Assignment 1 Report
Zequan He 1068069

Introduction
in this project, I am using TCP protocol to provide a reliable communication between server and clients in the multi-threaded dictionary system. The system's server allows concurrent clients to modify or query remote dictionaries by using a thread-per-request architecture in the server. In addition, I am using JSON format to exchange the message between server and clients. For the failure handling, the system can handle parameter error, I/O error and connection error, and give proper solution on server and client.

Server
In the server side, it is using thread-per-request architecture and handle different failure. For every request from clients, the server would create a thread to process request and give the result response to clients. In addition, the server can handle the request from different clients concurrently because of multi-thread system. Furthermore, server can response different feedback to client due to different requirement from the clients. Also, Server has a GUI that can show the information about the server and some request from the clients.

For the Server design

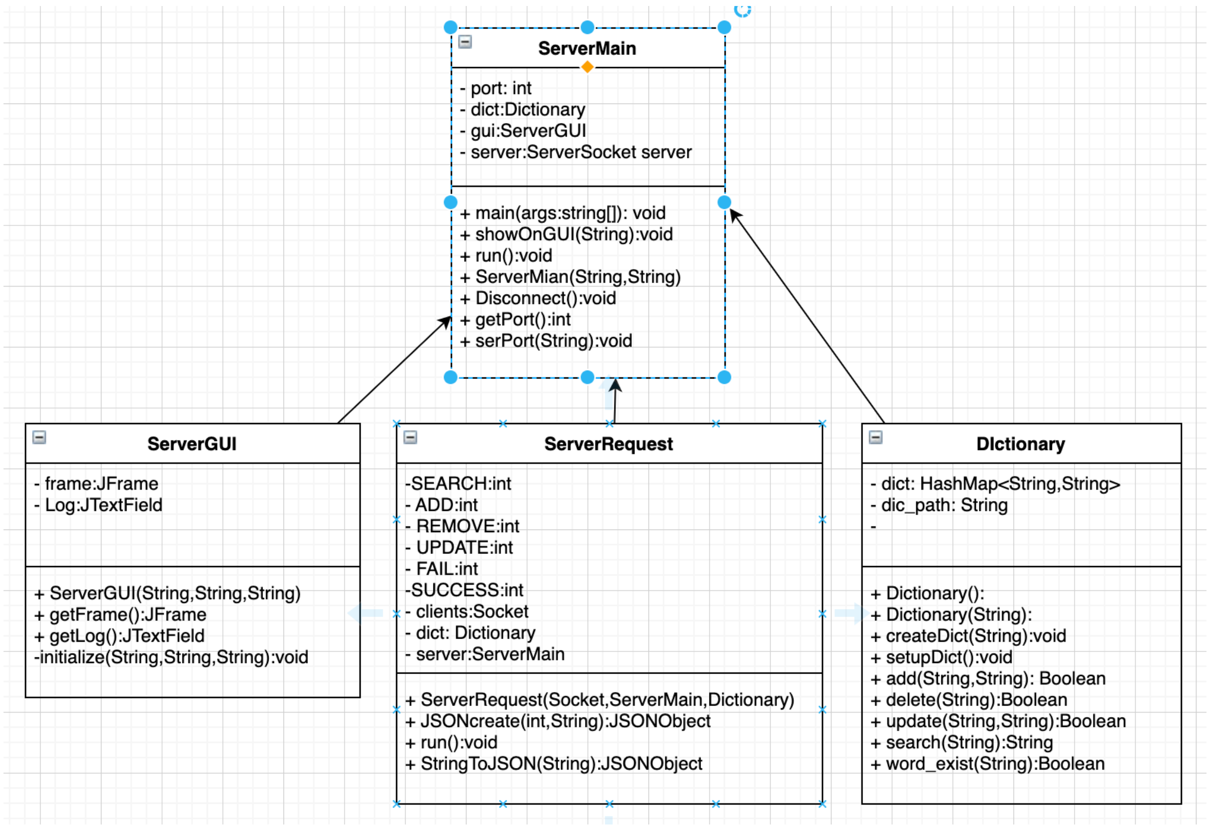
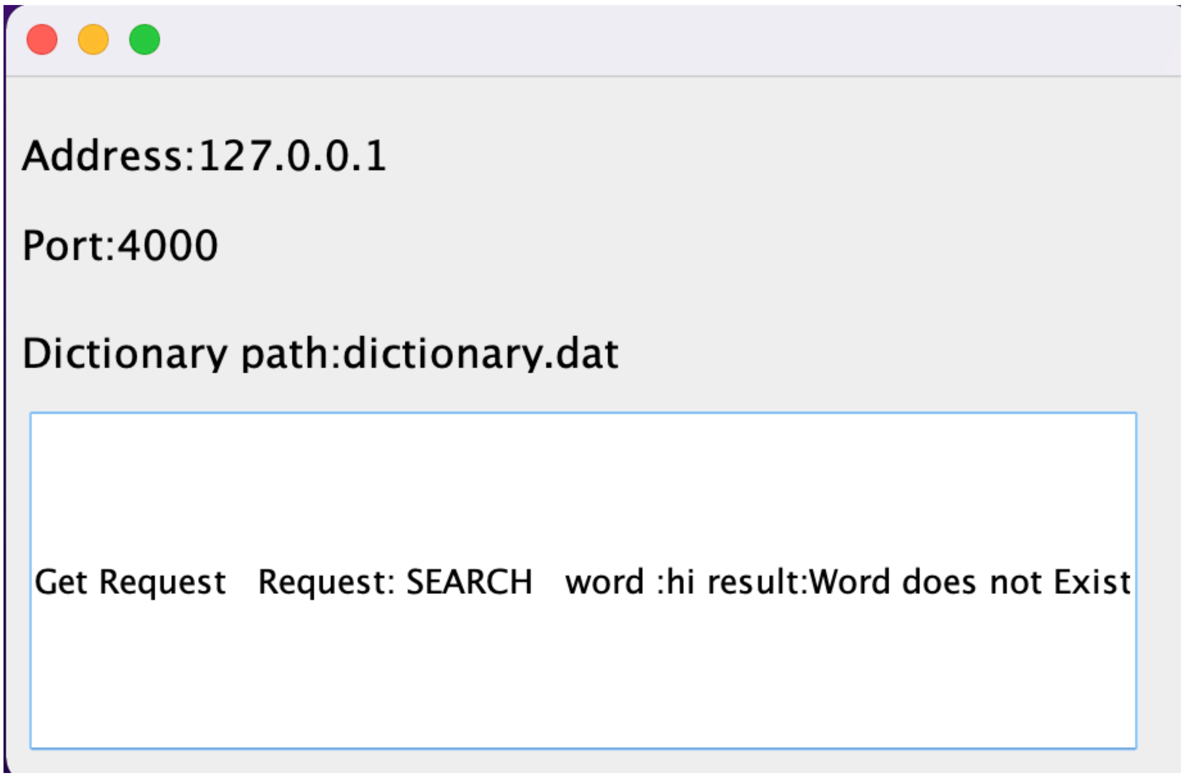


