Name: Bhaswata Sarkar

BCSE 3rd Year, 2nd Semester

Roll no. : 001910501080

*Internet Technology Assignment#1*

**Problem statement:**

Implement a key-value store using socket programming. The server implements the key-value store and clients make use of it. The server must accept clients’ connections and serve their requests for ‘get’ and ‘put’ key value pairs. All key-value pairs should be stored by the server only in memory. Keys and values are strings.

The client accepts a variable no of command line arguments where the first argument is the server hostname followed by port no. It should be followed by any sequence of “get <key>” and/or “put <key> <value>”.

./client 192.168.124.5 5555 put city Kolkata put country India get country get city get Institute India

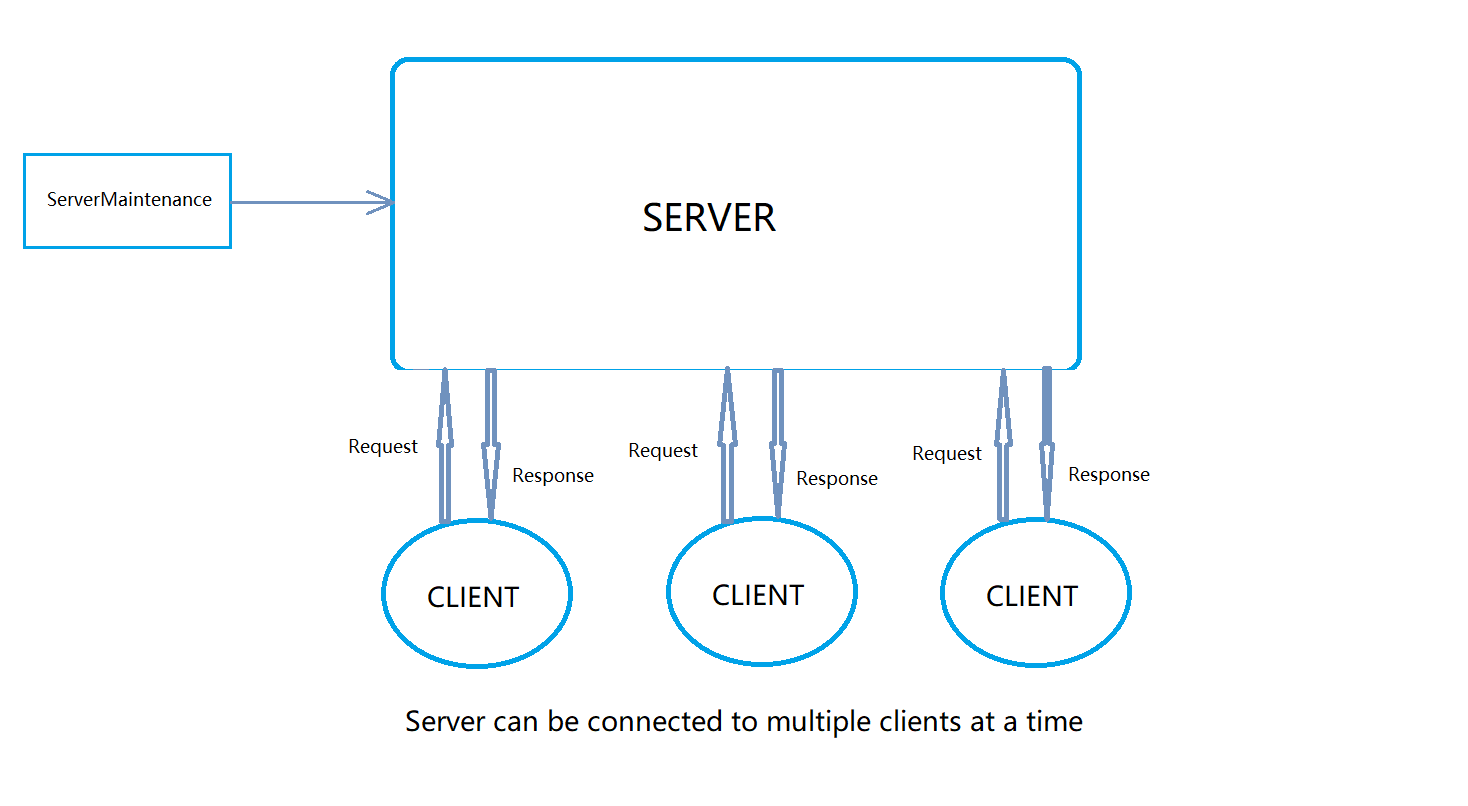
Kolkata

<blank>

The server should be running on a designated port no. The server should support multiple clients and maintain their key-value stores separately. Comment on the port nos used by the server and the clients.

