Project Part 2 (50%)

You company has asked you to make your twitter service available as a microservice with an appropriate API (you were advised to redesign the interface to use either REST or RPC (ONC RPC /Thrift /gRPC).

1. Describe a redesign of your twitter service using RPC or REST – explain why you choose the interface method you chose (REST/RPC) and using pseudo-code describe the implementation of the four methods of your service: **10 marks**
2. Implement one of the four methods: **15 marks** (Note you do not need to host it on Docker). Look at the labs on RPC (ONC RPC/Thrift/gRPC) or REST(Netbeans/DropWizard) to help you.
3. Compare and contrast your original protocol-based solution with the new REST/RPC solution **25 marks**

Save your code in a zip file with your T number (or provide a github address) and save your report with your Tnumber. You don’t have to follow any report structure this time, just answer the 3 questions above.

Upload the code to the X drive (Project part 2) and your report to Turnitin by: May 11th

Tips:

YOU DO NOT NEED TO BUILD YOUR OWN PROTOCOL AGAIN THIS TIME AND YOU DON’T NEED TO IMPLEMENT PROTOCOL RESPONSE MESSAGES EITHER. YOU ARE A LAYER ABOVE THE PROTOCOL NOW.

INSTEAD ALL YOU NEED TO DO IS EXPOSE A METHOD LIKE LOGIN USING EITHER RPC OR REST

If you go the RPC route, you really only need to redo the ONC RPC lab (which took 2 hours) and redesign it for instance to send a username + password to a function called login

If you go the REST route, you just need to expose a resource in a URI like this: /login?username\_password

Then send a GET message to this URI

You can use any IDE you wish if using REST (we did a tutorial on Netbeans), Eclipse/intelliJ/Visual Studio – just locate a good tutorial on building REST applications with the chosen IDE

Please contact me if you have any questions.