Design and Implement a Key-Value store over TCP

Souvik Dutta (001810501070)

Introduction

The following assignment is implemented using Java, with OOP as the programming principle.
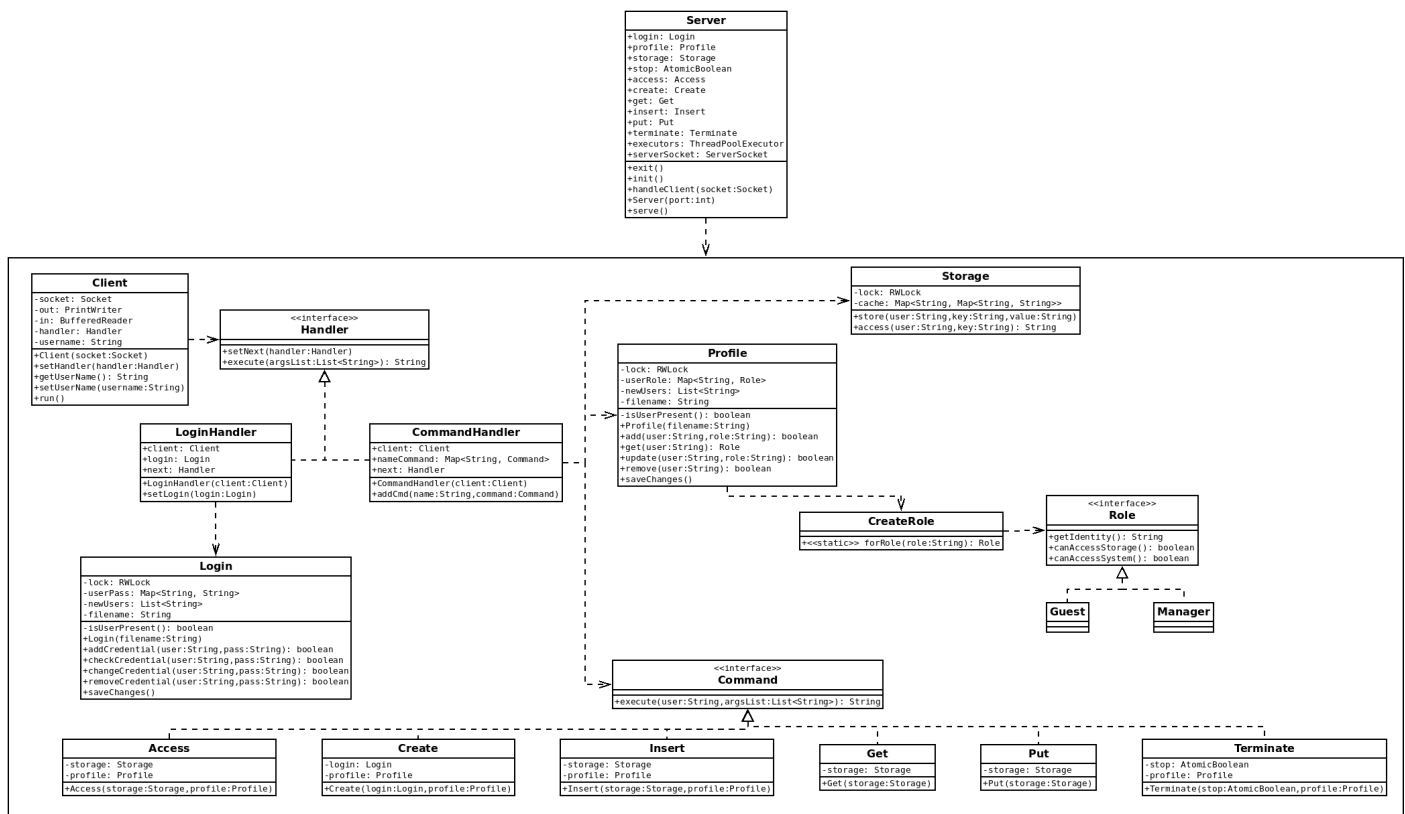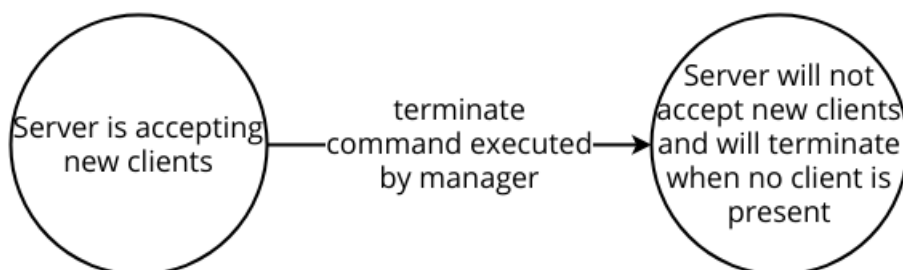

Fig. Implementation of Server

Socket Connection

- *java.net* library is used to establish the Socket over TCP connection.
  - *java.net.ServerSocket* is used at the <u>server side</u> to listen to and accept incoming client connections.
  - *java.net.Socket* is used at the <u>client side</u> to establish connection
- The server uses *java.net.ThreadPoolExecutor* to handle the clients concurrently.

Server Client Architecture

Server

- Server listens to and accepts Clients concurrently. A threadpool (numThreads = maxCpuThreads) is used to ensure the concurrent execution of the Client Commands.
- Server maintains two kinds of states:
  - Global state

  - Local state (wrt to each Client)

- The Server is responsible for
    - Parsing the incoming string from each client
    - Interpreting it with respect to the state of the server and client role
    - Executing the interpreted instruction
    - Sending the response over the connection

Client

- The Client takes the user input as a string and sends it to the Server
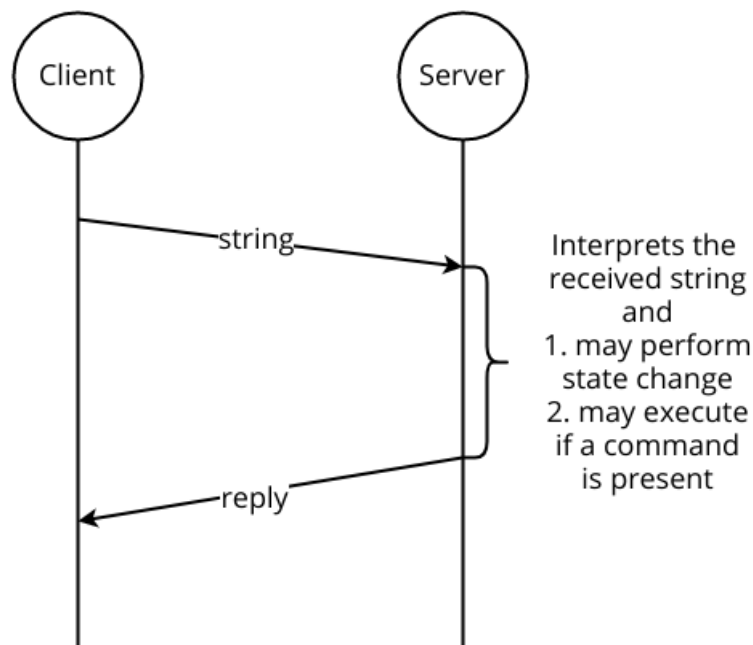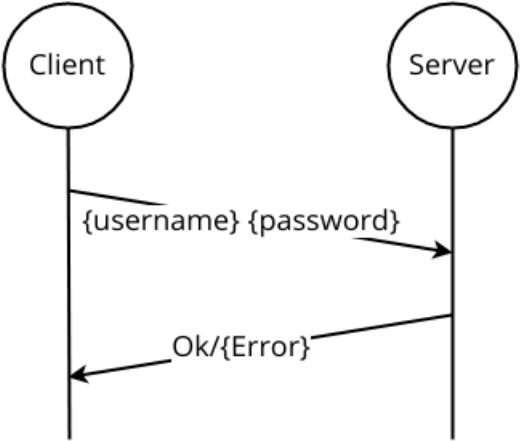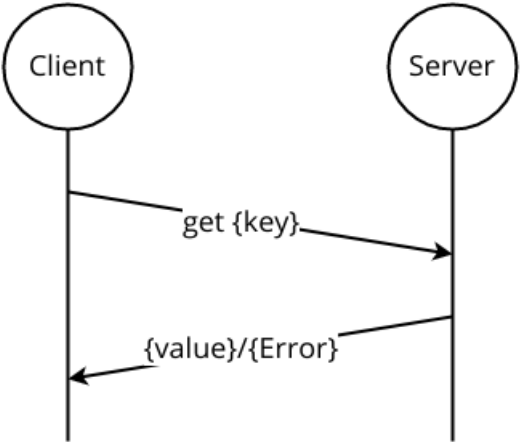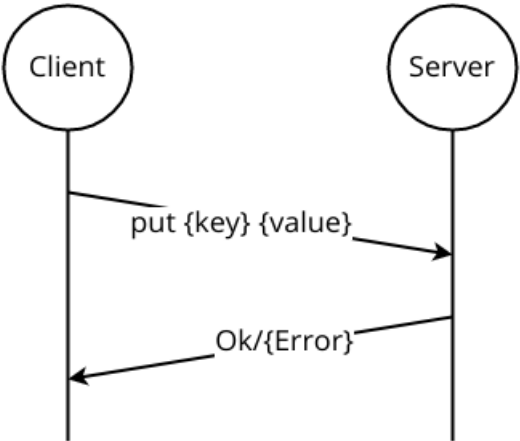- The Client displays the Server response to the user.

