Student Name: Yuhao Zong

Student Number: 1024974

2022 SEM1 COMP90054: Distributed System Assignment 1 Multi-threaded Dictionary Server report

Problem context description

1. **Socket & Threads**

   The assignment has been designed to demonstrate our understanding of two fundamental technologies frequently used among the distributed computing world.

   The **socket** is used to connect a two-way communication link between two programs running on the network. In the problem context, each communication side is described as one client and one dictionary server.

   The **thread** has been considered as a lightweight process in a program, and the program will have multiple threads executing concurrently and also sharing resources. In the problem context, each connection between the client and dictionary server is considered as one thread in the server program.

2. **Interaction**

   2.1. **Communication protocol-TCP**

   In the problem context, it's required to perform connection under two optional communication protocols: TCP and UDP. Both of them have advantages and disadvantages, TCP is called a reliable transmission protocol which means it will ensure the data(packet) transfer during the network won't be lost or perform data(packet) recovery strategy. The UDP protocol is called a connectionless protocol which performs faster in data transmission than TCP but doesn't ensure reliability. In this problem, I'll choose the TCP as my transmission protocol, as the dictionary server will be performed better under a reliable transmission environment, which ensures each response and request will arrive at the correct destination.

Student Name: Yuhao Zong

Student Number: 1024974

2.2. Data interchange format-JSON

In the data transmission, the data format used is JSON format, it is a common data format used very frequently during web applications with servers. And it is an easy-to-learn and human understandable format. The format sending from client to the dictionary server can be simple as only three attributes are “RequestType”, “Word” and “Definition”. No need to perform data processing on the dictionary server side.

3. Failure Model

As the data transmission protocol used is TCP connection, therefore, the reliability is guaranteed.

4. Functional Requirements

- **4.1. GET:** Client asking for a word’s definition.
- **4.2. POST:** Client adding a new word with definition.
- **4.3. DELETE:** Client updating an existing word’s definition.
- **4.4. UPDATE:** Client deleting a word from the dictionary.

5. Graphic User Interface

In the context problem, it is required to implement a GUI, this has been implemented by using the WindowBuilder in the Eclipse.

Components of the system

1. **Client**
   Client Class, the main class/program as the client in the problem. Under the TCP connection, it has the ability to perform four request types: “POST”, “UPDATE”, “GET” and “DELETE”.

2. **Server**
   Server Class, the main class/program as the dictionary in the problem. It will keep listening to the client connection and sending responses to those clients concurrently.

