**Resource Reservation System Documentation**

**Network Creation**

The NetworkNode java class contains the main method that processes the arguments passed during program execution. The resource allocation string specified among the arguments is extracted and processed to be strored in a resources HashMap. A new object of the Network class which extends the java Threads class is instantiated with the tcp port number and resources HashMap passed as arguments to its constructor then the objects start method is called.

The Network class overrides java Thread’s run method and in it, a ServerSocket object is initialized along with the tcp port value previously passed to the Network class constructor. Connections are listened for infinitely to initialize a Client Socket. The initialized Socket object and the Network object is passed as constructor arguments to the NetworkThread class which also extends java’s Thread class, and the start method is called.

The NetworkThread class handles all the Client socket request logic using r the resources HashMap from the Network Class. The communication protocol is further described below.

**Communication Protocol**

-NetworkClient  
After client server connection, the communication begins with the NetworkClient sending a command to the NetworkNode in the following string format: “Sending: 321676 A:3 C:2 B:1”. The NetworkNode processes the command by splitting the string command into tokens and extracting the resource request from the command tokens.

-NetworkNode

After the NetworkNode receives collects the resource request, it compares the quantity of the resources with the Network’s resources stored in an HashMap.

In the event of resource availability the NetworkNode responds as such: outputStream.write(("ALLOCATED: " + requestedResourceKey + ":" + requestedResourceValue + "\n").getBytes());”

The NetworkNode’s resources are computed accordingly, and its resources HashMap updated.

-NetworkClient

Now the NetworkClient receives the “ALLOCATED: ” message response and prints out the response from NetworkNode along with a message of the remaining resources left at the node.

**How to Compile/Execute**

The compiled .class files have been provide in the project folder. The program only needs to be executed using the java command and providing the necessary arguments. NetworkNode.java is to be executed first as such: java NetworkNode.java -ident 123 -tcpport 9991 -gateway localhost:9990 A:5 B:4 C:3

NetworkClient.java is to be executed next, as such: java NetworkClient -ident 321 -gateway localhost:9991 A:3 C:2 B:1

The two programs assume validity and correct sequence of the arguments provided during execution.

**What Was Implemented**

* The resource allocation logic on the Network node was implemented to model a valid response in the event of resource availability and resource insufficiency.
* The network and client identifiers were implemented to be unique by append a random three digit number to the identifier argument value provided during program execution
* Program state logs were implemented to indicate how much resources are left after an allocation has taken place.

**What does not work**

The TERMINATE command doesn’t work for both the NetworkeNode and NetworkClient.