**\*\*\*\*High-level approach\*\*\*\***

The goal of this project is to establish client server communication through socket programming. Familiarizing with python socket programming, we followed object oriented approach to integrate required functionalities. All properties of clients have been declared as variables and behavior of client (send/receive messages, and perform calculations) have been defined as methods.

The client initiates the protocol by creating a TCP socket connection to the server, listening at 27993 port number, and later extended our client such that it supports SSL connections. The project is made responsive to two optional parameters and two mandatory parameters i.e. provided as command line arguments while running project from the terminal.

1. Create an INET, Streaming socket.
2. 2.a For the normal TCP socket : Connect to the given web server on a given port number (default : 27993)
3. 2.b For the SSL socket : Wrap the underlying socket in an SSL context before connecting to the given web server on a given port number (default : 27994).
4. Start communication by sending HELLO message to the server.
5. Receive a response from the server and parse it.
6. If it is a STATUS message, the program is designed to solve mathematical expression based on the operator, and result will be sent back to the server Go to step 4.
7. If it is a BYE message, the program prints the secret flag, and exit the program.

**\*\*\*\*Challenges faced\*\*\*\***

1. We developed sample chat application to get acquainted with socket programming in Python

2. We started developing this project in python 3.4, while testing on server we realized that it does not support python34, we migrated our projected to python 2.7. Messages to sent are required to be encoded with 'UTF-8' in Python 3.4 and hence required to be decoded while receiving from the server whereas this encoding/decoding was not required in Python2.7

3. We faced problem running Python project from command line, and resolved this issue by setting path for environment variables to python.exe file of our respective machines.

4. We also faced issue while bypassing certificate for SSL socket. CERT\_NONE, CERT\_OPTIONAL or CERT\_REQUIRED options of cert\_reqs field are no more in use, instead integer values 0,1 and 2 are used to configure.

5. Validating optional command line arguments required to run application, was challenge keeping number of conditions to be parsed as least as possible.

**\*\*\*\*Overview of how we tested your code\*\*\*\***

1. We started with the base case to test this project (default port number 27993)

$ ./client servername.ccs.neu.edu 001753883

Expected Behavior: Prints a unique secret flag terminating connection successfully.

Output : Prints a unique secret flag.

2. We used unauthorized NUID to test if our program prints "Unknown\_Husky\_ID"

$ ./client servername.ccs.neu.edu 00175388300

Expected Behavior: prints "Unknown\_Husky\_ID" string terminating connection successfully.

Output : prints "Unknown\_Husky\_ID" string

3. We provided wrong server name to test if our program terminates with an error message.

$ ./client servername 00175388300

Expected Behavior : terminates the program and prints an error message.

Output : "Error occured!! Please provide correct parameters and try again!!"

4. We tested base case for SSL connecttion (default port number 27994)

$ ./client -s servername.ccs.neu.edu 001753883

Expected Behavior: Prints a unique secret flag terminating connection successfully.

Output : Prints a unique secret flag.

5. We tested this project overridding the default port number for SSL connecttion

$ ./client -p 27995 -s servername.ccs.neu.edu 001753883

Expected Behavior: terminates the program and prints an error message.

Output : "Error occured!! Please provide correct parameters and try again!!"

6. We tested this project overridding the default port number with same port number for SSL connecttion

$ ./client -p 27994 -s servername.ccs.neu.edu 001753883

Expected Behavior: Prints a unique secret flag terminating connection successfully.

Output : Prints a unique secret flag.

7. We performed negative testing ensuring that our application can handle invalid input or unexpected user behavior.

$ ./client 27994 -p -s servername.ccs.neu.edu 001753883

Expected Behavior: Prints error message

Output : "Error occured!! Please provide correct parameters and try again!!"