Sockets Assignment 3 Report

## **Issues and Modifications:**

When Implementing Ken’s Server with Emilio’s Client we ran into the issue where the client would add trailing blank spaces to the message so when a user would type “no” the server would receive “no ” which would cause the logic to fail. To remedy this after the server would receive and decode the data strip was called to remove any trailing or leading blank spaces.

## **Code:**

### Client( unchanged):

#!/usr/bin/env python3

from socket import \*

import sys

MLENGTH = 125

class MySocket:

def \_\_init\_\_(self, sock=None):

if sock is None:

self.sock = socket(AF\_INET, SOCK\_STREAM)

else:

self.sock = sock

def connect(self, host, port):

self.sock.connect((host, port))

def mysend(self, msg):

self.sock.send(msg.encode('utf-8'))

def myreceive(self):

chunk = self.sock.recv(MLENGTH)

chunk = chunk.strip()

return chunk

def myprepare(self,msg): # ensures that the message is of the right length

while len(msg) < 100:

msg = msg + ' '

return msg

class ConsoleReader:

def \_\_init\_\_(self):

pass

def get\_next\_line(self):

return input()

class FileReader:

def \_\_init\_\_(self, filename):

with open(filename) as f:

content = f.readlines()

self.file\_lines = [x.strip() for x in content]

def get\_next\_line(self):

return self.file\_lines.pop(0)

def main():

try:

start\_socket()

except FileNotFoundError:

print("File was not found")

def start\_socket():

s = MySocket()

s.connect("localhost",7069)

reader = get\_reader()

conversate(s, reader)

def get\_reader():

if len(sys.argv) == 1:

return ConsoleReader()

else:

return FileReader(sys.argv[1])

def conversate(s, reader):

response = s.myreceive()

print(response.decode())

while True:

currentline = reader.get\_next\_line()

currentline = s.myprepare(currentline)

s.mysend(currentline)

response = s.myreceive()

print(response.decode())

main()

### Server:

#!/usr/bin/env python3

from socket import \*

s = socket(AF\_INET, SOCK\_STREAM)

s.bind(("127.0.0.1", 7069))

s.listen(5)

c,a = s.accept()

counter = 0

data =""

while True:

if counter == 0:

c.send("System:\tHello, welcome to chatbot program. \nAre you importing messages from files? \n Enter only either \"yes\" or \"no\".".encode())

counter += 1

elif counter == 1:

if data == "yes":

c.send("System:\tImport file but currently not available.\n System:\tHello, are you male or female?".encode())

counter += 1

elif data == "no":

c.send("System:\tHello, are you male or female?".encode())

counter += 1

else:

c.send("System:\tPlease enter in the correct format ... \n".encode())

elif counter == 2:

if data == "female":

c.send("System:\tHow excellent! Are you a CS major?".encode())

elif data == "male":

c.send("System:\tMe too. Are you CS major?".encode())

else:

c.send("System:\tGreat! Anyways, are you CS major?".encode())

counter += 1

elif counter == 3:

if data == "no":

c.send("System:\tToo bad. Anyway, what's an animal you like, and two you don't?".encode())

elif data == "yes":

c.send("System:\tExcellent, I am too. What's an animal you don't like, and two you don't?".encode())

else:

c.send("System:\tCool! By the way, what's an animal you like, and two you don't?".encode())

counter += 1

elif counter == 4:

data1 = data.split(',')

msg = "System:\t%s awesome, but i hate %s too. Bye for now." % (data1[0].strip(), data1[-1].strip())

c.send(msg.encode())

counter += 1

else:

c.send(''.encode())

data = c.recv(1000).decode()

data = data.strip()

c.close()

## Output:

