

# **COMP90015: Distributed Systems - Assignment1**

## **Multi-threaded Dictionary Server**

**Name:Yuming Lin**

**Login user name:YUMINGL**

**Student number:883717**

**Email:yumingl@student.unimelb.edu.au**

**Tutor:Lakshmi Jagathamma Mohan**

## 1. Introduction

This program implements a multi-client network dictionary query system. The Socket is used to communicate between Client and Server, and the multi-threaded client is implemented. That is, a server can have multiple clients running a dictionary program. Meanwhile, the function of adding words and adding one or more explanations to words is implemented. In addition, there are query and delete functions for words. Finally, the UI interface of both the Client and the Server are also created.

## 2. Development environment

The development environment for this experiment is MacBook air.

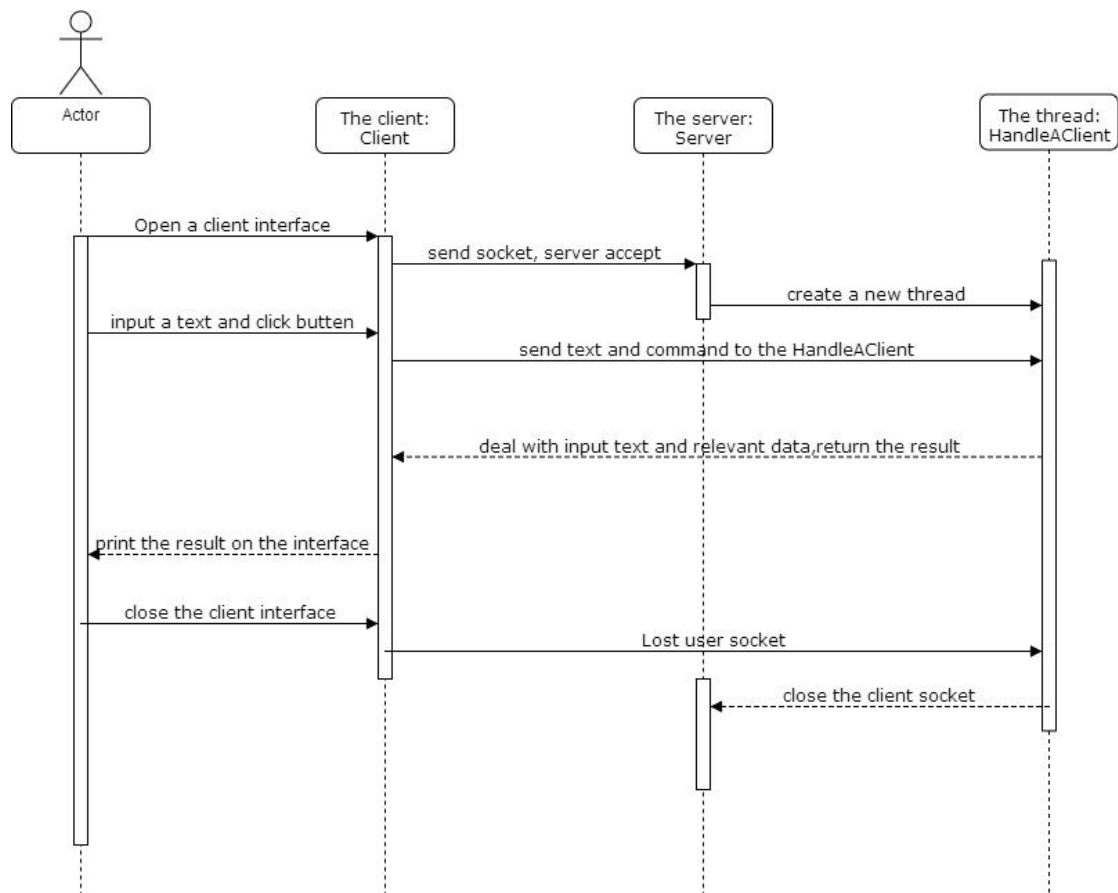
The operating system is macOS High Sierra.

Java development version: 1.8.0\_181

Editor: Sublime Text & Eclipse.

Compiler: Terminal.

## 3. Interaction diagram



Graph 1 Interaction diagram

According to Graph 1, first, the default Server is open. Whenever a user opens the client interface, Server listens to the socket sent by the client and accepts it. Next, open a new HandleAClient thread to handle the user's series of operations.

When the user enters text and clicks on the button, the content is transferred to HandleAClient to perform the corresponding action. Then return the result to the Client. Finally, the results are printed on the user's visual UI interface.

