

Explanation:

TCPServer.py:




We are first importing a socket library, to handle the TCP/IP communication. We set the HOST to 127.0.0.1 because it will connect to your own machine. Then we set a PORT number where the server listens on. We define a default message which is an apple a day keeps the doctor away. We also have a default password which is 123!abc. We then create a variable to set the default message to it, the reason is the message can be updated with the MSGSTORE command. We then create a TCP socket and bind it to the IP and port and start listening for up to 5 clients. So while the server is accepting new clients, we are going to accept the connection. We are then setting waiting\_for\_msgstore to false because it is not expecting a message to store just waiting for regular commands and if the user sends in a command, then the server will send an error message. We are then going to read in a command sent from the client and decode the bytes into a string and strip the whitespace. If the command is empty then we skip all the code after that in the while loop and goes to the next iteration. Then the server updates the message and sends an OK. If the command is to get the current message we send a 200 OK and the current message. If the command is to store a new message, we send a 200 OK and set the waiting\_for\_msgstore as true. If the command is quit, we send 200 OK and exit the while loop. If the command is shutdown then the server sends a 300 password required and gets the password from the client. If the password is right, the server sends a 200 OKAY message. And the server itself prints out the accepted password. Then we close the connection with that one client. And then we shut down the server completely. We then exit the program. If the password wasn't right then the server sends a 301 error message. If the user didn't list the known commands, the server sends a 400 error message.



TCPClient.py

We are getting the IP from the command line and using the same port as the server. We are connecting via IP and port. We are getting the user input for which command to run and sending the command to the server. If the command is for getting the current message, we then receive the server message and if the server message is 200 OK, then we receive the current message and print it out. If the command is storing a new message, we get the server message and if the server message is 200 OK, we then ask the user to enter in a new message and send that message to the server. We then receive 200 OK from the server and print it out. If the command is quit, we receive the server message which is 200 OK. If the command is shutdown, we receive a 300 message and print it out. We print the password and send it to the server. The server will let us know if the password is correct or not. We print the 200 OKAY message if it was correct and shut down the server. Else we received a 301 error message and are printing that out.

Instructions: Open the file in Pycharm. Click on the terminal and run the command `python TCPServer.py`, then open a new terminal and run the command `python TCPClient.py 127.0.0.1`.

Test Cases:

Test Case	Command	Input	Expected Output	Actual Output	Passed
1)	MSGGET	MSGGET	<p>Enter command (MSGGET, MSGSTORE, QUIT, SHUTDOWN): MSGGET</p> <p>Client: MSGGET</p> <p>Server: 200 OK</p> <p>An apple a day keeps the doctor away</p>	<p>Enter command (MSGGET, MSGSTORE, QUIT, SHUTDOWN): MSGGET</p> <p>Client: MSGGET</p> <p>Server: 200 OK</p> <p>An apple a day keeps the doctor away</p>	
2)	MSGSTORE	MSGSTORE with new message	<p>Enter command (MSGGET, MSGSTORE, QUIT, SHUTDOWN): MSGSTORE</p> <p>Client: MSGSTORE</p> <p>Server: 200 OK</p> <p>Enter new message of the day:</p> <p>Client: Imagination is more important than knowledge</p> <p>Server: 200 OK</p>	<p>Enter command (MSGGET, MSGSTORE, QUIT, SHUTDOWN): MSGSTORE</p> <p>Client: MSGSTORE</p> <p>Server: 200 OK</p> <p>Enter new message of the day:</p> <p>Client: Imagination is more important than knowledge</p> <p>Server: 200 OK</p>	
3)	QUIT	QUIT	<p>Enter command (MSGGET, MSGSTORE, QUIT, SHUTDOWN): QUIT</p>	<p>Enter command (MSGGET, MSGSTORE, QUIT, SHUTDOWN): QUIT</p>	

			Client: QUIT Server: 200 OK	Client: QUIT Server: 200 OK	
4)	SHUTDOWN w/ right password	SHUTDOWN w/ right password	Enter command (MSGGET, MSGSTORE, QUIT, SHUTDOWN): SHUTDOWN  Client: SHUTDOWN  Server: 300 PASSWORD REQUIRED  Enter shutdown password: 123!abc  Client: 123!abc  Server: 200 OKAY	Enter command (MSGGET, MSGSTORE, QUIT, SHUTDOWN): SHUTDOWN  Client: SHUTDOWN  Server: 300 PASSWORD REQUIRED  Enter shutdown password: 123!abc  Client: 123!abc  Server: 200 OKAY	
5)	SHUTDOWN w/ wrong password	SHUTDOWN w/ wrong password	Enter command (MSGGET, MSGSTORE, QUIT, SHUTDOWN): SHUTDOWN  Client: SHUTDOWN  Server: 300 PASSWORD REQUIRED  Enter shutdown password: 321!qwe  Client: 321!qwe  Server: 301 WRONG	Enter command (MSGGET, MSGSTORE, QUIT, SHUTDOWN): SHUTDOWN  Client: SHUTDOWN  Server: 300 PASSWORD REQUIRED  Enter shutdown password: 321!qwe  Client: 321!qwe  Server: 301 WRONG	

