ECE 5650: Project 1

Colin Phlypo

Robert Kruzel

October 2, 2017

Introduction

Project 1 was to create a customer data base server using Python 2.7 socket programming based on skeleton code of a class for a database.

Source Code

Refer to .py attachment/code below.

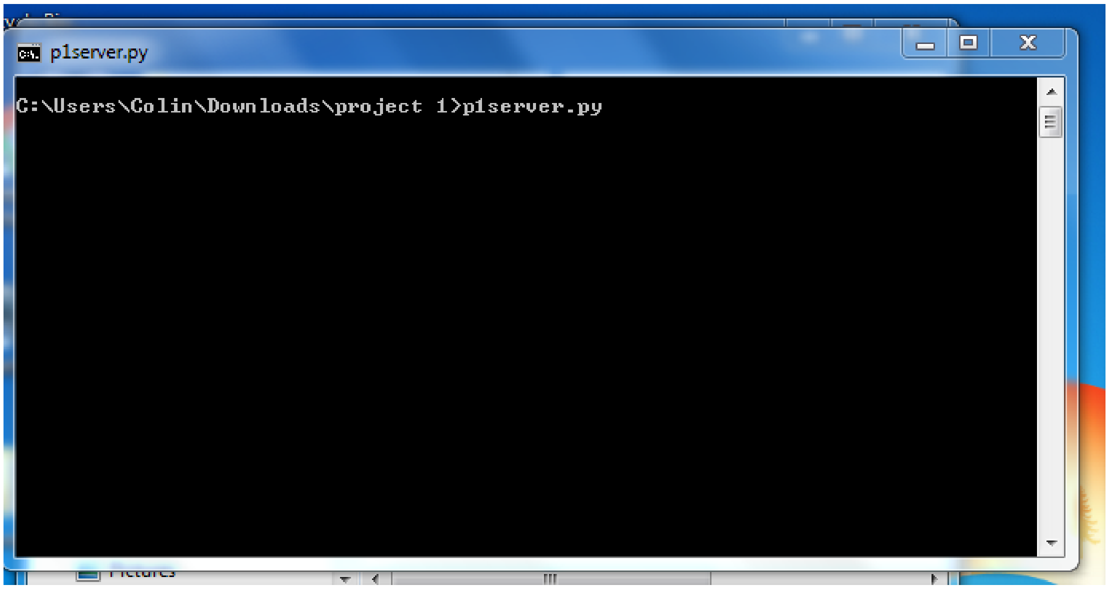
```
# ECE 5650: Project 1
# Names: Robert Kruzel, Colin Phlypo
# Program is interpreted correctly.
# Program does not give correct results.
# Explanation: In case of command Search/Remove, customerID is used. Our program was
unable to find this therefore,
# outputs no record when the record does exist. Otherwise, program runs correctly

import socket
import sys

class Node: # Customer Record Class
    def __init__(self,customerID, customerFirstName, customerLastName, customerPhone):
        self.id = customerID
        self.first = customerFirstName
        self.last = customerLastName
        self.phone = customerPhone

    def display(self):
    # display record
        return 'Customer record: ' + str(self.id) + ' ; ' + self.first + ' ; ' + self.last + ' ; ' +
self.phone

class customerDB: # Customer Database Class
    def __init__(self):
        self.db = []

    def display(self): # display all records
        allRecords = ''
        for record in self.db:
            allRecords = allRecords + record.display() + '\n'
        if (allRecords == ''):
            return 'Database is empty!'
        else:
            return allRecords
    def insert(self, record): # insert a record
        self.db.append(record)
        return 'Operation was completed successfully.'

    def remove (self, customerID): # remove a record
        found = False
        for record in self.db:
            if (record.id  == customerID):
                self.db.remove(record)
                found = True
                break
        if (found):
            return 'Operation was completed successfully.'
        else:
            return 'ERROR: No match was found!'

    def search(self, last): # search for all records with the specifiedlast name
        found = False
        matchingRecords = ''
        for record in self.db:
            if (record.last == last):
                matchingRecords = matchingRecords + record.display()
        if (matchingRecords == ''):
            return 'ERROR: No match was found!'
        else:
            return matchingRecords

    def show (self, customerID): # show record with specified ID
        found = False
        for record in self.db:
            if (record.id == customerID):
                return record.display()
                found = True
                break
        if (not found):
            return 'ERROR: No match was found!'

nextCustomerID = 0

cDB = customerDB()
# create customer database

HOST = "10.0.0.5"
```

```
PORT = 8889

servSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
servSocket.bind((HOST, PORT))

servSocket.listen(5)    #2.7 doesnt assign on no parameter given
serverRunning = True

while serverRunning:
    connection, address = servSocket.accept()    #blocks here til connection is made
    clientActive = True                                        #inner loop exit
condition

    while clientActive:
        mOutSucc = "Operation was completed successfully."
        mOutErr = "ERROR: The operation is not supported!"
        message = connection.recv(1024)
        command = message.split()


        if (command[0] == "display"):
            mOut = customerDB.display(cDB)
            connection.send(mOut)

        elif (command[0] == "show"):
            mOut = customerDB.show(cDB,command[1])
            connection.send(mOut)

        elif (command[0] == "insert"):
            nextCustomerID = nextCustomerID +1
            record = Node(nextCustomerID,command[1],command[2],command[3])
            mOut = customerDB.insert(cDB,record)
            connection.send(mOut)


        elif (command[0] == "remove"):
            mOut = customerDB.remove(cDB, command[1])
            connection.send(mOut)


        elif (command[0] == "search"):
            mOut = customerDB.search(cDB, command[1])
            connection.send(mOut)
```

```
                elif (command[0] == "exit"):
                        mOut = "Connection Closed"
                        connection.send(mOut)
                        clientActive = False
                        serverRunning = False

                else:
                        connection.send(mOutErr)


servSocket.close()      #end of program
```