When the user closes a visual UI interface, the socket will be closed. At this point, the HandleAClient accepts the socket close message and reports the message to the server.

#### **4. Program module**

The program is divided into two parts, namely Client and Server. The client graphical interface GUI is built into the Client class.

The Server class on the server side is responsible for communicating with the client. The HandleAClient class is responsible for handling multi-threaded clients and operations such as adding and deleting. It also references a data structure MultiValueMap and its subclass LinkedMultiValueMap.

##### **4.1 Server class**

First, in the Server class, the pre-set port number and dictionary path are passed in. Then, through the cyclic accept command, the incoming socket is received. When a new socket is accepted, a new HandleAClient thread is created, and more details such as one user and the user's IP are displayed.

##### **4.2 HandleAClient class**

The HandleAClient class implements the Runnable interface. First, create the data stream input and output, and then get the command sent by the client class. Next, the command is passed to the control method to perform functions such as adding words, increasing word meaning, finding words, and deleting words according to different commands.

Adding a word function requires typing a word and meaning at the same time. The method first searches for whether the word is in the dictionary, and if the word already exists, the word cannot be added. If the word does not exist, the word is successfully added to the dictionary and its meaning.

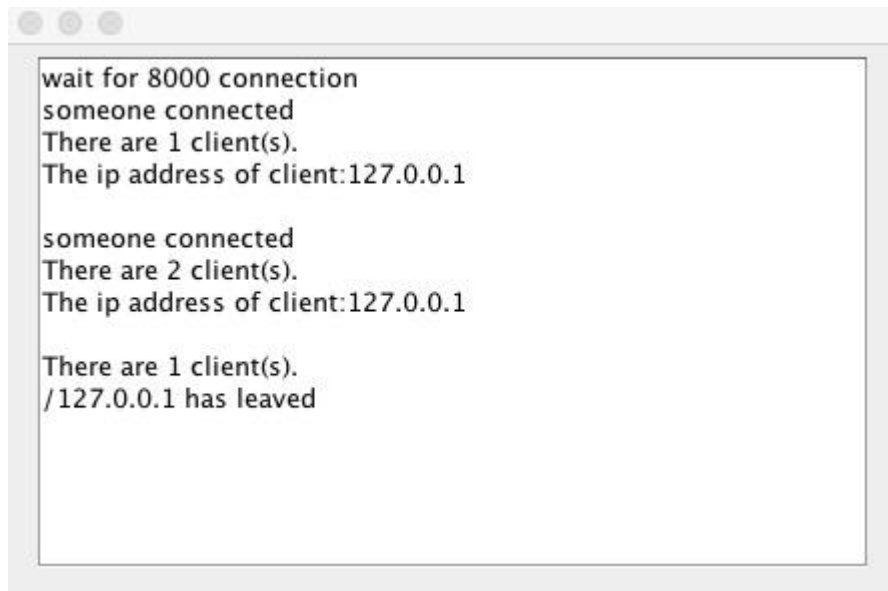
Adding a word meaning function also requires entering a word and meaning. This method first searches for whether the word is in the dictionary, and if the word does not exist, it cannot be added. Otherwise, if the word exists, add an explanation for it.

The Find Words function returns all the meaning of the words already in the dictionary to the client output on the screen.

The delete function first needs to enter a word, and then the word and all its meaning will be deleted from the dictionary.

### 4.3 ServerGUI

As shown in the Graph2, this class implements the Visual UI interface of the Server. According to the different operations of the user and login and logout, the corresponding results will be reflected in real time.



Graph 2 ServerGUI interface

### 4.4 MultiValueMap interface

The method can implement a Key corresponding to multiple Value values and introduces related functional methods. (Docs.spring.io, 2018) (GitHub, 2018)

