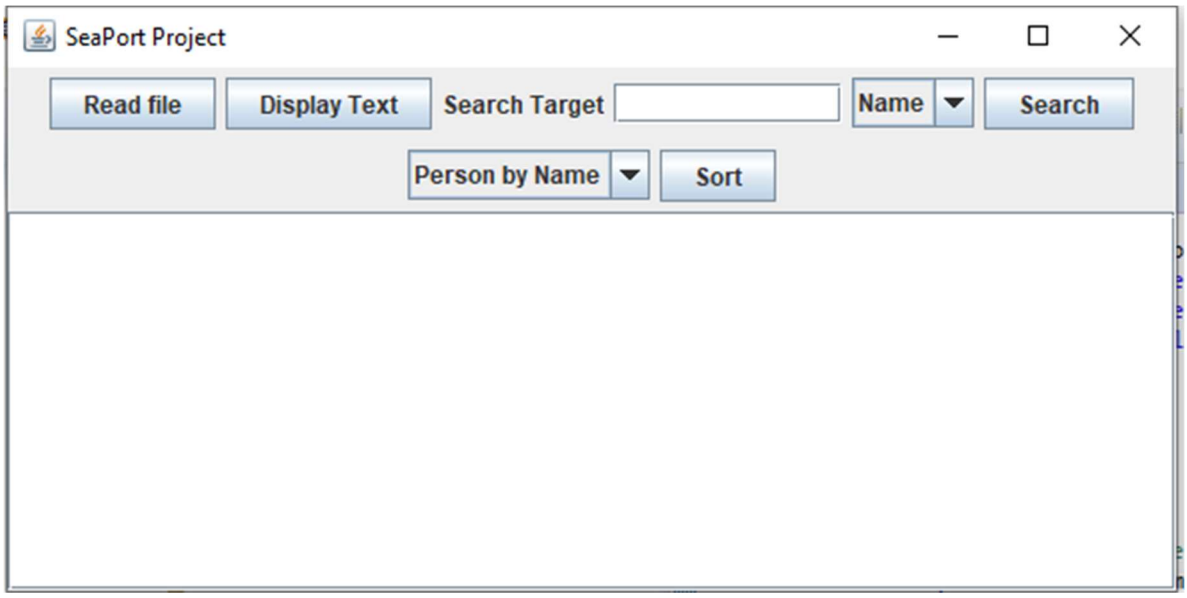
Class: CMSC 335

- Design:

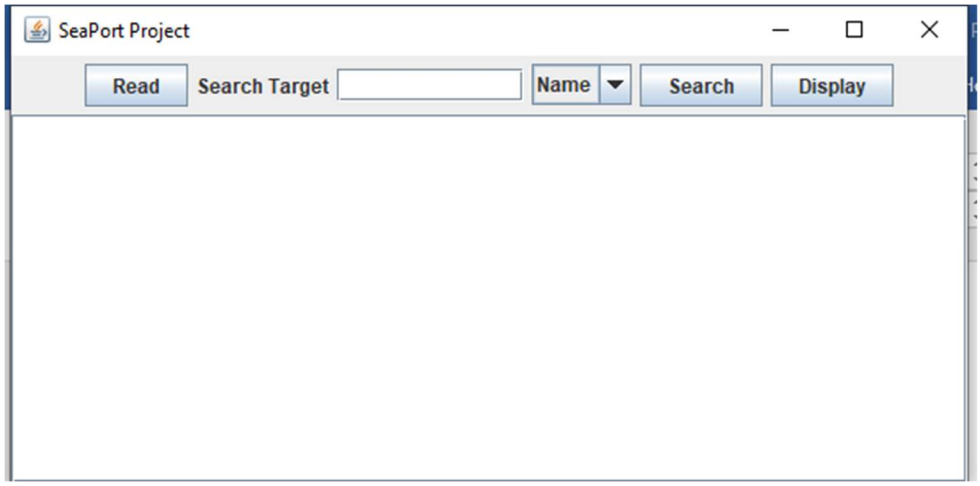


- User's guide:

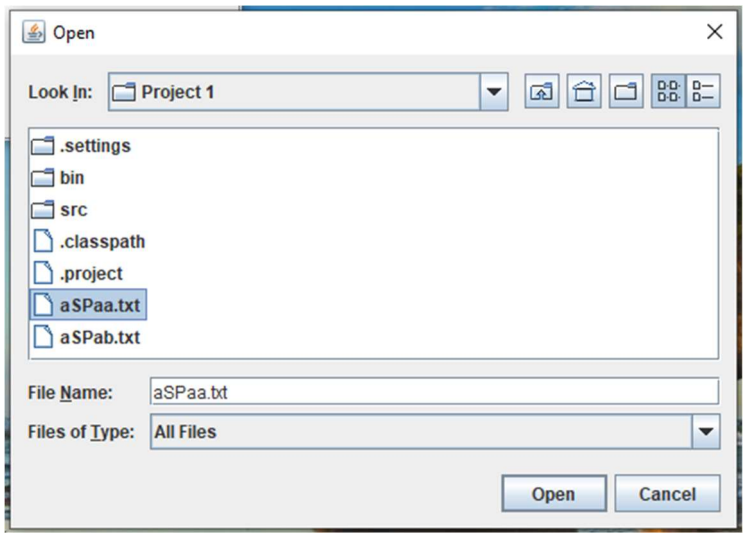
Note: How to setup and run the application is same as Project 1. The only difference in GUI is that Project 2 GUI has Sort button and Combo box for Sorting options. The Image of Project 2 GUI looks like this,



To run the Project 2 users, need to open the SeaPortProgram.java in IDE like Eclipse. After opening the SeaPortProgram.java user need to press run to running the program. Upon successful running of program, it will display the GUI screen like the image given below, (All the images below will not have sort button since the user guide would be same as Project 1 till searching functionality).



