[GUIDELINE before opening the project]

1) In order to run this project, you need to open two command prompts:
2) For the first command prompt, write the following:
   - [javac Server.java] - to compile the Server class
   - [java Server] - to run the Server app
3) The same for the second prompt, but with corresponding name:
   - [javac Client.java] - to compile the Client class
   - [java Client] - to run the Client app
4) Now you can start typing messages in Client(Alice) window, and it will show up in the Server(Bob) window

[REPORT]

To start with, we have implemented the one-way Client and Server setup where a Client (in our case Alice) connects, send messages to the Server (Bob), and the Server in turn shows this message on his command prompt (screen) using socket connection. Frankly, there is a lot of low-level stuff that needs to happen for these things to work but the Java API networking package (java.net) takes care of all of that, making network programming very easy for us.

Regarding the encryption part, we decided to use already existing cryptographic tools instead of making all that work from scratch (javax.crypto, java.security). Generally, we have used *RSA, SHA-1* and *AES-128*. Lets discuss each of them separately.

- RSA

Server (Bob) generated pair of keys, then sends public key to the Client (Alice), while she in turn is going to use that key for further secret key encryption and sending back to Server (Bob). Thus, the server has Clients secret key.

- SHA-1

We know that AES-128 requires 128-bit key, that is why we have used this hash algorithm to convert our secret key into value with a fixed size,128-bit in our case.

- AES-128

This algorithm is used for messaging between Client (Alice) and Server (Bob). Classes that were used to implement this algorithm are: Cipher, SecreyKeySpec, etc.