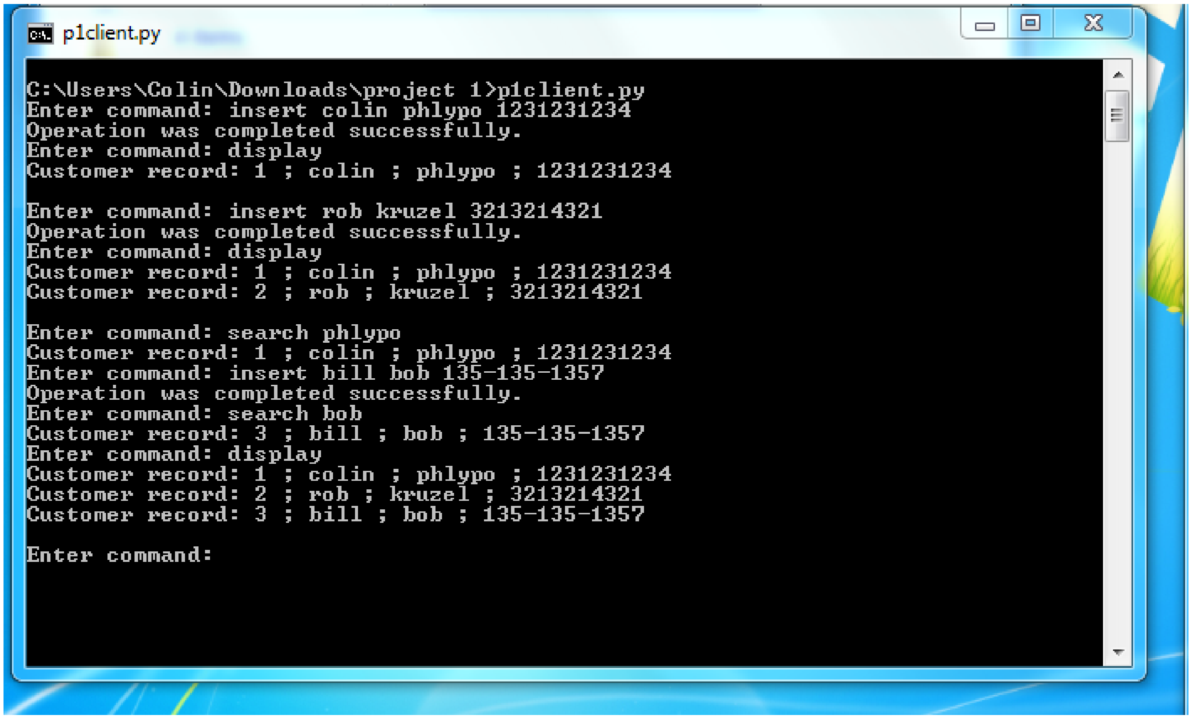
Testing Procedure

To test, the programs were run in two separate command windows. One was the server program and the other was the client program. The process was repeated until bugs/errors were removed. Refer to can screenshots below to see how this was done.



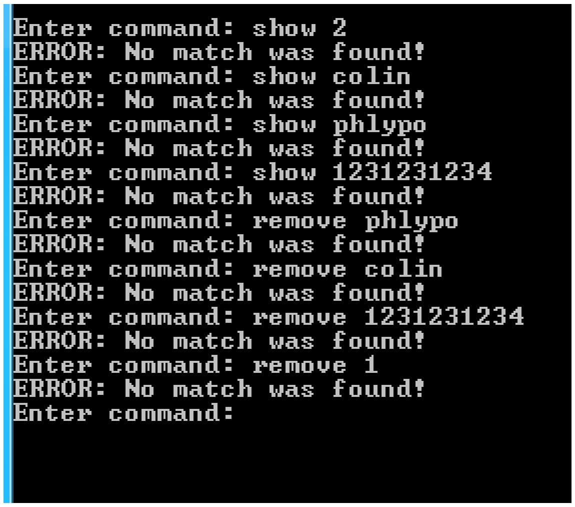
```
C:\Users\Colin\Downloads\project 1>p1server.py
```

