Executive Summary

Assignment Overview:

Implementation of the transport layer protocols TCP and UDP for a storage application. The client and server will try to connect over one particular protocol and communicate the data. For TCP communication, we need to establish the connection from client to server before transmitting the data packets. The connection is a 3 - way handshake protocol. In UDP, the data transfer is done over datagram packets with help of the port as the communication means. TCP provides reliable, connection-oriented communication, while UDP offers a lightweight, connectionless option suitable for scenarios where low latency is more critical than guaranteed data delivery. The project comprises network layer and application layer for both client and server. Where the network layer of the client and server is used to transmit data and pass that down to the application layer. The application layer receives the data from the transport layer and logs the content to the log file.

Technical Overview:

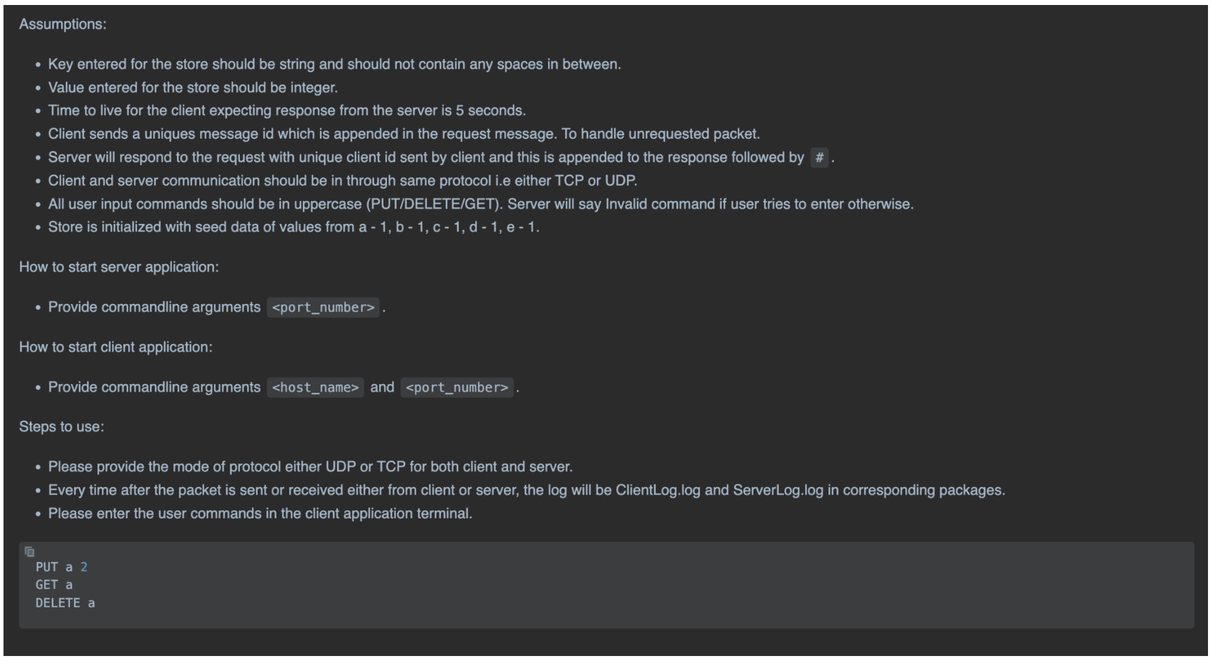
Implementing these protocols helped to comprehend network communication, highlighting the trade-offs between reliability and speed. Integrated the logger for logging communication activities, which helped to maintain a comprehensive record of communication between the client and server. Logging not only aided in debugging but also provided insights into the sequence of events during execution.

Single-threaded server-client implementation processes client requests sequentially, leading to limited concurrency and high response times for multiple clients. This blocking nature prevented simultaneous request handling, affecting overall server responsiveness. The client will be waiting for the response from the server and not able to perform any other operations. Multi-threading assigns each client connection to its thread, enabling concurrent service for multiple clients and simultaneous request processing. This enhances concurrency, leading to faster response times and optimal resource utilisation. Multi-threading allows clients to interact independently, even when one client's operation is ongoing.

