Security of Computer Systems

Project Report

Authors:
Michał, Błajet, 180564

Version: 1.0

Versions

Version	Date	Description of changes
1.0	27.04.2022	Creation of the document
1.1	…	…

1. Project – control term

1.1 Description

Version of application delivered to control term contains following functionalities:

- contact list
- creating new connection
- responding to incoming connection
- chatting as a part of connection (text only)
- settings defined by properties file

This version does not support encryption nor key exchange mechanism. Backend side of the application supports multiple connections and message exchanging, but such functionality is not fully implemented in the user interface.

Instance of app by default runs on port 8080, but it can be overridden by setting suitable *port* property in *schat-application.properties* file or by passing it as a program argument.

Application is made in Java language in version 17. To create a graphical user interface the JavaFX has been used.

Application is divided into two parts: view and backend. The view is responsible for displaying and receiving information from the user and passing them to the backend part. Part of a view are controllers that communicate with each other and use the backend part to perform operations such as creating new connections, obtaining a list of available contacts or adding a new one.

Backend part is the main part of the application and there all business operations are performed. This package shares some of the classes to the gui package as an api. Some of the components are stored and managed by a container, from where they can be obtained. Thanks to that, dependency management is much simpler and straightforward.

Messages are sent as objects which contain type of message, byte array of message content (text of file) and index of message. In the future version encrypted will be only content of message and author, if added. Index of message is used to handle files and text messages that are too big to be sent in one message. Type of message

has been introduced as a preparation for encryption and key exchange functionality. Available message types are:

- secret key
- public key
- text
- file
- invitation

Invitation type will be used at the beginning of the communication to allow accepting or rejecting connection before sharing public key of host to which connection has been created.

MessageHadlers objects are responsible for dividing text (or file) into smaller pieces and passing them to the sender object and for receiving messages from the receiver object. These objects are also responsible for mapping message objects into gui components as text or file (and vice versa) and also for encrypting/decrypting messages before passing them further.

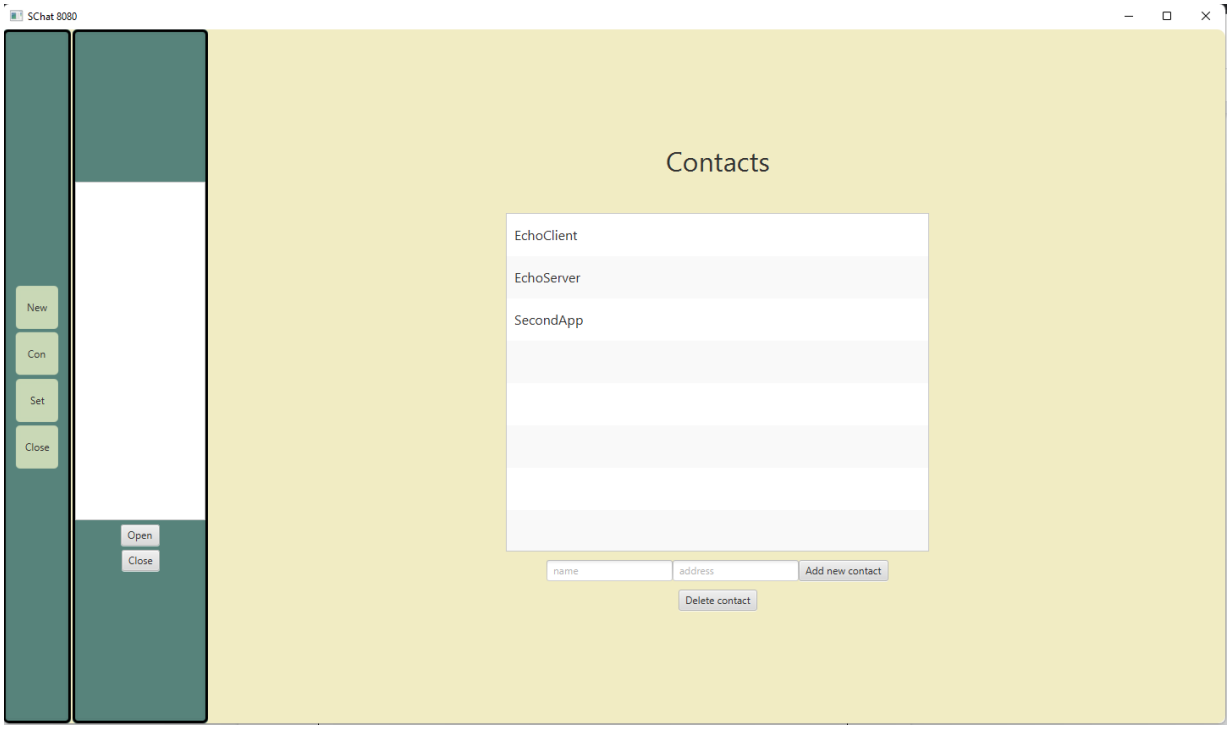
Another important component of the application is server object which is responsible for listening for connections, creating suitable connection objects using connection service component and notifying listeners. In the delivered version, listeners are expected to be gui controllers.

