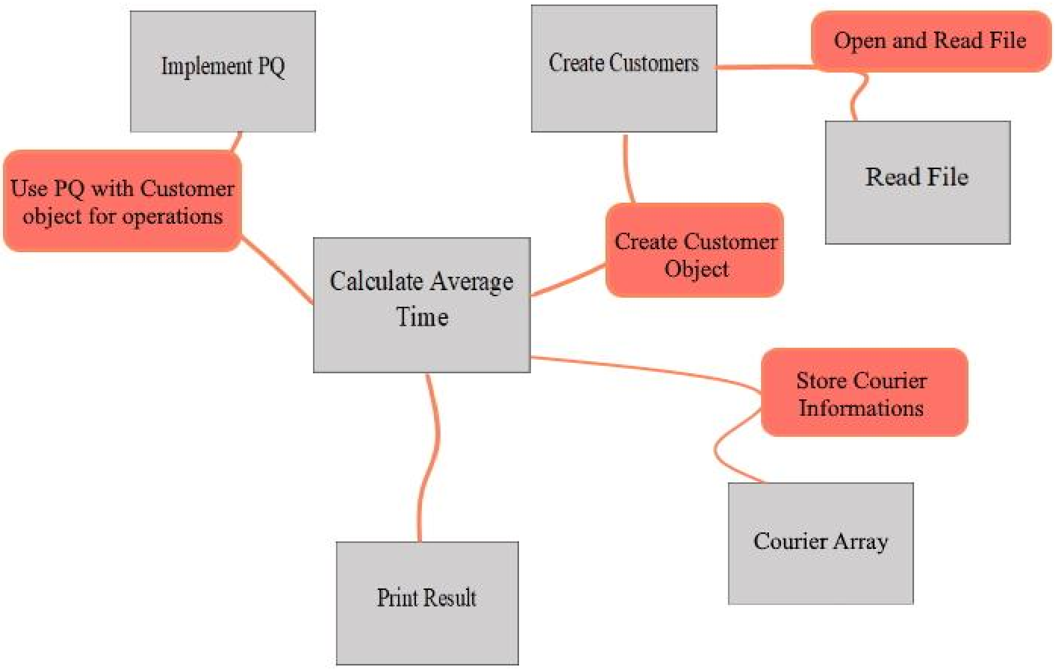
Mustafa Baran Ercan-28810555206
Bedir Esen-15203509106

CMPE_223–SEC06
Assignment-4

PA REPORT – 4

Problem Statement and Code Design

For this part of the assignment, we designed a heap-based priority queue program that stores customers who have ordered and are waiting for the courier to bring their order. In addition to that, our program calculates the minimum courier number needed to deliver the orders within the given average waiting time for customers. Also, our code includes several sub-parts to make it modular. These sub-modules are basically defined using the structure chart below.



Implementation, Functionality

Our code consists of the MaxPQ.java, Node.java, Customer.java and Main.java classes. In the MaxPQ.java and Node.java classes we have implemented a standard java primary queue structure on a heap base and the node class allowed us to store the key values of each customer in the PQ. In the MaxPQ.java class we have regular heap based PQ methods such as insert,swap, swim, sink etc. In the Customer.java class we create and carry the information of the customer objects that is given in the text file. The Main.java inserts these particular customer objects to the PQ and stores their values in the nodes. The Main method also contains arrays that keep the specific information of the Couriers such as are they available or what are their returning time etc. After some calculations, program finds the average waiting time for each specific courier time; and as long as the criterias are met, the program keeps running.

The program comprises six primary sub-modules, as demonstrated in the previous section. Details about each sub-module are defined in the next section:

1. **Read File:** This sub-module opens and reads the sample input file. The information on the file is used to create customer objects.
2. **Create Customers:** This sub-module instantiates customer objects. Each customer object has the values of orderTime, travelTime, year, and ID that is read from the input file.
3. **Implement PQ:** This sub-module uses its properties to sort the inserted customer objects according to the time they have been with the company. It also stores customers' other fields using nodes in order to reach them later, such as ID, travelTime.
4. **Calculate Average Time:** This sub-module calculates the waiting time for each customer, then calculates the average. This is needed in order to find the minimum number of couriers needed for the delivery.
5. **Courier Array:** This sub-module is the part where the program keeps track of the couriers. Courier count starts with 1 when the program begins, then increases in each step. The information about couriers such as is the courier available or when the courier will return and etc. are followed with these arrays in the Main class.
6. **Print the Result:** This sub-module is the part where the program prints out the desired outputs on the screen.

Testing

We have done multiple testing we have made, we faced tons of errors such as null pointer, out of bounds etc. We have rearranged our code to overcome these problems multiple times. Despite thinking about the correct algorithm, we could not increase the courier number in our loop, so we could not pass the VPL tests in the end.

Even though we have made a lot of brainstorming about our code, we couldn't handle our program to work with multiple couriers correctly. Therefore, despite the majority of our code is correct in the thought, we were not able to finalize our program to pass the VPL test.

Final Assessments

While creating our program, we were able to figure out a lot of things such as implementing a heap based primary queue, creating and inserting customers with given parameters, keeping track of couriers using arrays, and finding the average time. We also had a hard time while trying to return the couriers back from customers. It was nice about the assignment that we had to learn about how to use PQ structure to use it with user defined objects.