			PASSWORD	PASSWORD	
6)	MSGSTORE then, (QUIT or SHUTDOWN)	MSGSTORE then QUIT	Enter command (MSGGET, MSGSTORE, QUIT, SHUTDOWN): MSGSTORE  Client: MSGSTORE  Server: 200 OK  Enter new message of the day: QUIT  Client: QUIT  Server: EXPECTED MOTD, RECEIVED COMMAND	Enter command (MSGGET, MSGSTORE, QUIT, SHUTDOWN): MSGSTORE  Client: MSGSTORE  Server: 200 OK  Enter new message of the day: QUIT  Client: QUIT  Server: EXPECTED MOTD, RECEIVED COMMAND	✓

Screenshots:

1)

Client:

```
Enter command (MSGGET, MSGSTORE, QUIT, SHUTDOWN): MSGGET
Client: MSGGET
Server: 200 OK
An apple a day keeps the doctor away.
```

Server:

```
TCP Server is listening on 127.0.0.1:26310
Connected by ('127.0.0.1', 58365)
Received:from ('127.0.0.1', 58365): MSGGET
```

2)

Client:

```
Enter command (MSGGET, MSGSTORE, QUIT, SHUTDOWN): MSGSTORE
Client: MSGSTORE
Server: 200 OK
Enter new message of the day: Imagination is more important than knowledge
Client: Imagination is more important than knowledge
Server: 200 OK
```

Server:

```
TCP Server is listening on 127.0.0.1:26310
Connected by ('127.0.0.1', 58400)
Received:from ('127.0.0.1', 58400): MSGSTORE
Received:from ('127.0.0.1', 58400): Imagination is more important than knowledge
Server: MOTD updated to: Imagination is more important than knowledge
```

3)

Client:

```
Enter command (MSGGET, MSGSTORE, QUIT, SHUTDOWN): QUIT
Client: QUIT
Server: 200 OK
```

Server:

```
TCP Server is listening on 127.0.0.1:26310
Connected by ('127.0.0.1', 58474)
Received:from ('127.0.0.1', 58474): QUIT
[-] Connection closed with ('127.0.0.1', 58474)
```

4)

Client:

```
Enter command (MSGGET, MSGSTORE, QUIT, SHUTDOWN): SHUTDOWN
Client: SHUTDOWN
Server: 300 PASSWORD REQUIRED
Enter shutdown password: 123!abc
Client: 123!abc
Server: 200 OKAY
```

Server:

```
TCP Server is listening on 127.0.0.1:26310
Connected by ('127.0.0.1', 59044)
Received:from ('127.0.0.1', 59044): SHUTDOWN
Server Password accepted. Shutting down
[-] Connection closed with ('127.0.0.1', 59044)
```

5)

Client:

```
Enter command (MSGGET, MSGSTORE, QUIT, SHUTDOWN): SHUTDOWN
Client: SHUTDOWN
Server: 300 PASSWORD REQUIRED
Enter shutdown password: 321!qwe
Client: 321!qwe
Server: 301 WRONG PASSWORD
```

Server:

```
TCP Server is listening on 127.0.0.1:26310
Connected by ('127.0.0.1', 58560)
Received:from ('127.0.0.1', 58560): SHUTDOWN
```

6)

Client:

```
Enter command (MSGGET, MSGSTORE, QUIT, SHUTDOWN): MSGSTORE
Client: MSGSTORE
Server: 200 OK
Enter new message of the day: QUIT
Client: QUIT
Server: EXPECTED MOTD, RECEIVED COMMAND
```

Server:

```
TCP Server is listening on 127.0.0.1:26310
Connected by ('127.0.0.1', 59142)
Received:from ('127.0.0.1', 59142): MSGSTORE
Received:from ('127.0.0.1', 59142): QUIT
```

If you send MSGSTORE and then QUIT, it shouldn't read QUIT as a command, instead the program should say that is not applicable in this scenario. When you choose to store a message, it shouldn't be one of the commands because that will confuse the program.