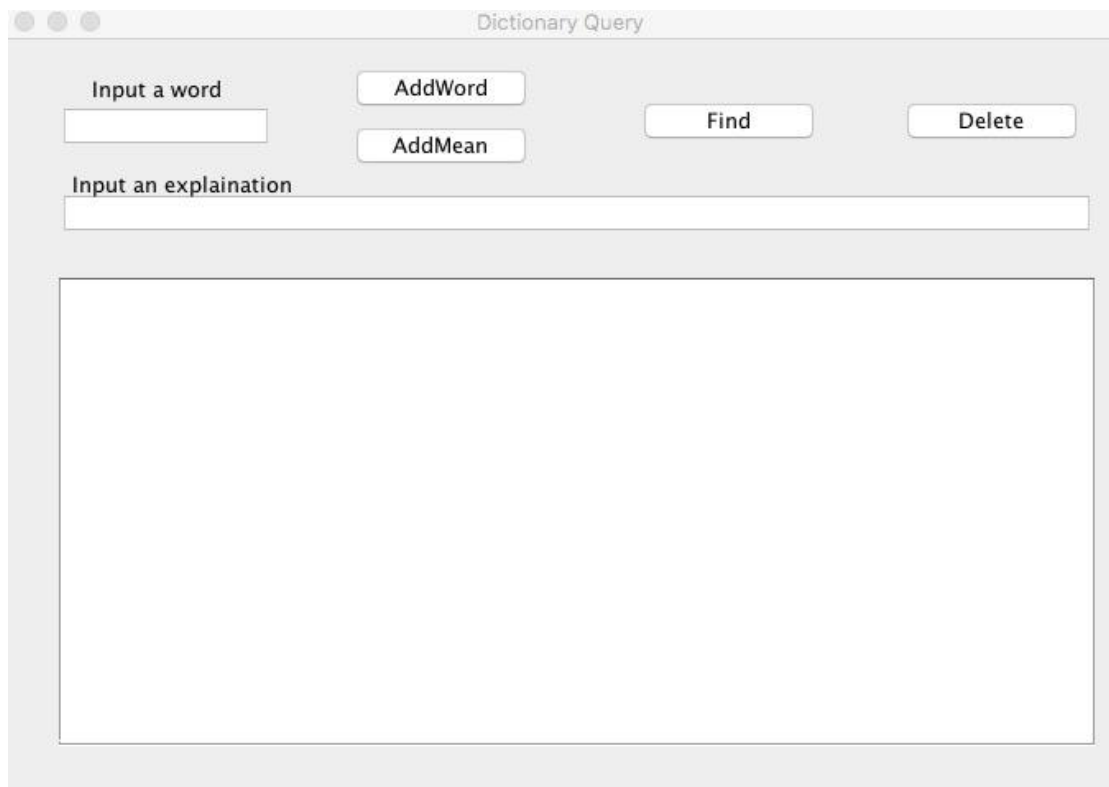
### 4.5 LinkedMultiValueMap class

This class implements MultiValueMap, which can store a Key value corresponding to multiple list[Value]. (Blog.csdn.net, 2018)

The method of use is `Map<K, List<V>> mSource = new LinkedHashMap<K, List<V>>();`

#### 4.6 Client class

As can be seen in the Graph 3, the Client class first passes the set host number and port number in advance and then creates a new data stream input and output. Next, the client UI interface module is initialized, and then the content input by the user in the input text area is obtained according to different instructions. When the user clicks on a different function button, its corresponding function will be executed. At this time, the content input by the user is transmitted to the server, and the dictionary is manipulated by the server. Finally, the server returns the result to the Client and prints the resulting content on the UI interface of the Client. When the UI interface is closed by the user, the connection between the client and the server is broken.



Graph 3 ClientGUI interface

#### 5. Program analysis

The main difficulty of this program is exception capture and exception handling. Next, explain some abnormal solutions in detail. The first is a read-write exception, where synchronized is used to make reads and writes separately to avoid conflicts. Next, the transport Socket exception will also be caught when it occurs. For example, if a Client disconnects, it will cause an EOFException because the socket is not closed. Then the exception is caught and the socket is closed, and the number of offline users is reduced by one.

In addition, if the dictionary file does not exist, a new dictionary file will be generated. Finally, the user enters an error command and the system will generate different feedback based on the error input from the user command.

In the program framework design, due to the unfamiliar use of the GUI interface, the Client and GUI are not divided into different classes. It is preferable to improve this design by separating the separate Client classes into GUI classes and other method classes. In addition, HandleAClient class can also separate functions such as adding, querying and deleting into other independent classes. This will be more in line with object-oriented design ideas.

## Reference

- Blog.csdn.net. (2018). Map 之一个 Key 存多个 Value 的 MultiValueMap(一个键多个值) - CSDN 博客. [online] Available at: <https://blog.csdn.net/yanzhenjie1003/article/details/51550264> [Accessed 6 Sep. 2018].
- Docs.spring.io. (2018). MultiValueMap (Spring Framework 5.0.8.RELEASE API). [online] Available at: <https://docs.spring.io/spring/docs/current/javadoc-api/org/springframework/util/MultiValueMap.html> [Accessed 6 Sep. 2018].
- GitHub. (2018). yanzhenjie/NoHttp. [online] Available at: <https://github.com/yanzhenjie/NoHttp> [Accessed 6 Sep. 2018].