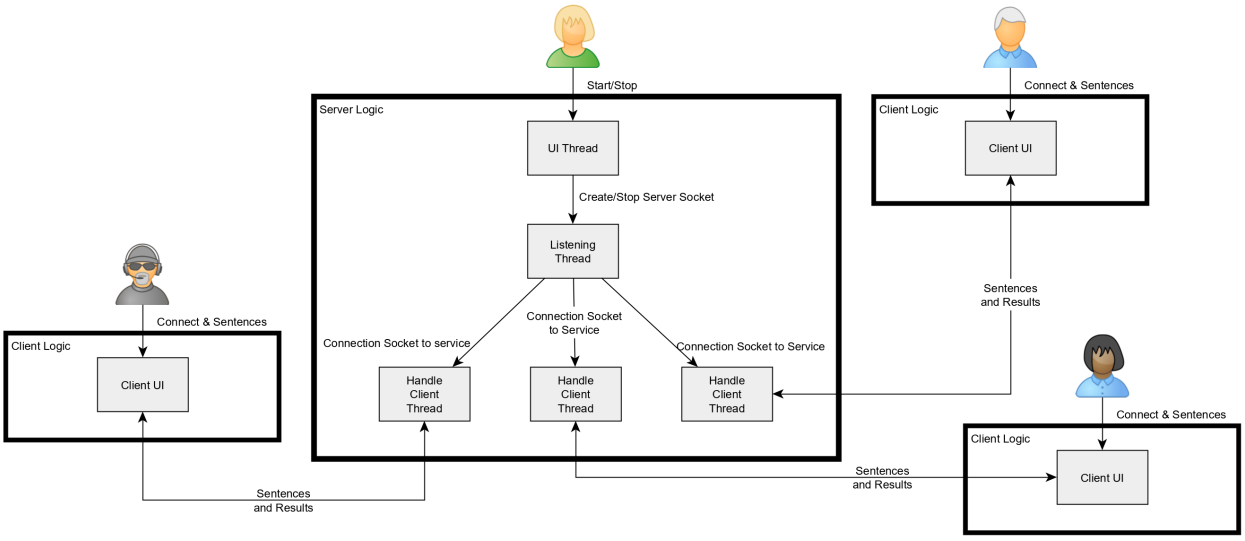
SE424: Distributed Systems	**Recitation 2**
Semester 2 5786	**15 March 2026**
Lecturer: Michael J. May	**Kinneret College**

Sentence Server Improvements: Multithreading and UI/Listening/Client Handling Levels

In this recitation we improve on the sentence server example from last week.

1. For those who did not get their server and client to allow the user to send multiple sentences in a single session, do that.
2. Next, we'll adapt the Java server to handle multiple connections simultaneously using *multithreading*. We'll do that by using a `Thread` class to encapsulate the logic of handling a client.
3. We will then add a third, top level thread to the server to allow the user to stop the listening process by entering 'q' at the server. After entering 'q', the server will stop listening, but be able to resume listening when the user requests it.

The structure of the server and client is sketched out in the following figure.



1 What to do: Client

1. Have the client interface allow the user to enter a sentence, see the result, and then choose to continue entering more sentences.
2. The sentences should all be sent in a single TCP connection session (don't disconnect after each sentence).
3. The user should be able to enter a blank sentence as the signal to stop.
4. When the client has entered the blank line (signal to quit), the client program must close the socket and quit

2 What to do: Server

1. The server must spawn a thread to handle each client session
2. The client session must support the submission of multiple sentences and responses
3. The session should only be closed when the client closes the socket
4. The server must have the following commands:
    - Start listening
    - Stop listening
    - Quit
5. It must be possible to quit only after stopping to listen
6. After stopping to listen, it must be possible to resume (restart) listening again.