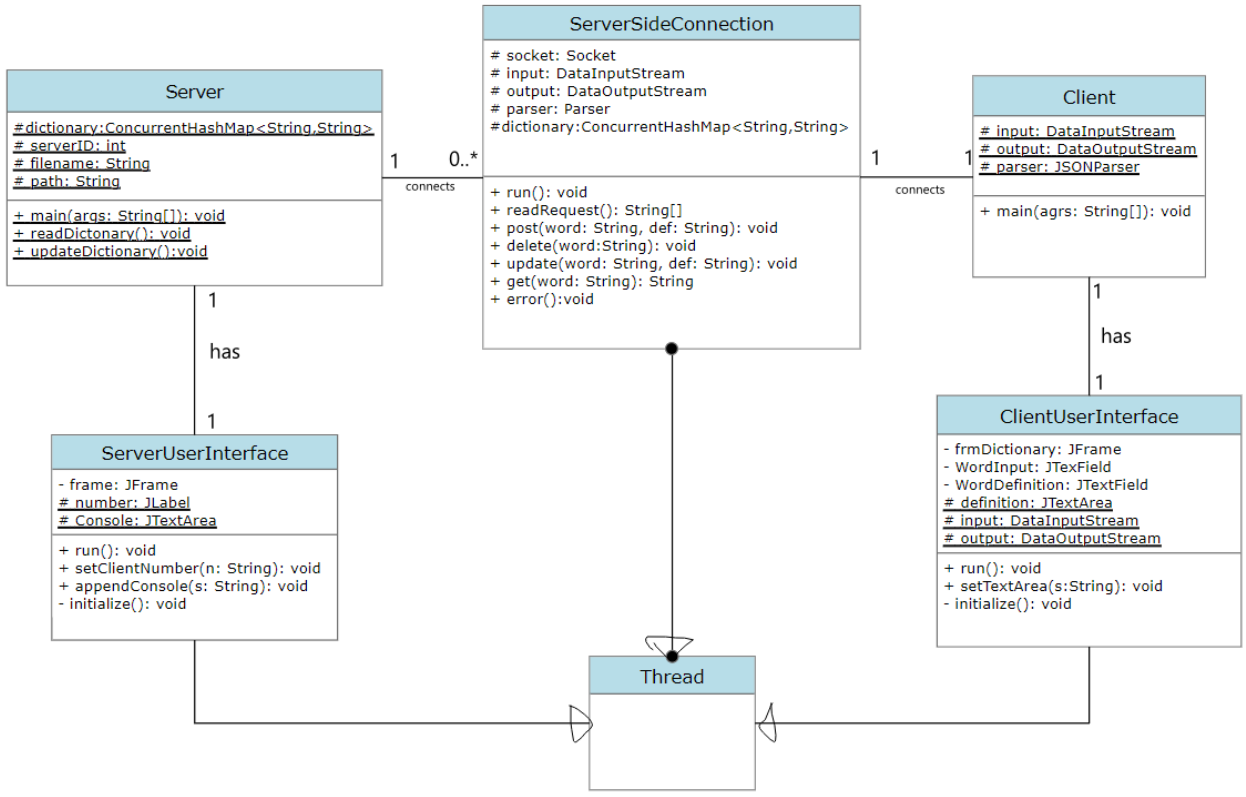
3. **ServerSideConnection**
   This class extends the Thread class, and has the ability to process multiple connections at the same time. It will handle the packet receiving and sending,

and also after it receives the JSON data(packet), it will unpack and process the JSON data, then send the different types of response accordingly.

4. **ClientUserInterface**
   It serves as a window to provide interaction with the client. In the window, you can perform the aboving 4 types of requests, and also receive messages from the dictionary server. Furthemore, it has another duty: after clicking on the button to send a request, it will pack the request into the JSON format and send it to the dictionary server.