Implement authorization so that only a few clients having the role “manager” can access other’s key-value stores. A user is assigned the “guest” role by default. The server can upgrade a “guest” user to a “manager” user.

**Design:**



**Features:**

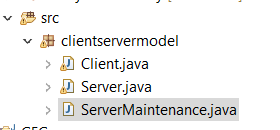
1. Multiple clients can be connected
2. Manager access to selected clients who knows the password(as mentioned in question)
3. “put” or “get” values as many times as the client wants(as mentioned in question)
4. Client “log out”
5. Managing Server to degrade any “manager” to a “guest” users
6. Guest users can access only their data while manager can access anyone’s data

**Implementation:**

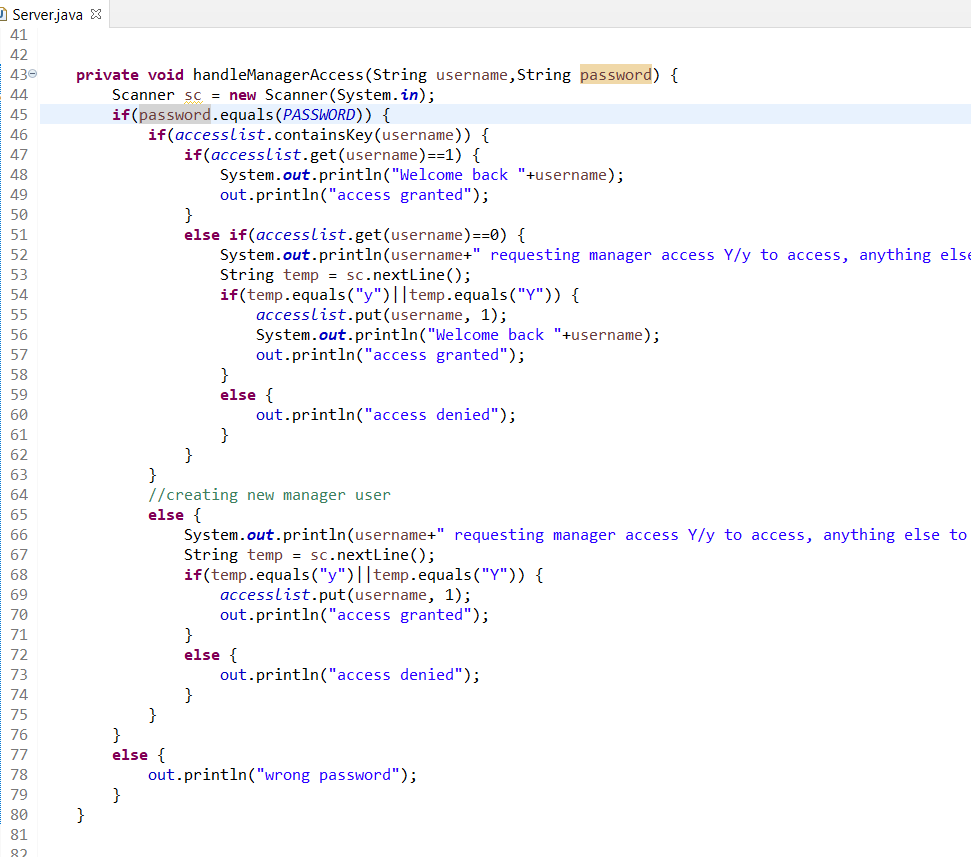
Every communication happening between client and server is via socket. Server is multithreaded and each server instance will handle one client and that instance will store data of that particular client. There is also a static hashmap in server to store all data of all clients for the support of manager access. Another static hashmap is also there to keep the information if the client is guest or manager. Client will send information about what it wants and accordingly send messages and server will send the data accordingly. There is also a module for degrading an manager user to a guest user when required.

**Code Snippet:**

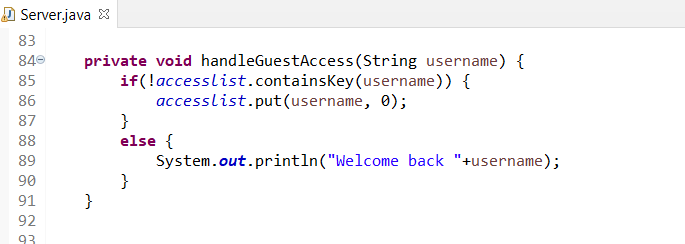
All files in this project.