Applications:

Real-time Applications: UDP is often preferred for real-time applications where low latency is crucial. This includes online gaming (for transmitting game data), video streaming.
P2P File Sharing: Peer-to-peer (P2P) file-sharing applications, like BitTorrent, use sockets to facilitate direct communication between peers for file sharing
Remote Access: Socket programming is used in remote access protocols like SSH (Secure Shell) and Telnet. These protocols enable users to access remote systems and execute commands on them securely.
Web Servers: Web servers, which handle HTTP requests and responses, often use sockets to listen for incoming connections from web clients (e.g., browsers). These connections typically use TCP.

Read-me



Assumptions:

- Key entered for the store should be string and should not contain any spaces in between.
- Value entered for the store should be integer.
- Time to live for the client expecting response from the server is 5 seconds.
- Client sends a uniques message id which is appended in the request message. To handle unrequested packet.
- Server will respond to the request with unique client id sent by client and this is appended to the response followed by `#`.
- Client and server communication should be in through same protocol i.e either TCP or UDP.
- All user input commands should be in uppercase (PUT/DELETE/GET). Server will say Invalid command if user tries to enter otherwise.
- Store is initialized with seed data of values from a - 1, b - 1, c - 1, d - 1, e - 1.

How to start server application:

- Provide commandline arguments `<port_number>`.

How to start client application:

- Provide commandline arguments `<host_name>` and `<port_number>`.

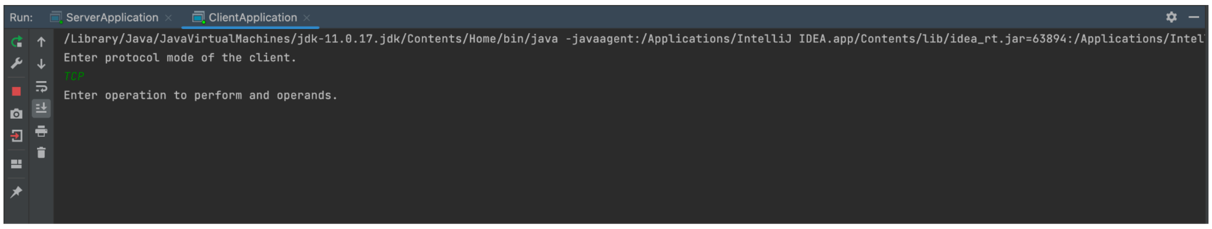
Steps to use:

- Please provide the mode of protocol either UDP or TCP for both client and server.
- Every time after the packet is sent or received either from client or server, the log will be ClientLog.log and ServerLog.log in corresponding packages.
- Please enter the user commands in the client application terminal.

```
PUT a 2
GET a
DELETE a
```

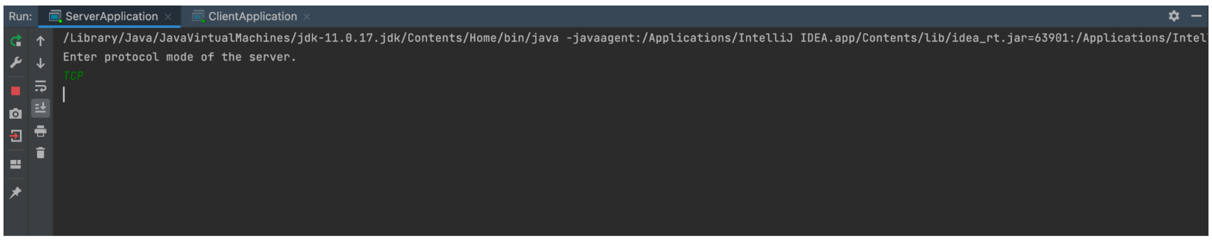
Usage:

Client - Terminal



```
/Library/Java/JavaVirtualMachines/jdk-11.0.17.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=63894:/Applications/Intel
Enter protocol mode of the client.
TCP
Enter operation to perform and operands.
```

Server - Terminal



```
/Library/Java/JavaVirtualMachines/jdk-11.0.17.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=63901:/Applications/Intel
Enter protocol mode of the server.
TCP
```