To read the data file press the Read button and it will open a new window that allows user to choose file from their computer, like this image

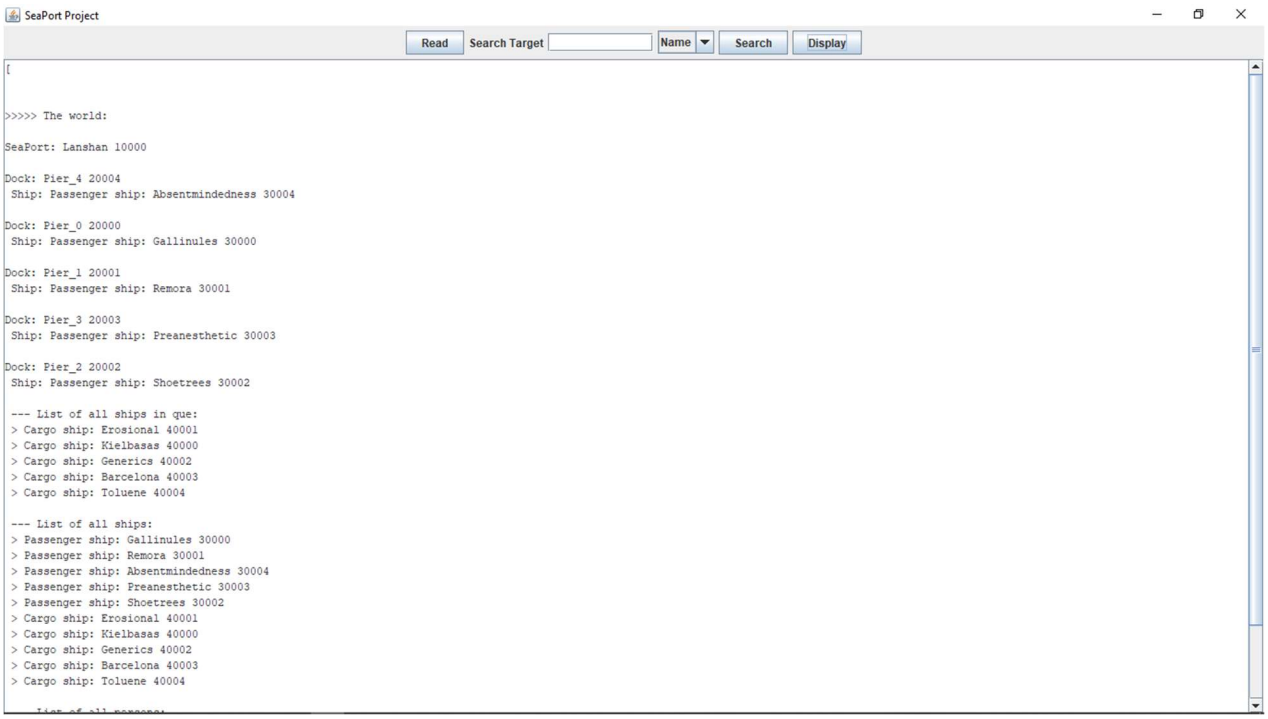


Choose the file that you want to read and press the open will read the all data from file.

Now to display the content of input file on GUI screen press the Display button. Pressing the Display would dispay all the data on GUI as shown in image below, we have chosen aSPaa.txt file here.

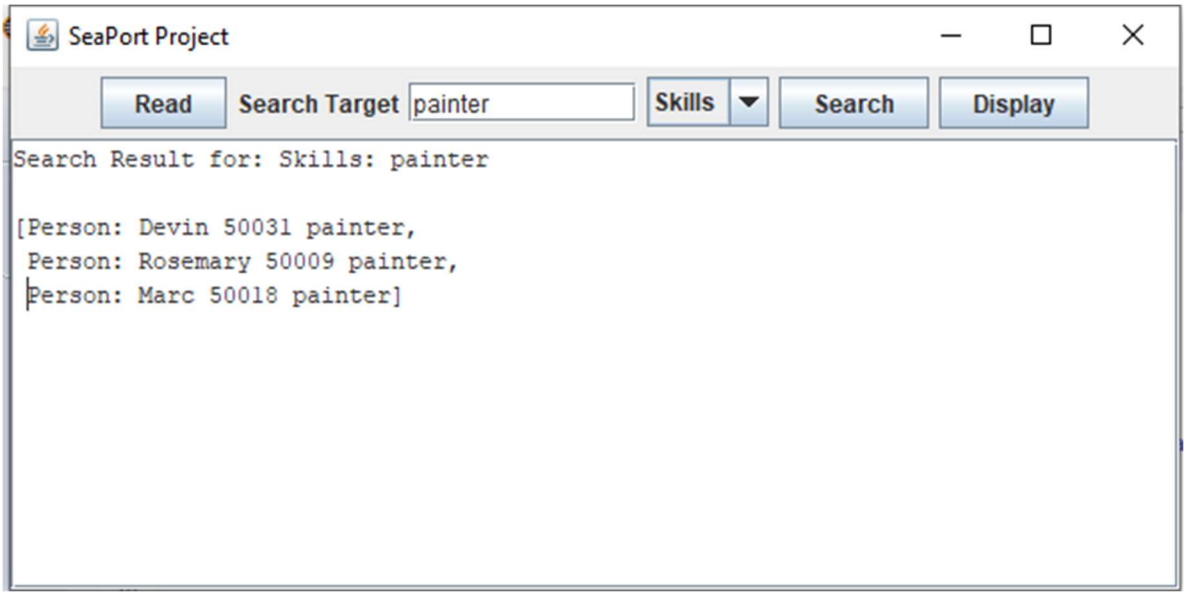
We can resize the window or scroll the scrollbar up or down to see the result.

