SE 311 Homework Assignment 3

SE 311: Software Architecture II

Overview

In this assignment, you will revise the KWIC system developed in Homework Assignment 2 to create a client-server KWIC search system composed of two components: a KWIC client and a KWIC server.

The primary goal of this assignment is to practice architectural decomposition, component-based design, and concurrent server design, while preserving the functional behavior of the KWIC search feature from previous assignments.

Your implementation must be written in Java and use socket-based communication. The server must support multiple concurrent clients.

Functional Requirements

1. KWIC Client

The KWIC client allows a user to perform keyword search through a client-server interaction. From the user's perspective, the behavior should be the same as the search function in Homework Assignment 1.

- The user enters a keyword from the console.
- The client sends the keyword to the KWIC server.
- If the keyword is found, the system returns all sentences containing the keyword.
    - In the returned sentences, the keyword must be highlighted using brackets.
    - The system reports the total number of sentences containing the keyword.

Example output:

```
1 sentence is found:

Crouching [Tiger] Hidden Dragon
```

If the keyword is not found, the system shall return:

```
[keyword] not found.
```

If the KWIC server does not respond within 30 seconds, the client shall prompt:

```
The KWIC server is not responding.
```

2. KWIC Server

The KWIC server provides search services to one or more KWIC clients.

- The server must support all functions implemented in Homework Assignment 2.
- The server must accept keyword search requests from multiple clients concurrently.
- The server shall be implemented as a multithreaded socket-based server.

The server shall maintain a log.txt file recording:

- The total number of search requests received.
- The number of successful search requests (at least one sentence found).

Submission Guidelines

Submit the following items on BBLearn.

Item 1: UML Component and Class Diagrams

- Submit a PlantUML .md file of your component diagram.
- Do not handwrite diagrams or reverse-engineer them from source code.
- You must include a component diagram with two components: KWIC-Client and KWIC-Server.

** The **KWIC–Client** component should be modeled as a **white-box view**, clearly showing all the classes it contains.

The **KWIC–Server component should be modeled as a **black-box view**, without exposing its internal classes.

- For each component, include a class diagram showing the relationships among its internal classes, that is, there should be 2 class diagrams.
  ** The **UML class diagram for the client side** should be identical to the one included within the **KWIC–Client** component.

  ** The **UML class diagram for the server side** should extend the version you submitted for **HW2**. If you have restructured your HW2 design, please include a brief note explaining the changes made.

- If design patterns are applied, label the pattern name and the role each class plays.

Item 2: UML Sequence Diagram

Submit a PlantUML .md file of your sequence diagrams, demonstrating the following scenario:

The user enters a keyword from the console. The client sends the request to the server. The server searches the sentence repository and returns the matching sentences with the keyword highlighted. If the keyword is not found, return "[keyword] not found." If the server does not respond within 30 seconds, return "The KWIC server is not responding."

Please note that you are required to submit two separate sequence diagrams: one modeling the client component and one modeling the server component.

In total, you should submit **five diagrams**:

1. **One component diagram** showing the relationship between the two components.
2. **Two class diagrams** — one for the client and one for the server.
3. **Two sequence diagrams** — one for the client and one for the server.

Item 3: Source Code (Java, VS Code Projects)

- Submit two Java projects: one for KWIC-Client and one for KWIC-Server, each packaged as **two `.jar` files**
- Use localhost as the server host.
- Submit a single .zip file containing both projects.
- Include a README file explaining how to compile and run both the client and the server.

Item 4: DSMs and Modularity Scores

Submit the .dv8-dsm and .dv8-clsx files for Homework Assignment 3.

For Homework Assignments 1, 2, and 3, submit the Propagation Cost (PC) and Design Level (DL) scores through the provided form. If you are using the DV8 trial version, also submit the M-score.