Contacts are defined as a name and ip address with port, on which application will be listening. Users cannot add new contact if name or address already exist in the repository.

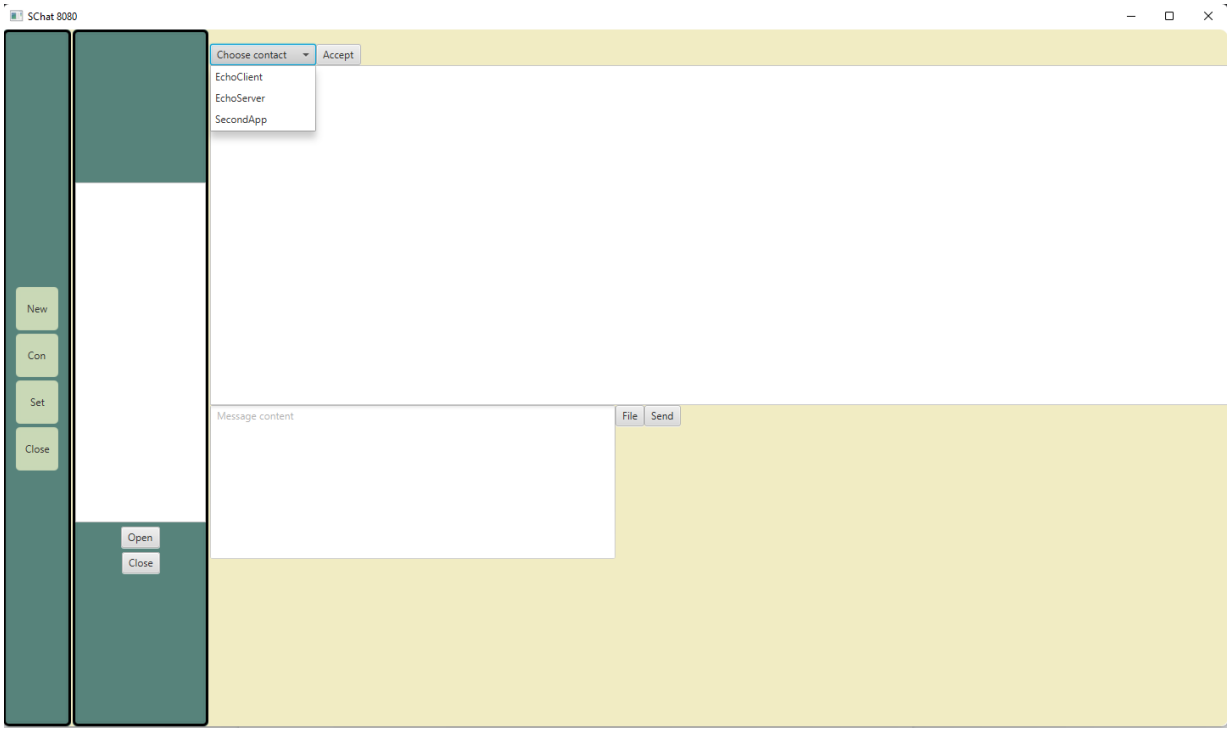
1.2 Results

Following pictures contain views of application instances, their functionalities and sample chat between two instances. These instances are runned on the same machine.

