COMPUTER NETWORK

Lab 5

Zia Ur Rehman

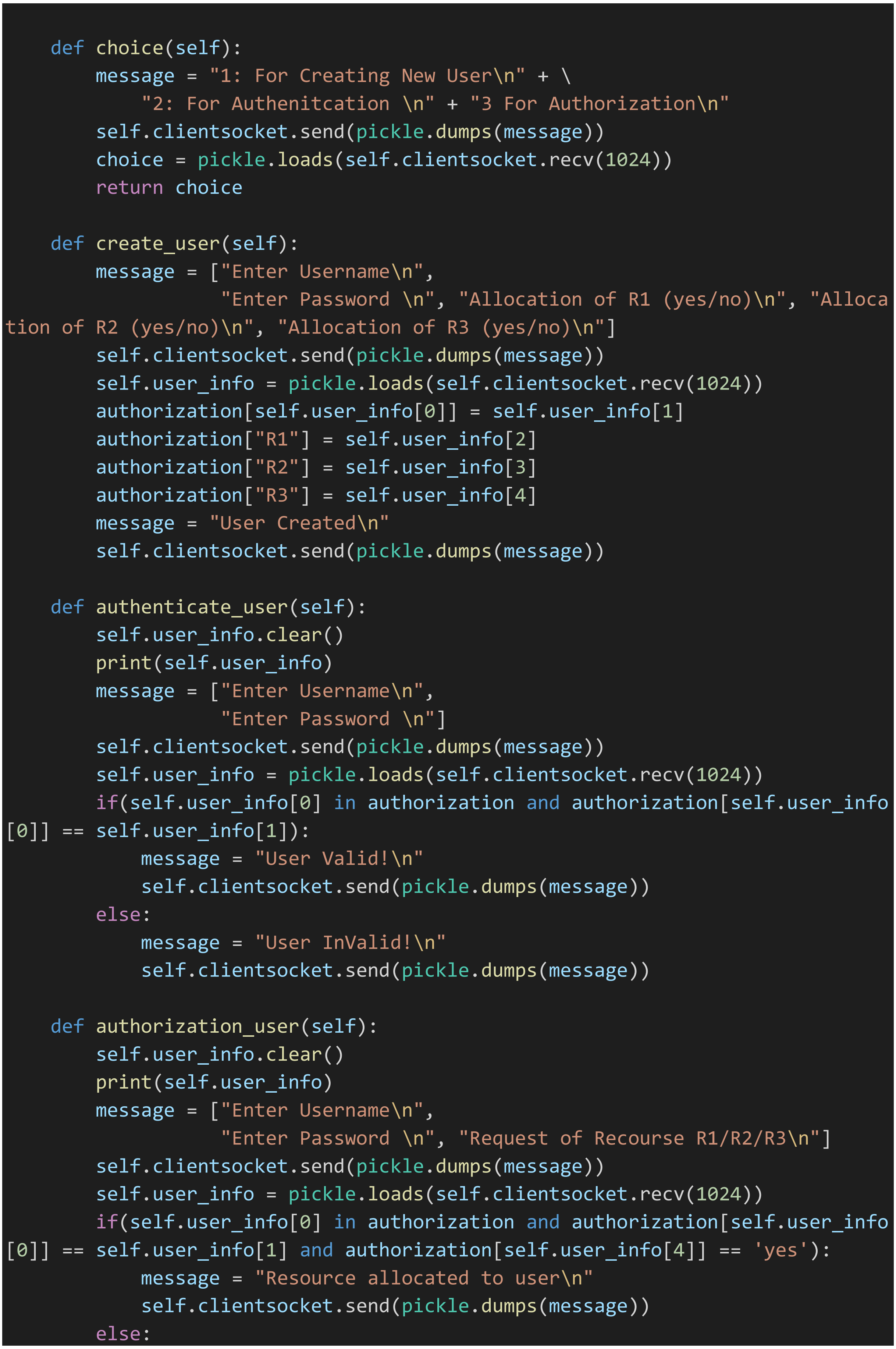
1802034

Server:

```
import socket
from common_thread import commonThread


address = "192.168.100.50"

port = 5050
def main():


    s = socket.socket()
    s.bind((address,port))

    s.listen(5)
    print("Listing for clients ...")
    while True:
        c , addr = s.accept()
        print("Client Connected :  "  , addr)
        clientThread = commonThread(c)
        clientThread.start()

if __name__ == "__main__":
    main()
```

Common Thread:

```
from threading import Thread
import pickle
authorization = {}


class commonThread(Thread):
    def __init__(self, clientsocket):
        Thread.__init__(self)
        self.clientsocket = clientsocket
        self.user_info = []

    def run(self):
        print("Client Connected")
        while(True):
            choice = self.choice()
            if(choice == '1'):
                self.create_user()
            elif(choice == '2'):
                self.authenticate_user()
            elif(choice == '3'):
                self.authorization_user()
```



```
    def choice(self):
        message = "1: For Creating New User\n" + \
            "2: For Authenitcation \n" + "3 For Authorization\n"
        self.clientsocket.send(pickle.dumps(message))
        choice = pickle.loads(self.clientsocket.recv(1024))
        return choice

    def create_user(self):
        message = ["Enter Username\n",
                   "Enter Password \n", "Allocation of R1 (yes/no)\n", "Allocation of R2 (yes/no)\n", "Allocation of R3 (yes/no)\n"]
        self.clientsocket.send(pickle.dumps(message))
        self.user_info = pickle.loads(self.clientsocket.recv(1024))
        authorization[self.user_info[0]] = self.user_info[1]
        authorization["R1"] = self.user_info[2]
        authorization["R2"] = self.user_info[3]
        authorization["R3"] = self.user_info[4]
        message = "User Created\n"
        self.clientsocket.send(pickle.dumps(message))

    def authenticate_user(self):
        self.user_info.clear()
        print(self.user_info)
        message = ["Enter Username\n",
                   "Enter Password \n"]
        self.clientsocket.send(pickle.dumps(message))
        self.user_info = pickle.loads(self.clientsocket.recv(1024))
        if(self.user_info[0] in authorization and authorization[self.user_info[0]] == self.user_info[1]):
            message = "User Valid!\n"
            self.clientsocket.send(pickle.dumps(message))
        else:
            message = "User InValid!\n"
            self.clientsocket.send(pickle.dumps(message))

    def authorization_user(self):
        self.user_info.clear()
        print(self.user_info)
        message = ["Enter Username\n",
                   "Enter Password \n", "Request of Recourse R1/R2/R3\n"]
        self.clientsocket.send(pickle.dumps(message))
        self.user_info = pickle.loads(self.clientsocket.recv(1024))
        if(self.user_info[0] in authorization and authorization[self.user_info[0]] == self.user_info[1] and authorization[self.user_info[4]] == 'yes'):
            message = "Resource allocated to user\n"
            self.clientsocket.send(pickle.dumps(message))
        else:
            message = "Resource not allocated to user\n"
            self.clientsocket.send(pickle.dumps(message))
```

Client:

```
# Import socket module
import socket
import pickle
# Create a socket object
s = socket.socket()

# Define the port on which you want to connect
port = 5050

# connect to the server on local computer
s.connect(('192.168.100.50', port))
print("Connected with server")
user=[]
while True:
    choice=input(pickle.loads(s.recv(1024)))
    s.send(pickle.dumps(choice))
    message = (pickle.loads(s.recv(1024)))
    for i in message:
        user.append(input(i))
    s.send(pickle.dumps(user))
    message = (pickle.loads(s.recv(1024)))
    print(message)
    continue
    # s.close()
    # break
# close the connection
```