Figure 1: Server System design

For the server start, it will check the port which is given by admin, if is correct, will check the dictionary path is exist and can read or not. if not, the server will create a dictionary which setting itself. Then, when the server receive the request from the client, it will create a thread to handle the request in “ServerRequest” class, and according to requirement from the client, the server will sent different respond to clients with JSON format. The JSON format will contain “state” which tell client is success to get the result or not, and contain the “meaning” which contain the definition of the word.

In addition, server has a GUI to show the information of the server and show the request information from the client which show in figure 2.


Address:127.0.0.1

Port:4000

Dictionary path:dictionary.dat

Get Request   Request: SEARCH   word :hi result:Word does not Exist

Figure 2:Server GUI

Client

In the client side, it support by two part which is Controller which is request handle and GUI which for client to enter the request and send to Controller. For each request which is QUERY,ADD,DELETE,UPDATE given by user, the controller will establish a TCP connection with the server and disconnect after the server give the response and send the result to client. In addition, after the controller receive the result, the GUI will pop up a window that to tell user about the requirement is success or not. if it fail, it will give the error information to the user.

For the client design

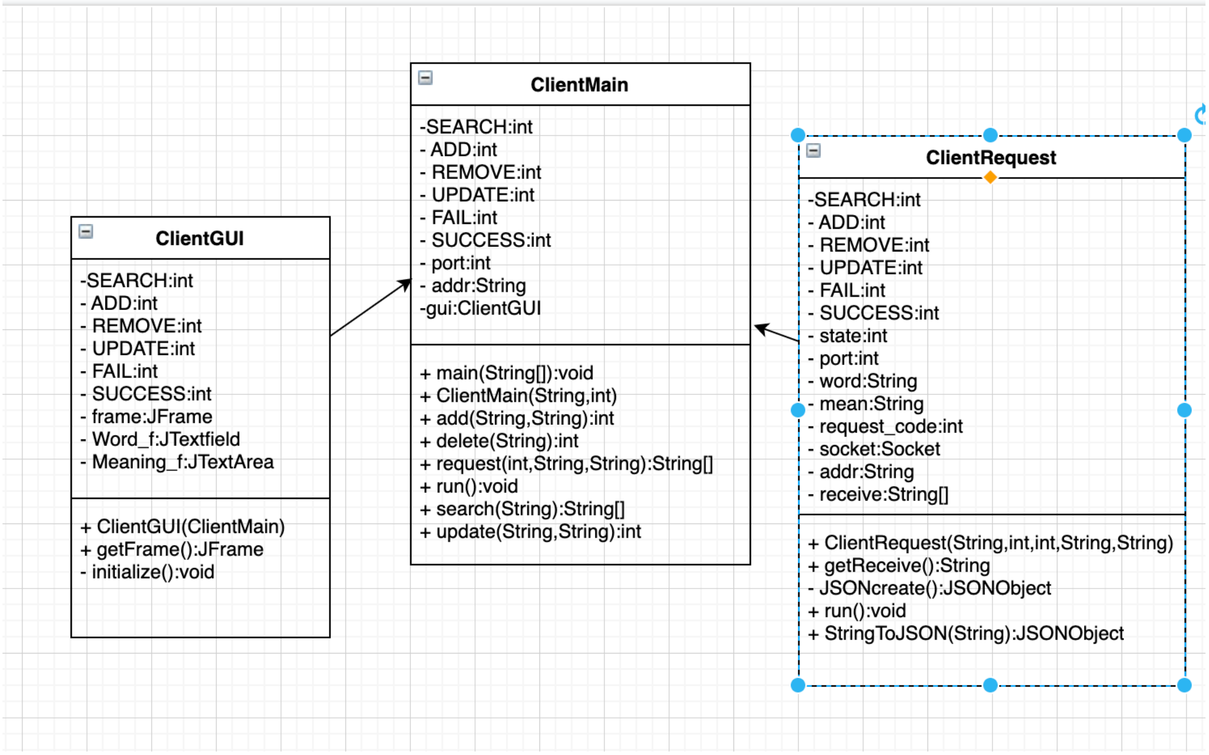


Figure 3: Client system design

For the client run, it will check the user provide the port and server address or not. and then, it will check the port is valid or not. if not, will return the error to user. In addition, the client side will connect and check the server address when user call QUERY,ADD,DELETE,UPDATE. If not, the client will not create the connection with server. When the connection success, the client side will create request in JSON format which contain “request_code”, which means what requirement from client, and “word” and the “meaning” for definition of the word when the ADD request from client. After the receive the respond from server, the GUI will pop up a window to show is success or not, if not will show the reason why it fail.