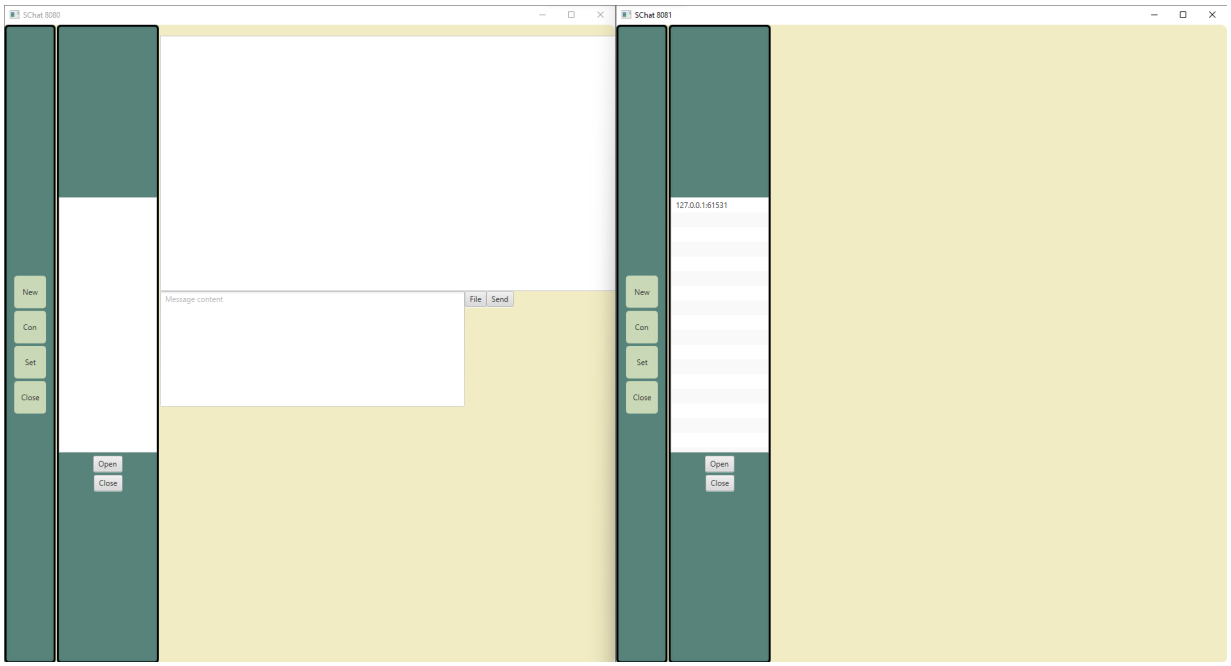
View after launching application



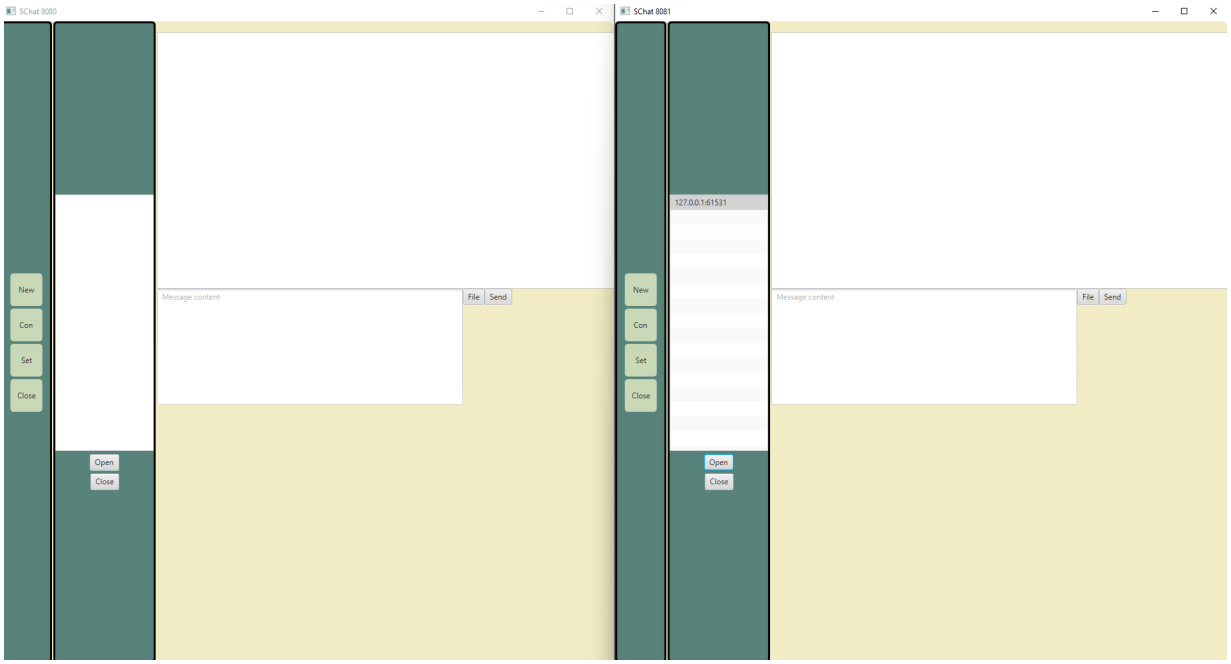
View of contacts tab



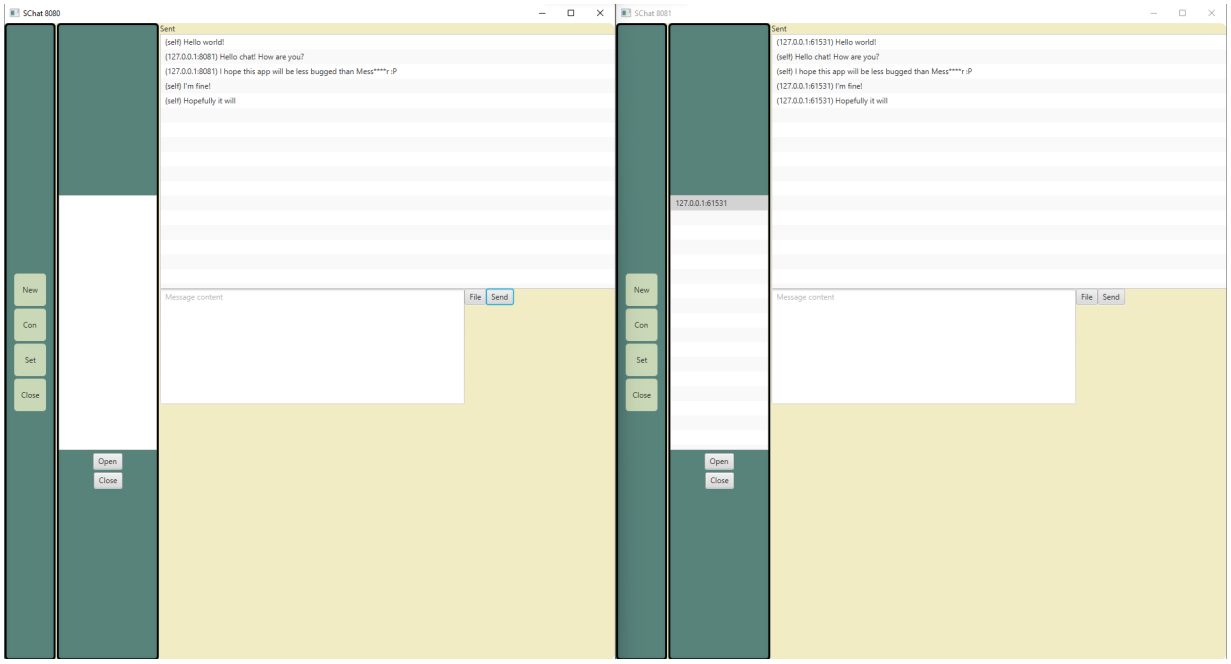
View of new conversation tab with expanded list of contacts



Two instances, left one after selecting contact associated with the right one instance. Right instance has new element in list of incoming conversation, that can be accepted or rejected using one of the button below the list



After accepting incoming connection the conversation view has been displayed (right window)



View of conversation between two instances

1.3 Summary

App allows communication between two users and is prepared to handle multiple connections at the same time. There is a need to adjust handling errors and displaying suitable status of connection and state of the app as well. User interface will be improved to be more readable. In the future version settings view will be added to manage settings directly from the app. Another improvement planned in the next version is encryption/decryption mechanism for messages, sending files and multiple conversations handling with switching between them without cancelling the previous one.

2. Project – Final term

2.1 Description

Content

2.2 Description

Content

2.3 Description

Content

2.4 Results

Content

2.5 Summary

Content

3. Literature

[1] Article.

[2] Website, (access date).

[3] Book.