Below is the image of maximize window.



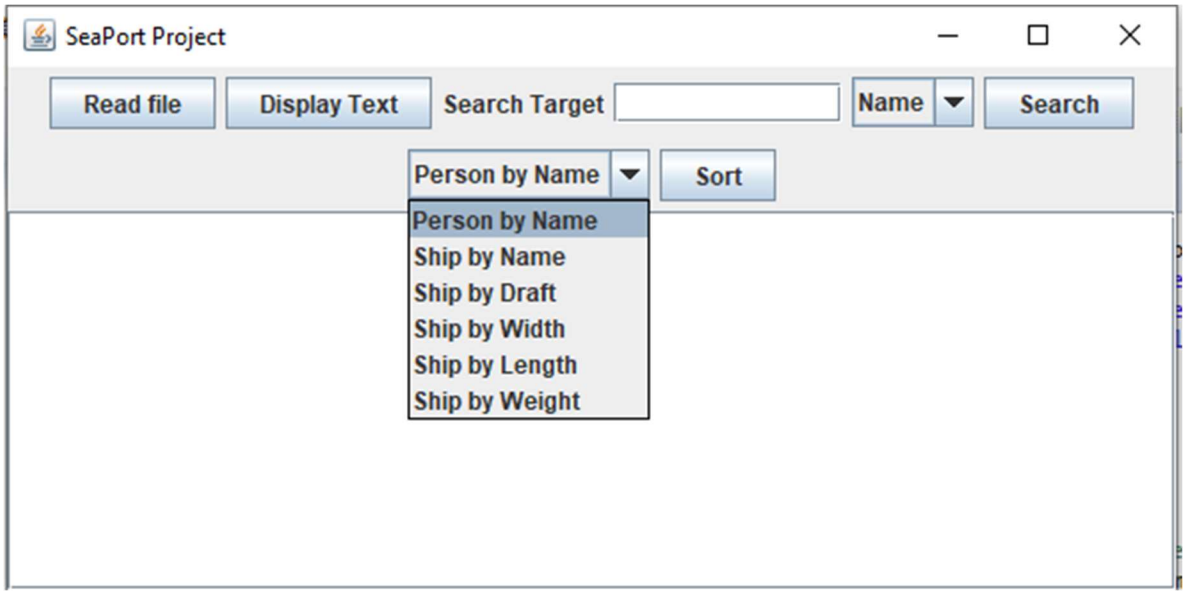
User can Search for any field by choosing a type from Combobox such as,Name, Index or Skill. And then adding a text in Search Taget textfield and press the Search button. For,example Search for a Name: Pier_1 will display the result shown below.

Another search example is Searching for a skill: painter will display all the person with painting skill as shown in image.