Fig. Communication between Server and Client

Message Structure

Command	Supported Role	Structure
login	guest, manager	Client → Server: {username} {password} Server → Client: Ok/{Error}
get	guest, manager	Client → Server: get {key} Server → Client: {value}/{Error}
put	guest, manager	Client → Server: put {key} {value} Server → Client: Ok/{Error}

create	manager	 Client → Server: create {user} {pass} {role} Server → Client: Ok/{Error}
access	manager	 Client → Server: access {user} {key} Server → Client: {value}/{Error}
insert	manager	 Client → Server: insert {user} {key} {value} Server → Client: Ok/{Error}
terminate	manager	 Client → Server: terminate Server → Client: Ok/{Error}

Screenshot

```
subho@pop-os:~/Programming/Internet-Technology/ServerClient/out/production/Serv
erClient$ java server.ServerMain
emjzxika bfnmknzk
Create new accounts using this
Accepting client:: 1
/127.0.0.1 50808
Accepting client:: 2
/127.0.0.1 50810
Accepting client:: 3
/127.0.0.1 50812
subho@pop-os:~/Programming/Internet-Technology/ServerClient/out/production/Serv
erClient$
```

```
subho@pop-os:~/Programming/Internet-Technology/ServerClient/out/production/Ser
verClient$ java client.ClientMain
Connected
[14:33:48.515026]emjzxika bfnmknzk
Login Successful
[14:34:48.082097]create souvik s0uv1k manager
New profile created
[14:35:03.536864]insert subho city kolkata
Successfully stored
[14:37:07.987196]terminate
Terminating server
[14:37:37.095512]exit
Exit successful
Stopping communication
subho@pop-os:~/Programming/Internet-Technology/ServerClient/out/production/Ser
verClient$
```

```
subho@pop-os:~/Programming/Internet-Technology/ServerClient/out/production/Serv
erClient$ java client.ClientMain
Connected
[14:34:12.756276]souvik s0uv1k
Login Successful
[14:35:17.636787]create subho subh0
Three Arguments Required
[14:35:40.689642]create subho subh0 guest
New profile created
[14:35:57.659552]access subho city
durgapur
[14:36:48.361362]exit
Exit successful
Stopping communication
subho@pop-os:~/Programming/Internet-Technology/ServerClient/out/production/Serv
erClient$
```

```
Connected
[14:34:34.332814]subho subh0
Login Successful
[14:36:05.376985]get city
No value returned
[14:36:13.204319]put city durgapur
Successfully stored
[14:36:19.758307]get city
durgapur
[14:36:23.821668]get city
kolkata
[14:37:14.843223]terminate
Role not supported
[14:37:18.154372]exit
Exit successful
Stopping communication
subho@pop-os:~/Programming/Internet-Technology/ServerClient/out/production/Ser
verClient$
```