Comp90015 Assignment 1 report
Yumin li 1083371

Problem description

The assignment is aimed to perform a dictionary service via the interaction between server and client. The dictionary service should allow multiple clients connect and perform functionality at the same time. The functionality of the program should include searching, adding, updating and delete. The search functionality should take user input and return with the meaning of the given word. If the word that user gave does not exist in the database, inform the user and return to wait for the next user operation. The add functionality should take 2 user inputs, one is the word to add to our database, another one is the meaning of the word that user wishes to add. The user should be informed if the word has been successfully added to the database. The delete functionality takes user input as the word they want to remove from the database and remove it. The user should be informed if the word exists in our database and if it has been successfully removed. The update functionality takes the word that user wants to update and the updated meaning and update it for user. The user should be informed if the word has been found and if the meaning of the word has been successfully updated.
Communication between client(s) and server should follow TCP or UDP. All communication should be taken place via socket. The project should use Java to develop.

Description of component of the system

Server

The server is a multithread server for a dictionary service. It allows multiple clients connected to it and perform add, search and delete functionality. This is a thread per client request server structure. The main class of the server opens up a new server socket to listen to the port which is given by the command line arguments. When a client connects to a server, new socket object has been created and new thread starts to read the client request, processes the request and response it via a “ClientHandler” class.
The ClientHandler class implements runnable interface, it reads the input from the client and split it into command and data. When a search command is received by the server, the client handler will exams the dictionary database txt file for the given word by the user. If the word can be found in the database, the meaning of the word will be returned to the client. If the word cannot be found within the database, the client received a null string. When an add command is received, the data server received should be in the structure of “word/meaning”, the client handler will split by “/” and read the database to see if the word is already in the database. if the word is within the database, returns the error message string “duplicated” to the client. If the word is not found in our database, the client handler will write the word and corresponded meaning of the word into database in the structure of “word:meaning”. When a delete command received, the client handler reads a dictionary file and see if the word exist in our database, if the word has been found, it removes the word and its meaning by creating a new file and copy everything from the file except for the word and corresponded meaning that user wants to delete and returns status of “deleted” to the client. If the word has not been found, return “not found” to the client.

Client

This client class is for a simple client that connects to a server via user specified port and server address. The client connects to the server via port number and service's address provided by the command line arguments. The client is taking user input and put it into the string word entered. The main method creates a socket which gives the connection to the server and communicates with the server by input and output stream. Three commands are given in the code. The search command gives the server a word and got the meaning of the word or error message. The search command gives the server a word and its corresponded and got the status of addition from the server. The delete command gives the server a word and got the status of the deletion.

GUI

The GUI allows user to enter the server address and port they want to connect and connect to the specified server via listener of the connect button which creates a new client. After connection, it can perform search, add, delete and update functionality. The search functionality requires user to type the word they want to search into JTextfield txtword and click search button, the listener of search button will set the word into client and get the meaning or error message from the client. The add button requires user to input the following structure: "word/meaning" and click add button. The status of addition will be displayed in JOptionpane. The delete functionality requires user to type the word they want to delete into JTextfield txtword and click delete button, The status of deleteion will be displayed in JOptionpane. The update feature is combination of delete and add. The status will be displayed in JOptionpane as well. The main method creates the basic frame with set size and make it visible. The default operation is also set in this main function.

Database

The database uses simple structure like "word:meaning" to store the word and meaning. When other class trying to use it or add it. The ".split()" function will separate the word and meaning into 2 different parts for desired purpose.

Overall class design

simpleClient

Attributes

- wordEntered: String (for storeage of incoming word from user)
- received: String (for storeage of returned meaning from client)
- port: int (for storeage of command line arguement of port)
- serverName:(for storeage of command line arguement of Server adress)
- client: Socket (for open up communication with server)
- in: BufferedReader (for reading server message)
- out: BufferedWriter (for sending message to server)
- errorMessage: String (for user to get error message that server return)

Method

- main(): void (set up the client socket for further communication)
- welcomeMessage(): String (only printed if there is no error caught)
- getError(): String (only print the error to user if an error got caught in server)
- setError(): void (for reset the error message to null state for further user input)
- getMeaning(word: String): String (take user input and ask server for meaning of it, error message will be privided if the word cannot be found in database)
- add(word: String): String (take user input and ask server to write it into the database, error messga of duplicated word will be provided.)
- delete(word: String): String (take user input and ask server to delete it from the database, error messga of not found word will be provided.)