Figure 1: Server running in cmd window



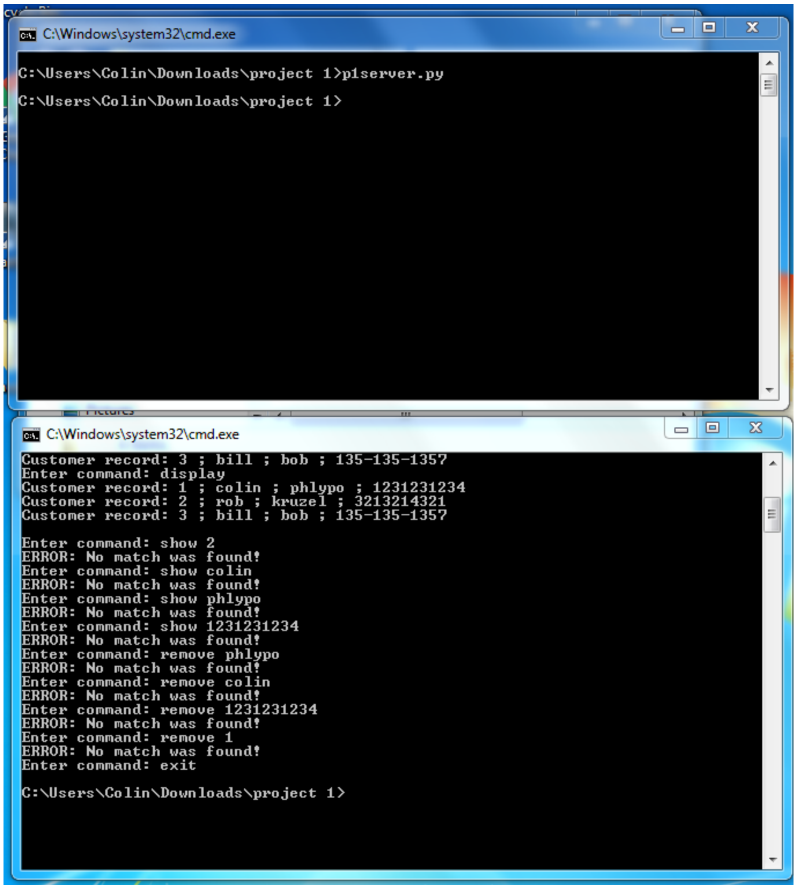
```
C:\Users\Colin\Downloads\project 1>p1client.py
Enter command: insert colin phlypo 1231231234
Operation was completed successfully.
Enter command: display
Customer record: 1 ; colin ; phlypo ; 1231231234

Enter command: insert rob kruzel 3213214321
Operation was completed successfully.
Enter command: display
Customer record: 1 ; colin ; phlypo ; 1231231234
Customer record: 2 ; rob ; kruzel ; 3213214321

Enter command: search phlypo
Customer record: 1 ; colin ; phlypo ; 1231231234
Enter command: insert bill bob 135-135-1357
Operation was completed successfully.
Enter command: search bob
Customer record: 3 ; bill ; bob ; 135-135-1357
Enter command: display
Customer record: 1 ; colin ; phlypo ; 1231231234
Customer record: 2 ; rob ; kruzel ; 3213214321
Customer record: 3 ; bill ; bob ; 135-135-1357

Enter command:
```

Figure 2: Client running in cmd window with error free operations



```
Enter command: show 2
ERROR: No match was found!
Enter command: show colin
ERROR: No match was found!
Enter command: show phlypo
ERROR: No match was found!
Enter command: show 1231231234
ERROR: No match was found!
Enter command: remove phlypo
ERROR: No match was found!
Enter command: remove colin
ERROR: No match was found!
Enter command: remove 1231231234
ERROR: No match was found!
Enter command: remove 1
ERROR: No match was found!
Enter command:
```

Figure 3: Show & Remove commands running in cmd window with error. We believe this is due to error in searching the customerID.



```
C:\Users\Colin\Downloads\project 1>p1server.py

C:\Users\Colin\Downloads\project 1>
```

```
Customer record: 3 ; bill ; bob ; 135-135-1357
Enter command: display
Customer record: 1 ; colin ; phlypo ; 1231231234
Customer record: 2 ; rob ; kruzel ; 3213214321
Customer record: 3 ; bill ; bob ; 135-135-1357

Enter command: show 2
ERROR: No match was found!
Enter command: show colin
ERROR: No match was found!
Enter command: show phlypo
ERROR: No match was found!
Enter command: show 1231231234
ERROR: No match was found!
Enter command: remove phlypo
ERROR: No match was found!
Enter command: remove colin
ERROR: No match was found!
Enter command: remove 1231231234
ERROR: No match was found!
Enter command: remove 1
ERROR: No match was found!
Enter command: exit

C:\Users\Colin\Downloads\project 1>
```

Figure 4: Exit command closing python operation in each window.

Conclusion

This project was an introductory to socket programming and communicating between a server and a client with python to create and view a customer data base. We were able to send server commands from the client. At the server end, the commands would be split into a usable fashion and distributed to the proper functions with the provided class.