**Computer Engineering 4DN4**

**Lab #2**

**Online Grade Retrieval Network Application**

**March 8th , 2018**

**Kai Ling Yan 001325791**

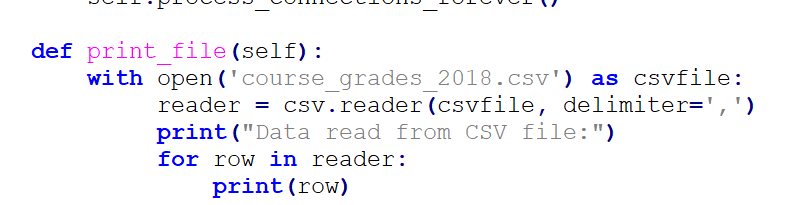
**Xinrun Shan 001323005**

**Experiment**

The changes are made based on the file EchoClientServer.py with blocking.

**Server**

1. Start the server in shell window. The server prints out the data read from the CSV file.



Using csv module to return a reader object which will read and iterate over lines for the given csvfile.

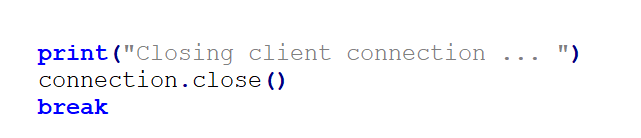
1. The server decodes the receive bytes and check the receive string is ‘GAC’ or identification.

A screenshot of a social media post

Description generated with very high confidence

If the received message is GAC, then take the row with the first object = “Averages” and transfer the elements from dictionary form into a single string before it is sent. On the other hand, if the received message is an identification, it transferred the username and password on the table into hash object and compares with the message that is received. If it is matched one of the identification, then a flag is set, and the corresponding row is copied. In addition, if flag=1, transfer the copied row into string and send to the client. Otherwise, send a error message to the client to indicate that the username and password that is received is incorrect.

1. The connection to the particular client should be closed after the require information is sent to the client.



Closing the connection after the above step is finished.

**Client**

1. Start the client in a different shell. It prompts the user for a command which is echoed by the client shown on the terminal.

A screenshot of a social media post

Description generated with very high confidence

First, an input is prompted to the user. If the command entered is GAC which requests the average grade, then client output “Fetching grade averages” and is followed by the grade that returned by the server. If the command entered is not GAC, the user is asked to enter the student ID and password. Besides, getpass is used to hide the input message for security. The client is always printed out what is entered on the terminal. After entering the command, the client is established connection with the server.

1. Send the input information to the server.

A screenshot of a social media post

Description generated with very high confidence

If the user is entering ID/password, the information should be transferred into hash object before it is sent to the server. Using m.hexidigest() to should the hash value of the entered ID/password. If the input is GAC, send the message directly as normal.

1. Since the connection should be established each time after the user inputs the command, the loop is shown below.

Get\_command()🡪get\_socket()🡪connect\_to\_server()🡪send\_console\_input\_forever()🡪connection\_send()🡪connection\_receive()…. Then it loops back to get\_command().