Protocol

Client/Server Sockets:
Client and Server sockets were generated using SSLSocket object and the Transport Layer Security (TLS) v1.2 protocol. Implementation included RSA encryption, digital signatures and client authentication.

Keys and Certificates:
OpenSSL was used to generate keys and X509 certificates. Certificates are valid for 365 day and private keys were generated with 2048 bit RSA.

Client and Server keys and certificates can be generated using the following command line input (note: appropriate key_file.pem and certificate_file.pem names associated with the respective socket are required):

```
openssl req -newkey rsa:2048 -nodes -keyout key_file.pem -
x509 -days 365 -out certificate_file.pem
```

PKCS12 file format X509 certificates were generated using the following command line input and the associated password:

```
openssl pkcs12 -inkey key_file.pem -in certificate_file.pem -
export -out certificate_file.p12
```

Keystores and Truststores/ KeyManagerFactory and TrustManagerFactory:
KeyManagerFactories used the SunX509 algorithm, whereas the TrustManagerFactories used the PKIX algorithm. Both algorithms were from SunJSSE.

Client UI:
The interface should provide two options to the user: 1. DNA Optimisation and 2. Exit.

DNA Optimisation, option 1, sends the “START DNA” message to the Server and enables the user to enter a DNA sequence once a Server acknowledgement has been received.

The Exit option sends the “DISCONNECT” message to the server, which results in Client and Server socket closure and exit.

Client/Server Interactions:
Client sends the “START DNA” message to Server by selecting menu option 1 to begin the interaction. The Client must wait for the Server to acknowledge the “START DNA” interaction with a “SERVER READY” message prior to allowing the user to enter a DNA sequence to be optimised.

The Client only accepts strings composed of A, T, G or C characters that of a length divisible by 3 or the “SERVER READY” message from the Server. All other strings will

result in sending the “DISCONNECT” message to the server, closure of the socket and exit.

The Server only accepts a string composed of A, T, G or C characters that is a length divisible by 3 or the “START DNA” or “DISCONNECT” messages. All other messages will result in the server sending the Client the “CLOSE SERVER” message which results in the Client and Server socket closure and exit.

The Client can accept lowercase and/or uppercase user input text, whereas the Server only returns uppercase text to the Client.

Valid User Input:
Client only accepts strings of a length divisible by 3 that contains A, T, G or C characters as valid user input. Invalid input will return the user to the Options menu.

Optimisation Rules:
The Server will accept DNA sequences, identify the optimal codons for the sequence based on the optimal codons in Table 1 and generate the optimal DNA sequence.

Table 1. *Optimal Codons for human cell protein expression.*

Amino acid	**Codon**
Ala	GCT
Arg	CGT
Asn	AAC
Asp	GAC
Cys	TGC
Gln	CAG
Glu	GAG
Gly	GGC
His	CAC
Ile	ATT
Leu	CTT
Lys	AAG
Phe	TTC
Pro	CCA
Ser	TCT
Thr	ACT
Tyr	TAC
Val	GTG

Program Execution:
The Client requires a hostname and an integer port number to execute the program, whereas the Server only requires a port number. Missing or excess arguments or invalid arguments will result in an error and prevent execution.