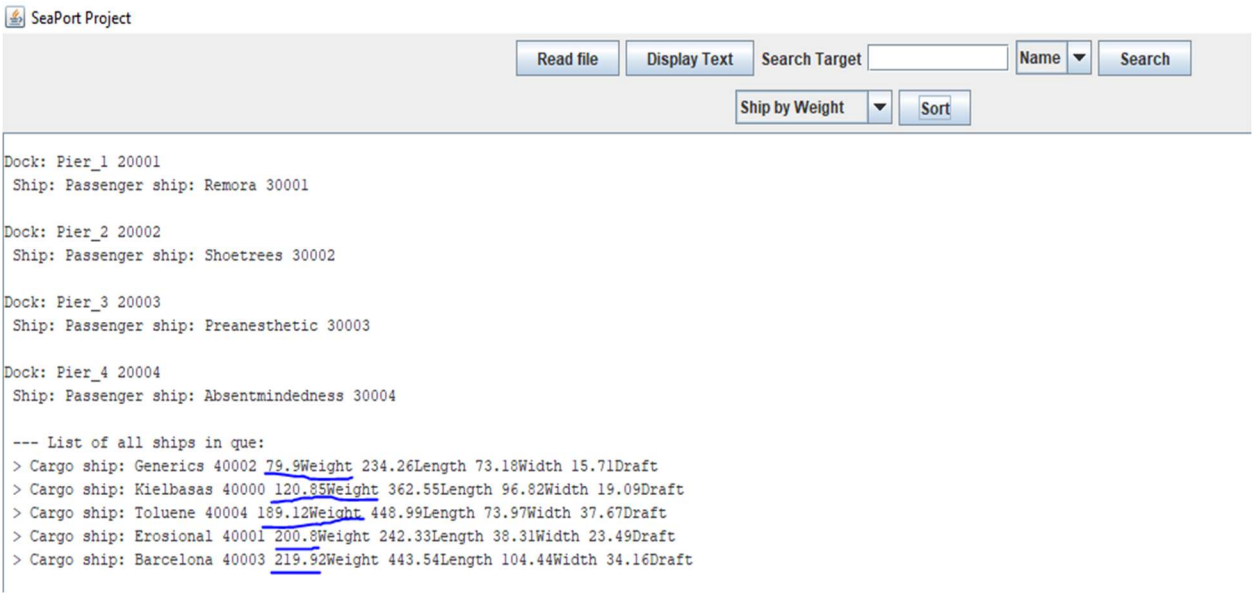


Project 2 Added functionality:

As Project 2 adds functionality of sort button user can sort for different field by name and sort the ships by different attributes in ques.



For example user can sort the Ships in que by Weight by chossing Ship by Weight from combobox and pressong sort button.

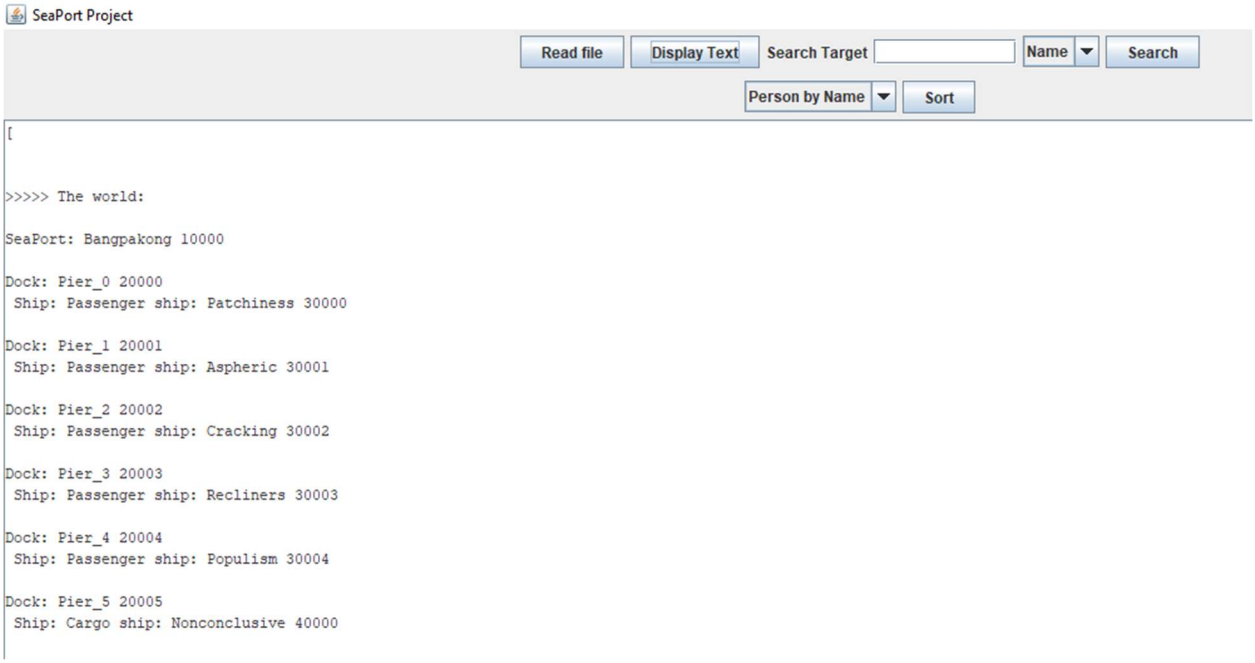


All the ships in above image is sorted by weight in que.

- Assumption:

  All the docks in in SeaPort is already sorted, since it is saved in HashMap so sorting function on Docks is not implemeted.

  Here is the image of already sorted Docks.



- Test Plan:

  I have used 2 data files to test this program aSPaa.txt and aSPab.txt.

  aSPaa.txt has only one port whereas, aSPab has more than one Seaport.

  I have modified aSPab.txt file. I removed all the line that starts with Jobs since we are not implementing Job untill project 3 or 4. I have atteched aSPab.txt file in submission that I used for testing this program.

  For testing of testcase 6 to 10 aSPab.txt is used.