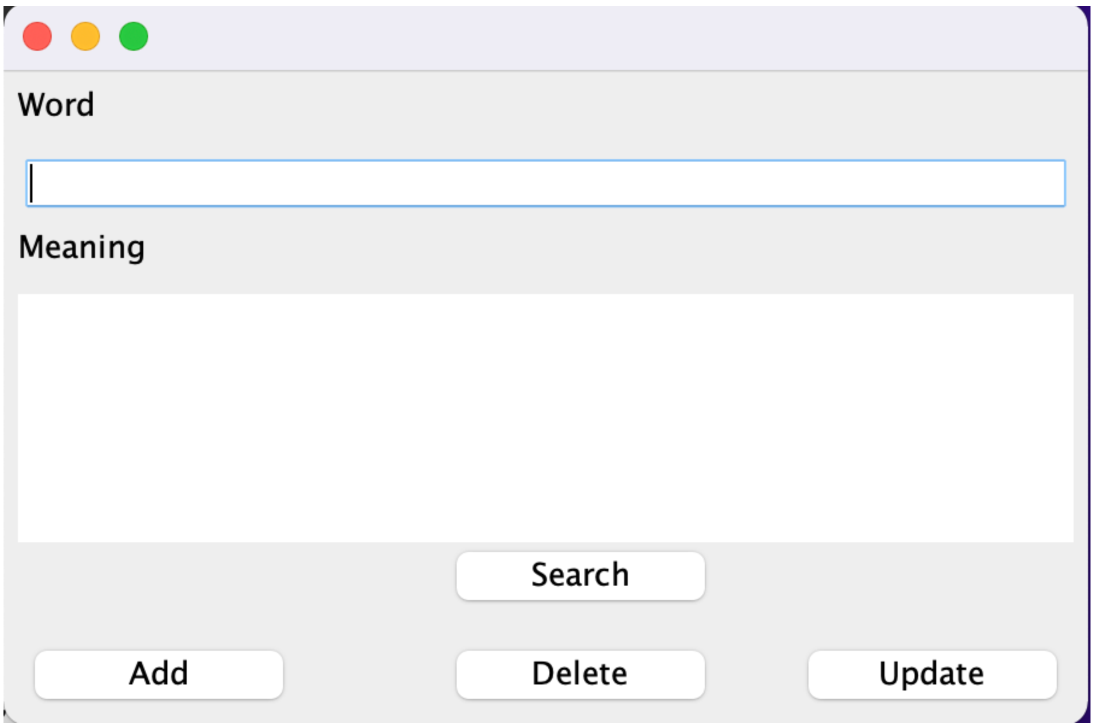
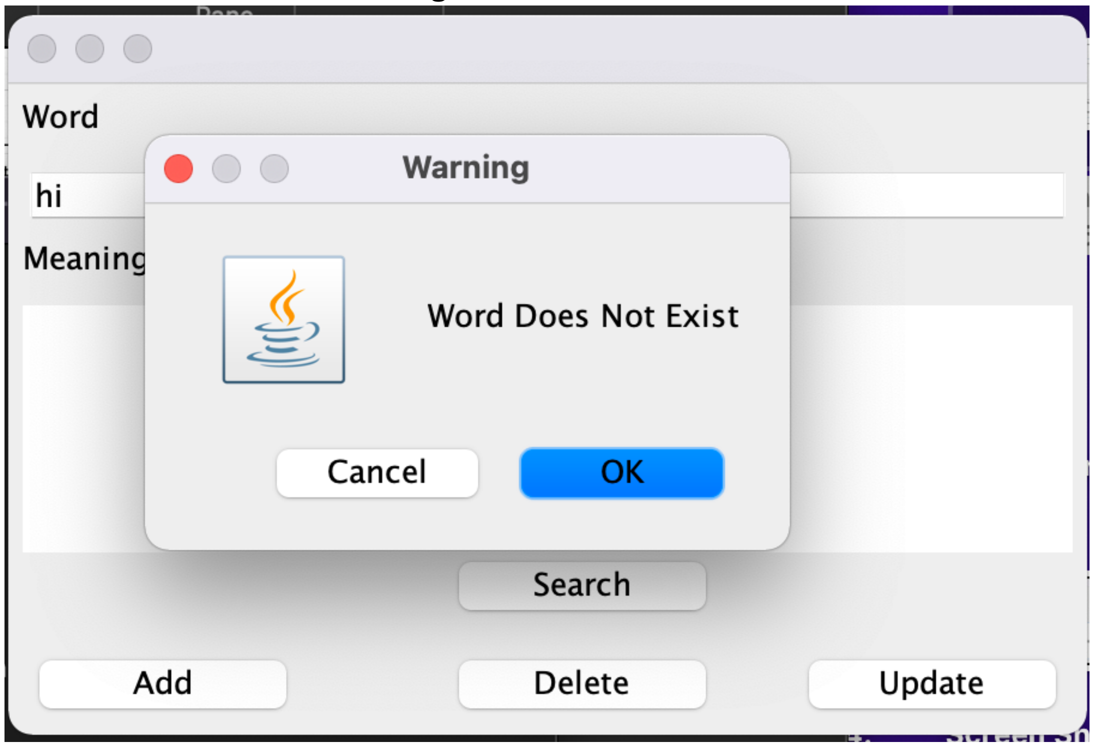