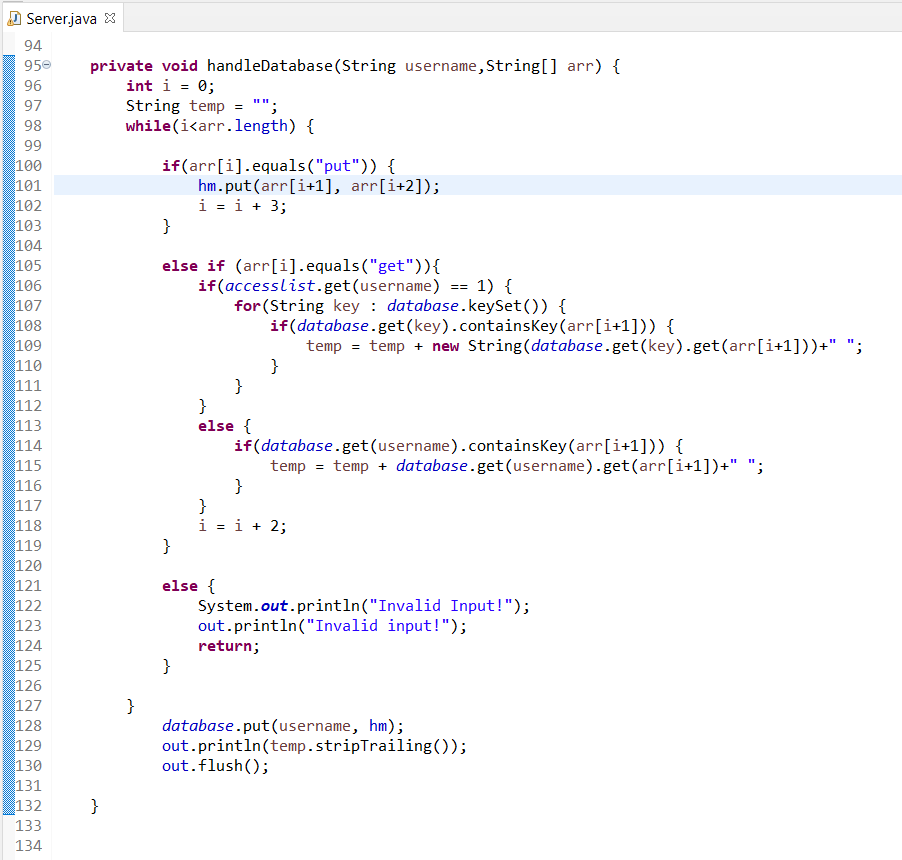
Handling clients who want manager access:



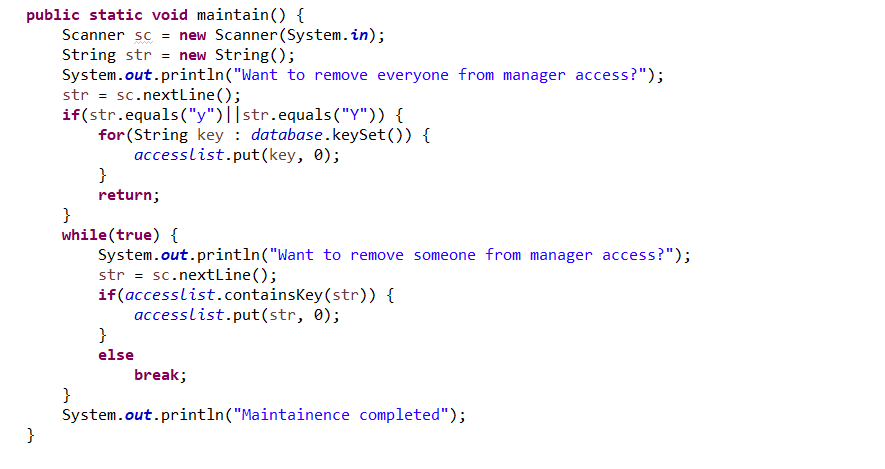
Handling clients for guest access:



Handling the database:



Server maintenance:

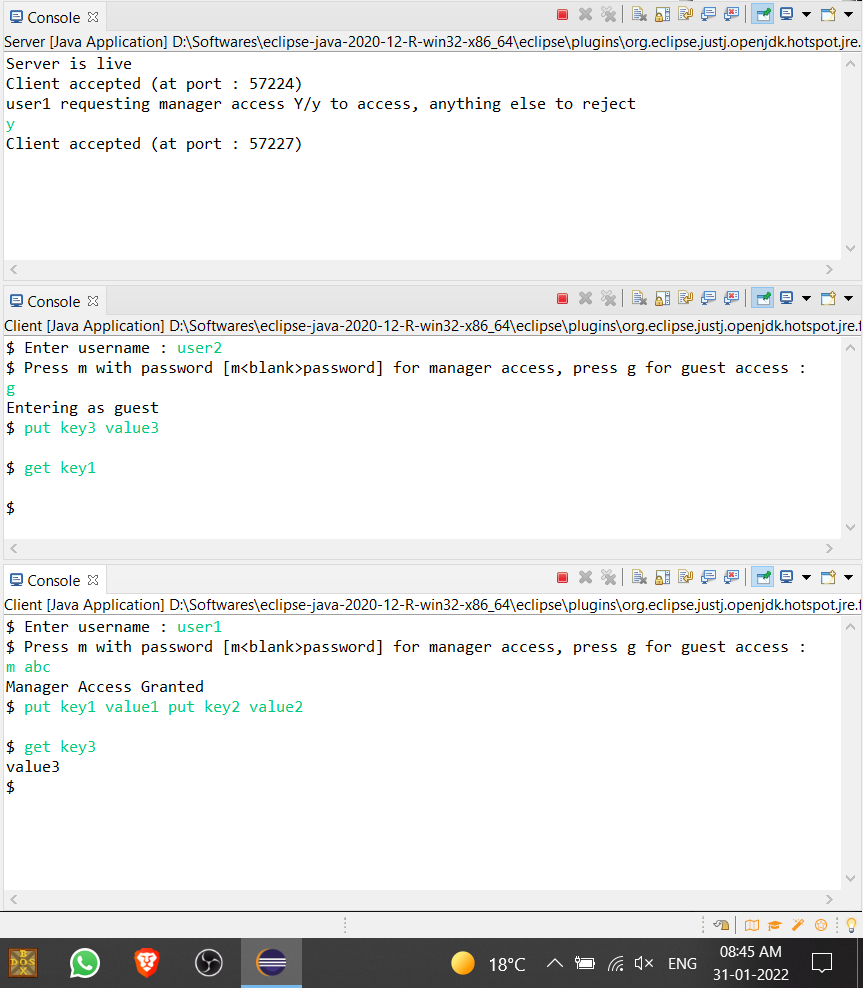


Logging in at Client side, client will ask server to check if password is correct or not in case of manager access.



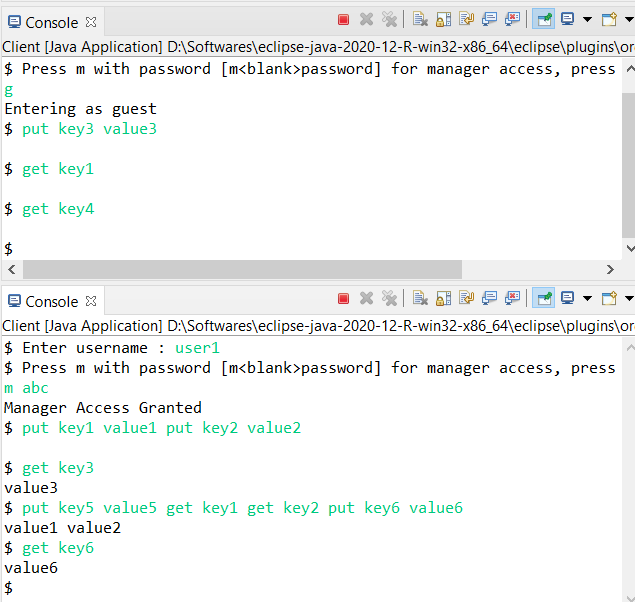
**Output examples:**

Example of a manager and guest:



Manager access needs the password and manager can access the data of guest. Though guest can access only his data.

We can provide input and get output this way too(multiple ‘get’ and ‘put’ statements)



**Conclusion:**

This assignment was simple. It helped me to understand the Client-Server model and TCP.