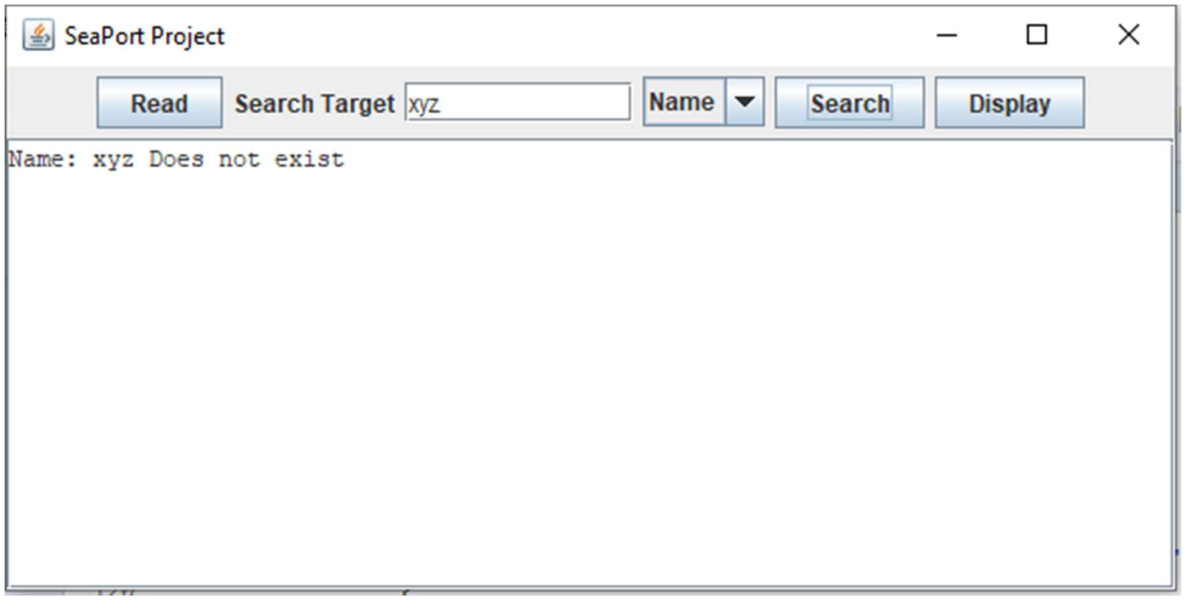
Number	Test Scenario	Input	Pass/Fail
1	Searching for a Dock by index	Index: 20004	Pass
2	Searching for Passenger Ship by Name	Name: remora	Pass
3	Searching for something that does not exist.	Name : xyz	Pass
4	Searching for a person with skill stevedore	Skills: stevedore	Pass
5	Searching for a SeaPort by name	Name: Wuchun	Pass
6	Sorting for Ships by Name	Choose Ships by Name	Pass
7	Sorting for Persons by Name	Choose Person by Name	Pass
8	Sorting for Ship by Length	Choose Ship by Length	Pass

9	Sorting for Ship by Width	Choose Ship by Width	Pass
10	Sorting for Ship by Draft	Choose Ship by Draft	Pass

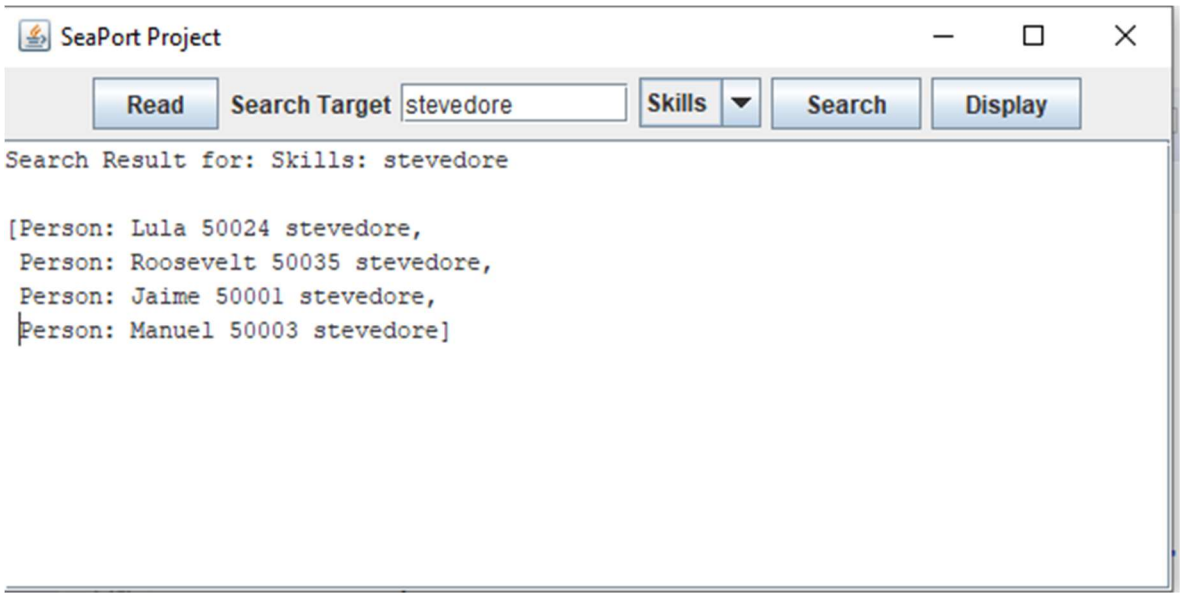
TestCase 1 Screen shot:

Test Case 2 Screenshot

TestCase 3 Screenshot:

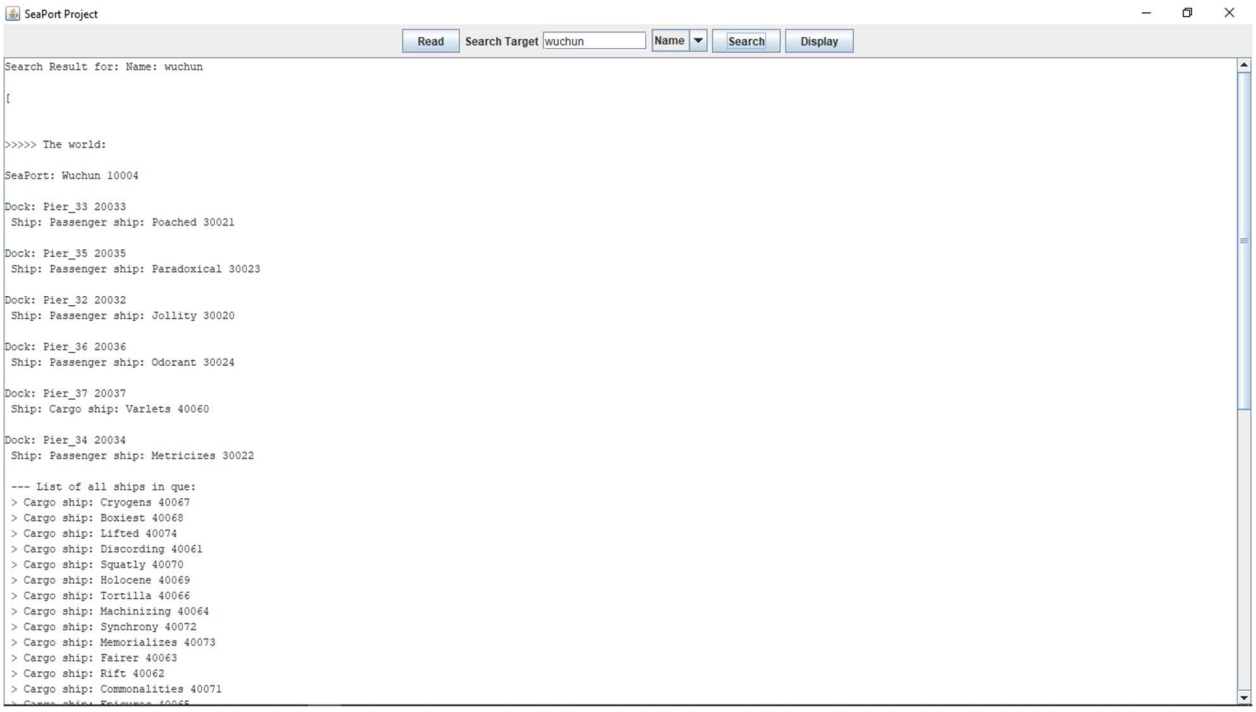


Testcase 4 Screenshot:

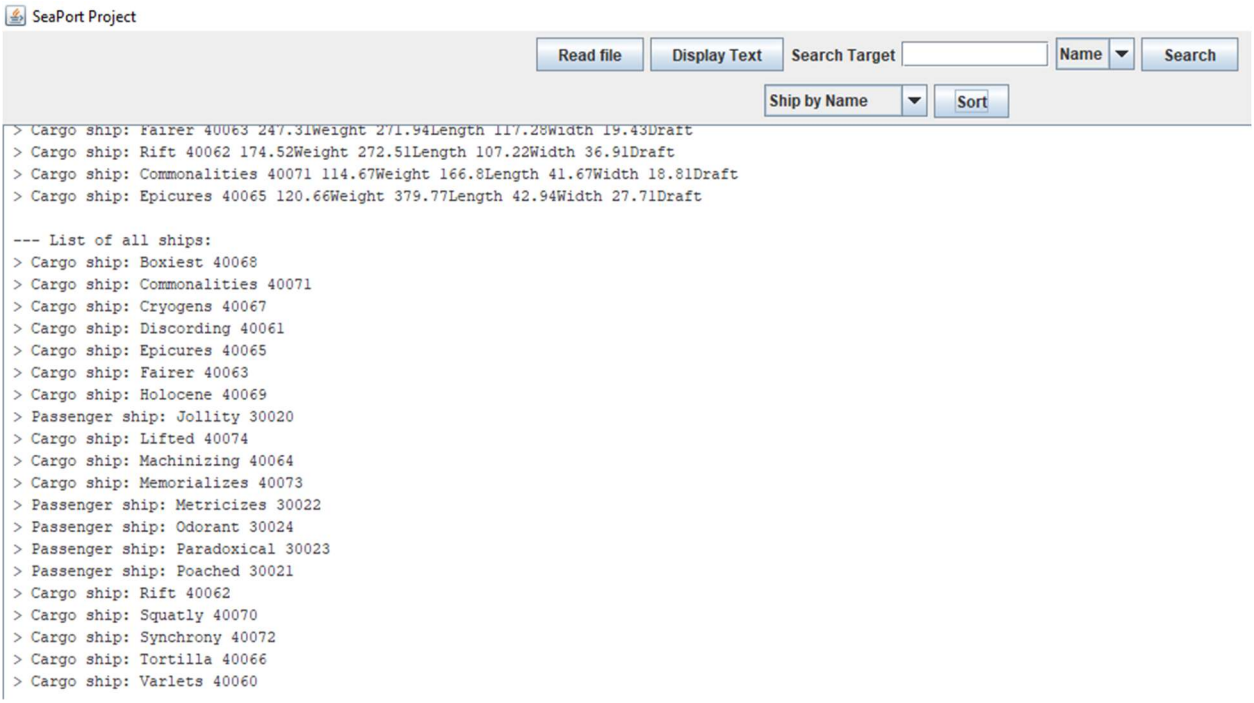


TestCase5 screen shot:

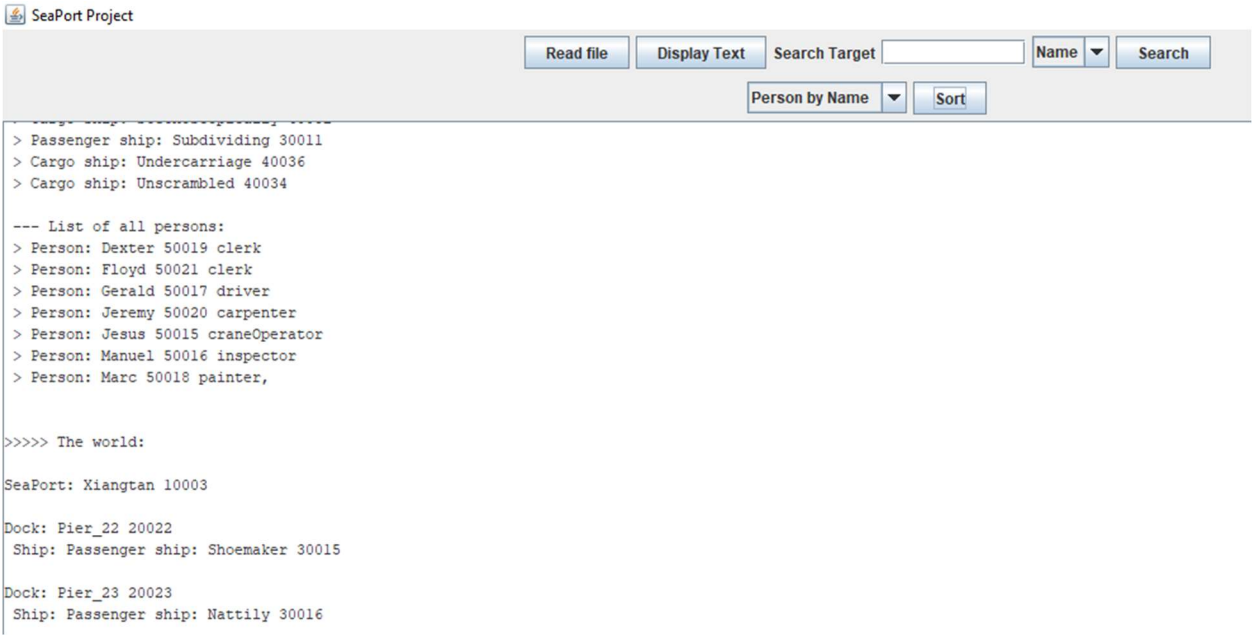
The resized window will look like this,



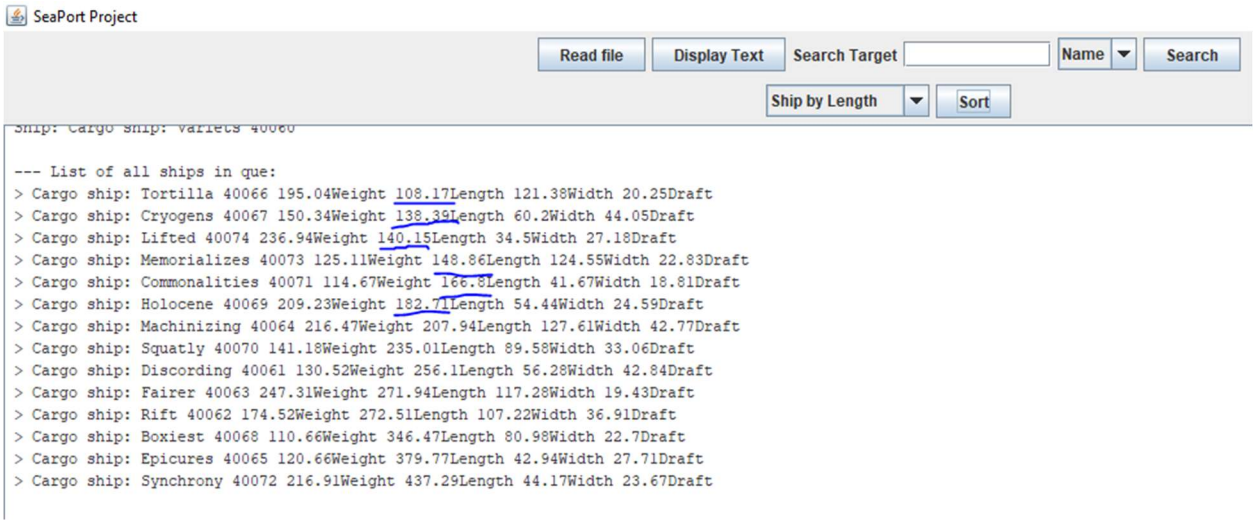
TestCase6 screen shot:



