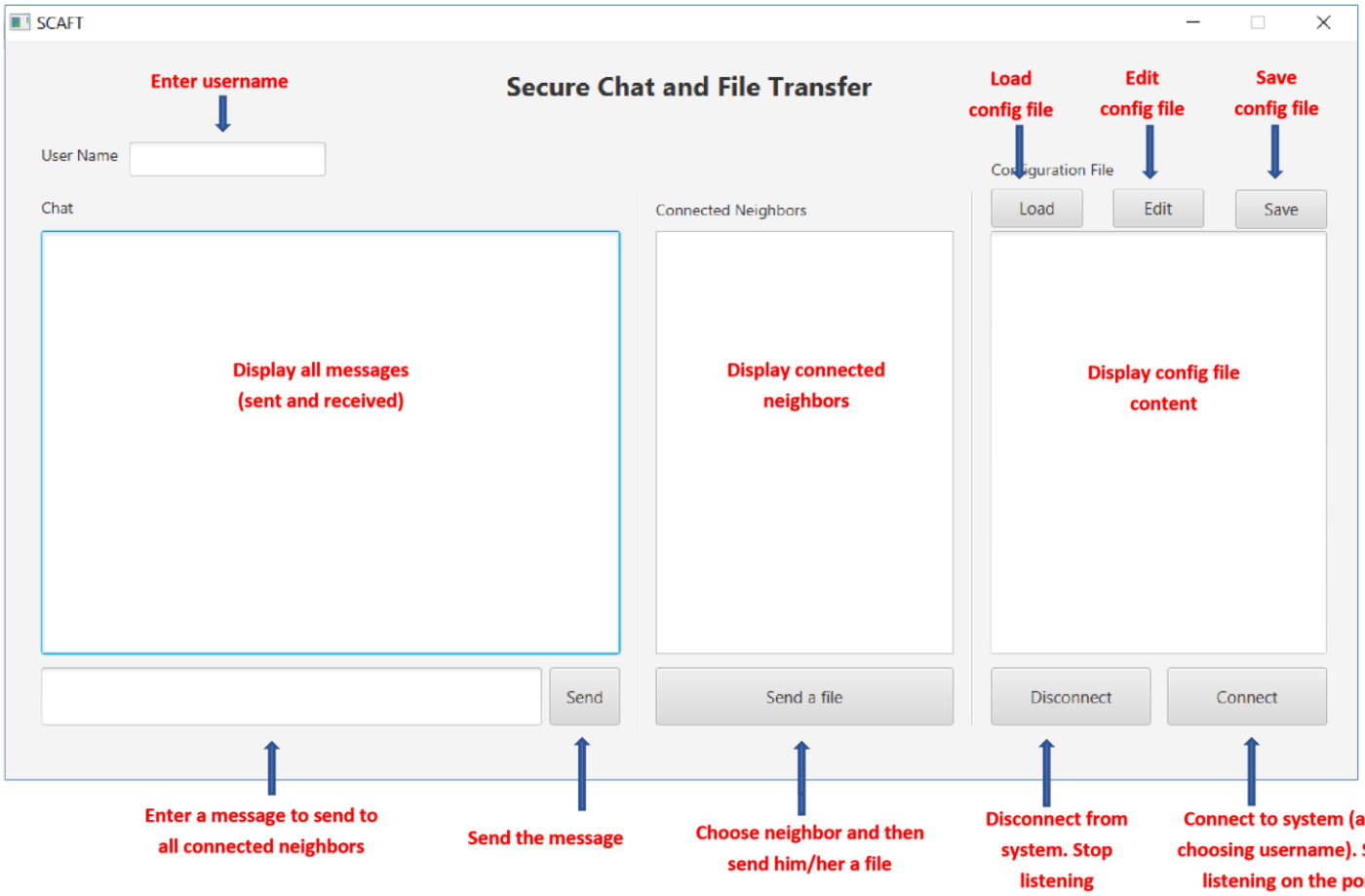
**Instructions for compiling the source code for SCAFT:**

1. Open SCAFT project using IntelliJ.

2. Press on the Run icon (green triangle).

**Instructions for running the SCAFT programs:**




**Documentation of the communication protocol:**

**Message formats:**

1. HELLO message - HELLO Username IP Port

2. BYE message - BYE Username IP Port

3. OK message - OK Username IP Port MessageContent

4. NO message - NO Username IP Port MessageContent

5. MESSAGE message - MESSAGE Username IP Port MessageContent

6. SENDFILE message - SENDFILE Username IP Port FileName

Finally: we encrypt the message and send it with the corresponding IV.

For example: Encrypt(HELLO Username IP Port)-IV

4a28b97db2e4-fa45d46bafad4cd46e5b4a28b97db2e4

**Encrypt/Decrypt messages explanation:**

**Encryption:**

1. Generate key – get password from config file, encode it using UTF8 encoding and then hash it with SHA256 algorithm.
2. Generate random IV – each sending we create new random IV (128 bits).
3. Encryption using AES/CTR.

**Decryption:**

1. Parse the encrypted incoming message by splitting it by "-" character. First section is the message and the second is the IV.
2. Generate key – get password from config file, encode it using UTF8 encoding and then hash it with SHA256 algorithm.
3. Decryption using AES/CTR.

**How SCAFT parses the messages:**

The SCAFT tool listens to incoming messages on the port from the config file. When received a new message, SCAFT decrypt the message.
SCAFT parse the decrypted output by splitting it by " "(space) character.
SCAFT takes the information from the parsed message according to the formats we described above.

**How the SCAFT tool reacts to the different messages:**

1. HELLO message – if the sender of the message isn't in the connected neighbors list add the sender to the connected neighbors list, and send HELLO message back. Displays an appropriate message
2. BYE message – delete the sender of the message from the connected neighbors list. Displays an appropriate message.
3. OK message - displays a message that the receiver accepted the file.
4. NO message - displays a message that the receiver refused to accept the file.
5. MESSAGE message – display the message content in the chat.
6. SENDFILE message – display a message to the user that someone wants to send a file to him, and wait for the user to answer. According to the user answer – send back OK or NO message.

**File Transfer Explanation:**

After sending a SENDFILE message and receive OK message back:

- Encrypt the file:
    - Generate key and random IV (see Encrypt/Decrypt messages explanation).
    - Encode the file name using UTF8 encoding and encrypt it.
    - Read all file bytes, encode the bytes using BASE64 encoding and encrypt them.
- Encrypted file looks something like this:
  Encrypted(ip:port)-Encrypted(file name)- Encrypted(file content)-IV

- After encrypting, we set up a new socket with another port and sending the encrypted file.
- The receiver listens to the same port, receive the encrypted file and decrypt it in a similar way.
- SCAFT ask the user to choose location to save the file – with the original name.
- If a file with the same name already exists, SCAFT ask the user if he/she wants to overwrite:
    - If the user says yes – overwrite, and save the new file in the selected location. ○ If the user says no – abort.