MultiHthread Server

Attributes:

Server

- serverSocket: ServerSocket
- clients: List<ClientHandler>

ClientHandler

- socket: Socket
- input: BufferedReader
- output: PrintWriter

Method:

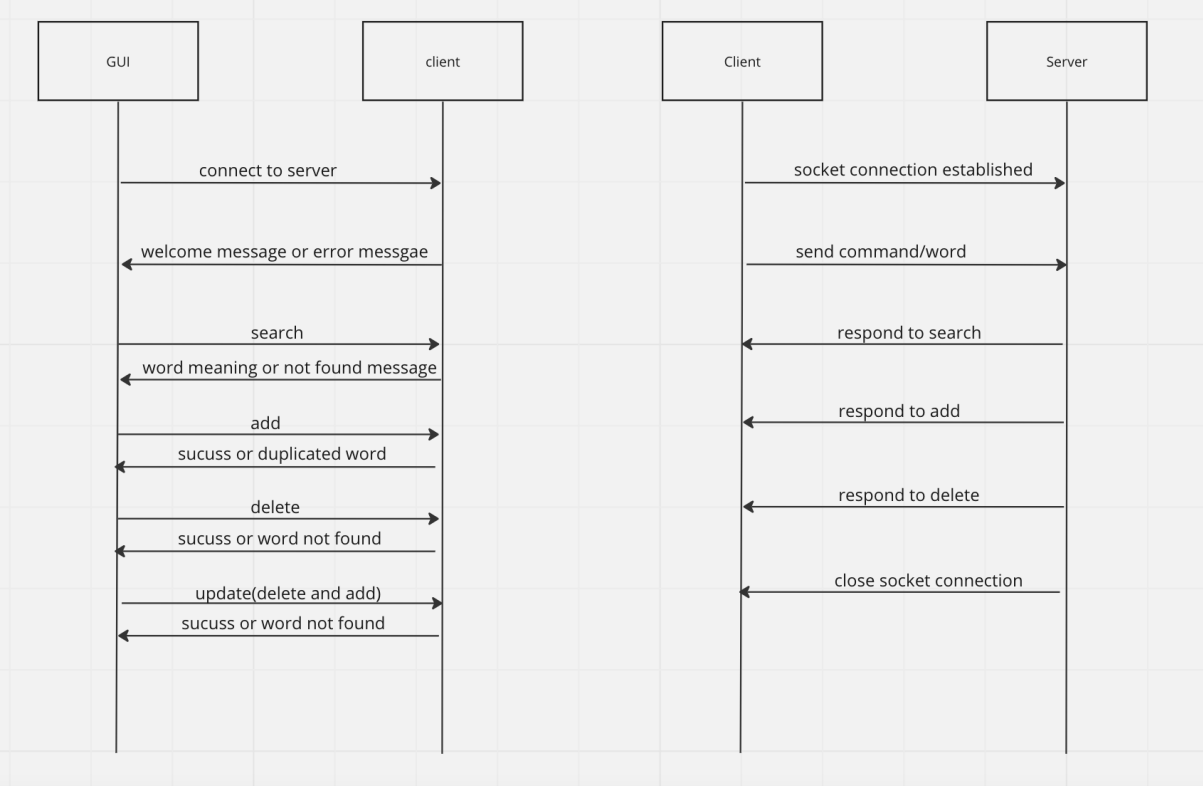
Server

- start(): void (this is use to start the server and listen for client connection)

ClientHandler

- run(): void (start the thread and listen for incoming messgae from the client)
- sendMessage(message: String): void (used to send error message)

Interaction diagram



Critical analysis of work done and conclusion

For the project, this code provides a Dictionary service with server and client architecture and present it with a GUI formation. The server provides currency via multithreading architecture and communicate with the client via TCP. The functionality of search, add, delete and update were fulfilled by the code written. However, the code can be improved by adding more error handling to handle the error that may occur during the program execution. The code can also be improved by using more robust data transfer protocol. The security measurement can also be taking in place. At the moment, any user can access to the database which could be harmful to the system. Overall, the code has fulfilled the requirement descripted in problem description session, but there is still improvement that can be add for the system to be better.