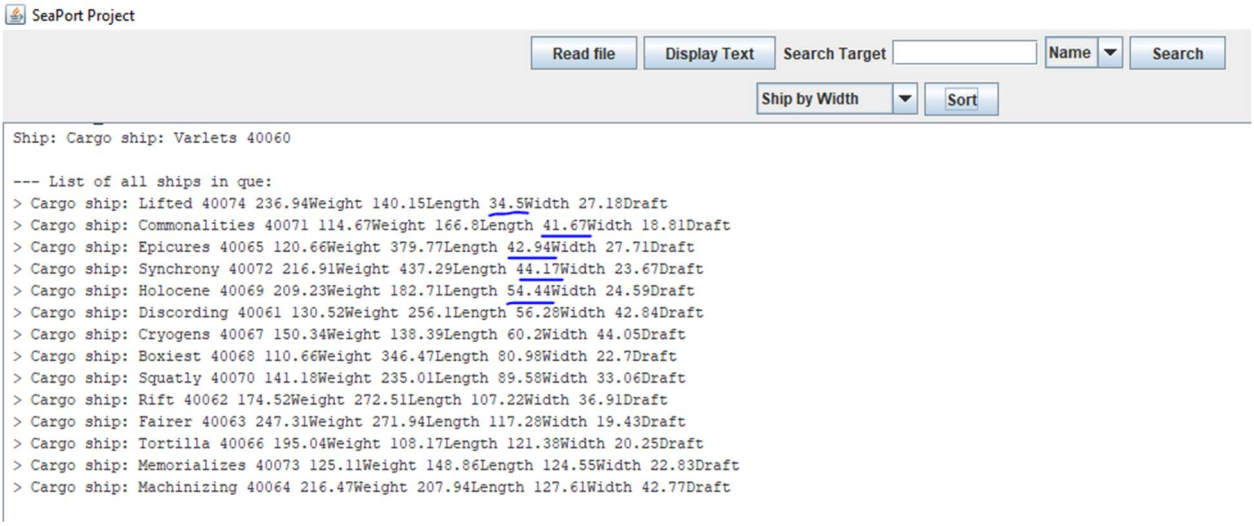
TestCase7 screen shot:



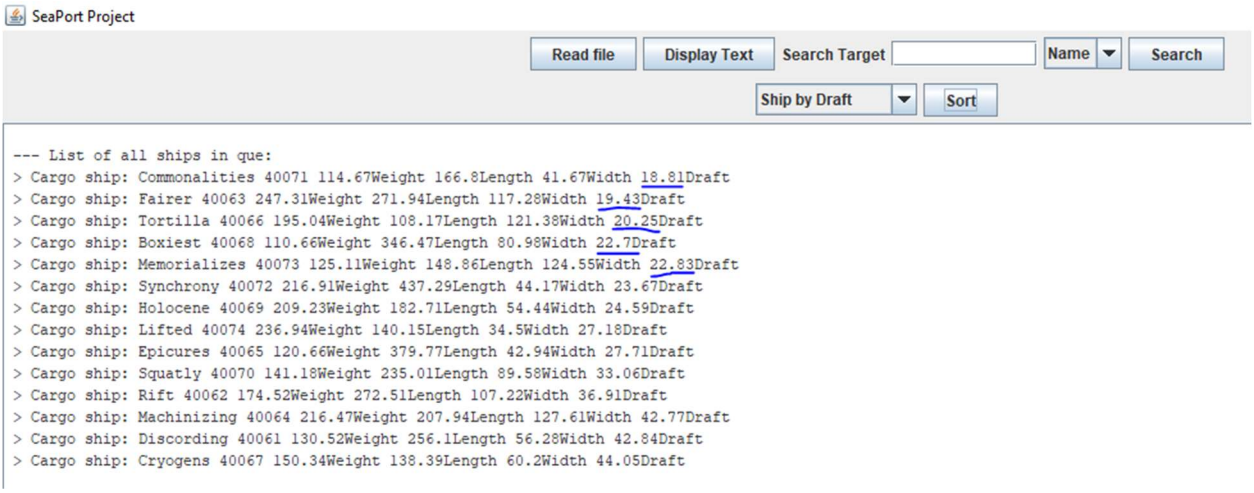
TestCase8 screen shot:



TestCase9 screen shot:



TestCase10 screen shot:



- Not Implemented:

  Not Applicable.

- Lesson Learned:

  Project 1:

  During working on this project, I realized that How I can connect the idea of object-oriented programming with the real world. I learned the use of inheritance so that we can reuse the code, and all the class is a type of a Thing (Thing.java) so that in future projects we can compare and sort them. I also learn that how to select file using JFileChooser in javaFx.

  In this project I can more improve on displaying my search result.

  Project 2:

  During working on Project 2 had a very hard time figuring out how to do sorting in HashMap. After lot of trial and error finally implemented sorting on HashMap.