Figure 4:Client GUI



Word Does Not Exist

Figure 5: warning window

Dictionary

For the dictionary, I am using HashMap to store word and the definition, it is because as a dictionary, it may have huge data. So that HashMap can provide a fast search speed which is O(1) for time complexity, it will not take so much for client to wait for the result. Also HashMap has dynamic length storage data, so that admin will not concern about the length of storage data and worry about the length will not be exhausted. And I use .dat format for the default dictionary data, it is because it makes attacker cannot directly modify the dictionary content in the server backend.

For the Interaction Diagram

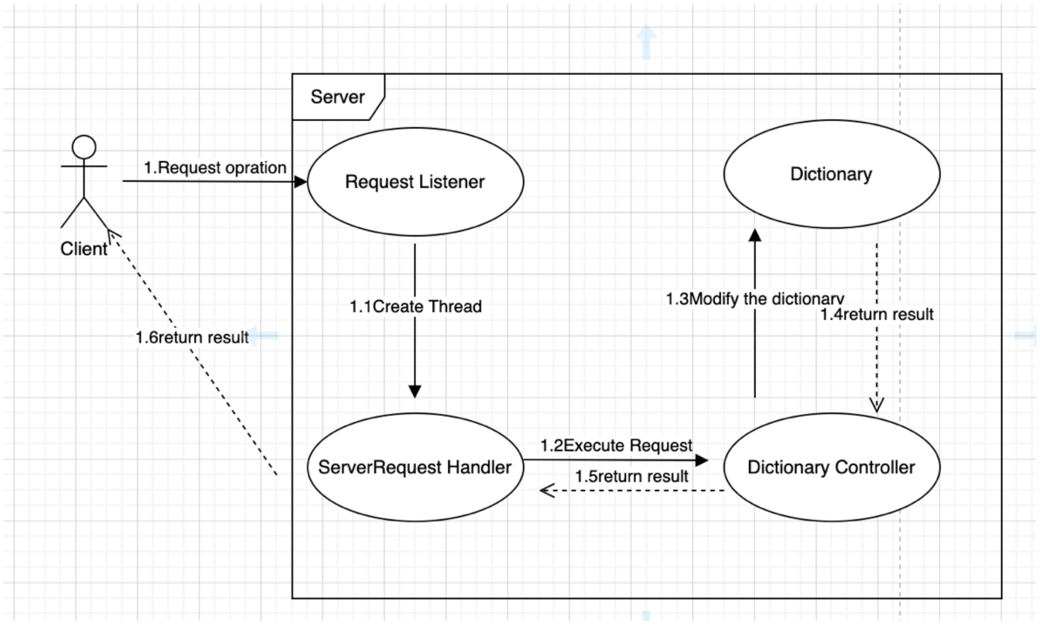


Figure 6: Interaction Diagram

Error Handling

To handle the failure, the system has provided notification to help user to know where it goes wrong.

the table 1 will list all the error and how to handle

Error description	Handling
Input invalid port number	Prevent start-up and ask valid port
The dictionary does not exist	Create specify dictionary
Add a exist word	send word exist to client
Remove a non-exist word	Send word does not exist to client
Query a non-exist word	Send word does not exist to client
Update a non-exist word	Send word does not exist to client

Advantage for the system

For the advantage for my design, the first advantage is the server need fewer thread to handle the same number of users compare to thread-per-connection. It is because the thread is only run when the request is being process, and based on features that users do not frequently do in dictionary applications. So that if many users is in idle, my design will use less resource than thread-per-connection.

Disadvantage for the system

For the disadvantage of my design, the first disadvantage is security problem, it is because everyone can send requests to server and change the content in the dictionary. It means that server can be attacked easily and destroy the dictionary database. The Second disadvantage is resource usage for the thread-per-request architecture. it means it need more resource to create a new thread for each request. So that server needs to spend more time and resource on create a new thread and destroy thread after send the respond. When all users request together, resources may be exhausted.

Creativity

For creativity elements, I create a server GUI for the server admin to check information of server easily and check the request in the system.