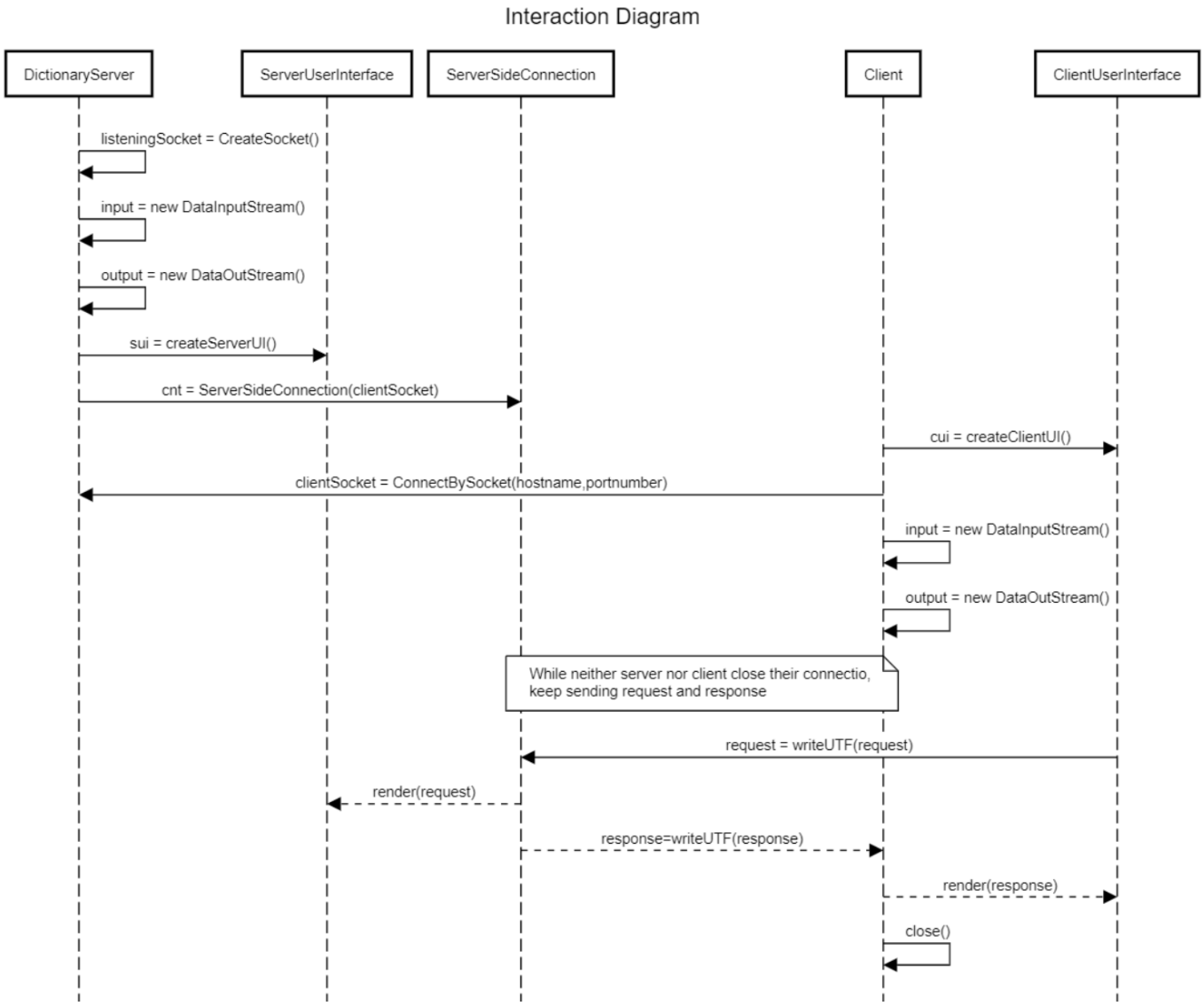
5. **UserUserInterface**
   It serves as a window to provide interaction with the dictionary server. Feature provided: showing the number of online clients, showing the requests from clients.

Class diagram and interaction diagram



Student Name: Yuhao Zong

Student Number: 1024974



Critical analysis

1. **Architecture**

   The architecture used in the assignment is thread per connection, each time the server receives one connection it will create a thread for processing the request and response. It is straightforward to implement in the program and it is suited for the client - dictionary server (multi-threaded) connection. However, it is quite expensive and will also impact the scalability of the server as the number of connections may grow significantly above the ideal amount of thread of the server.

Student Name: Yuhao Zong

Student Number: 1024974

2. **Concurrency**
   As we are performing a multithreaded server, therefore leads to a question: How to perform operations concurrently? One of the properties of thread is they are running in a random order. So, when it comes to multiple clients the server may send the response to the wrong clients(through thread of connection). Using synchronization is necessary, one of the strategies is to put “synchronized” in the method declaration, hence ensuring the order is kept.

   Alternatively, I’ve implemented a **ConcurrentHashMap** to store all the vocabulary, the main difference between a normal HashMap is It allows concurrent access to the map. Therefore, ConcurrentHashMap allows concurrent threads to read the vocabulary in the dictionary without locking at all.

3. **Diction server User Interface**
   It is a newly implemented functionality, a GUI for the diction server to know how many clients are currently online and read all the requested messages from the clients. It helps manage the dictionary server manager.

   Things could be improved: Adding more managing functionalities such as dictionary management (showing all words, delete/update words).

4. **Communication between the client and server**
   All communication messages will appear both on the client side and server side, such as error messages, client/server closed, request/response messages. It helps either client or server to better understand what’s going on in the program and no need to look at the terminal anymore.

Conclusion

The multiple-threaded dictionary assignment improves my understanding of how distributed network transferring packets under TCP communication also the multi-thread programming is also a powerful technique for those problems like multiple connection problems.