

# Dokumentacja Projektu

## Komunikator internetowy typu IRC

Sieci komputerowe - ćwiczenia projektowe

### Cel projektu:

Celem projektu było stworzenie komunikatora internetowego typu IRC w oparciu o architekturę Klient-Serwer. Zadaniem programu jest umożliwienie wymiany wiadomości pomiędzy podłączonymi do serwera użytkownikami.

### Środowisko:

Projekt został zrealizowany w języku Java w środowisku Eclipse Java Neon. Natomiast Graficzny Interfejs Użytkownika GUI został wykonany za pomocą programu Scene Builder i plików .fxml.

### W skład projektu wchodzi:

#### Projekt CommunicatorServer:

- klasa Server – odpowiada za połączenie sieciowe,
- klasa ClientThread – w celu sprawnej obsługi klientów serwer przydziela każdemu użytkownikowi osobny wątek,
- ServerController – kontroler pliku Server.fxml, który odpowiada za prawidłowe funkcjonowanie po naciśnięciu odpowiednich przycisków w oknie,
- Server.fxml – plik fxml, przygotowane okno serwera z odpowiednimi przyciskami umożliwia jego uruchomienie i zatrzymanie.

-Klasa Main – umożliwia uruchomienie projektu, następuje pojawienie się okna serwera.

#### Projekt CommunicatorClient:

-klasa Client – odpowiada za połączenie sieciowe,

-klasa ClientController – kontroler pliku Chat.fxml, czyli głównego okna aplikacji, służy do obsługi Graficznego Interfejsu Użytkownika

-Chat.fxml – plik fxml, główne okno aplikacji z polami tekstowymi, obszarem tekstowym i przyciskami, umożliwiającymi wygodne i proste w obsłudze korzystanie z komunikatora,

-Klasa Main – umożliwia uruchomienie projektu, powoduje pojawienie się głównego okna aplikacji.

W celu uruchomienia komunikatora najpierw należy uruchomić serwer a następnie można przystąpić do podłączenia klientów.

#### Opis działania programu:

-po włączeniu aplikacji serwera wyświetla się okno z dwoma przyciskami, umożliwiającymi uruchomienie serwera oraz zakończenie jego pracy, odpowiednio „Start” i „Stop”,

-początkowo aktywny jest tylko przycisk „Start”, którego naciśnięcie powoduje stworzenie wątku serwera i uruchomienie go,

-po uruchomieniu serwera przyciskiem „Start” można przystąpić do podłączania klientów,

-po uruchomieniu aplikacji klienta wyświetlone zostaje okno główne z aktywnym polem tekstowym do wprowadzenia nazwy użytkownika i adresu IP serwera,

-aby połączyć się z serwerem użytkownik musi podać swoją nazwę oraz adres IP,

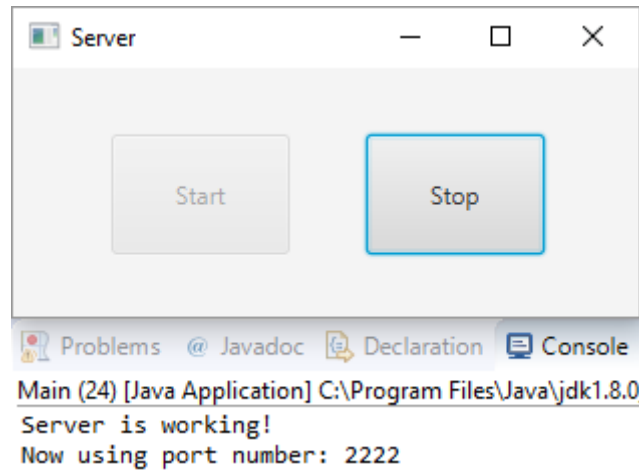
-po poprawnym wprowadzeniu danych i kliknięciu przycisku „Connect” zostaje nawiązane połączenie z serwerem,

-w głównym obszarze tekstowym pojawia się komunikat powitania nowego użytkownika oraz aktywne stają się pozostałe przyciski i pola,

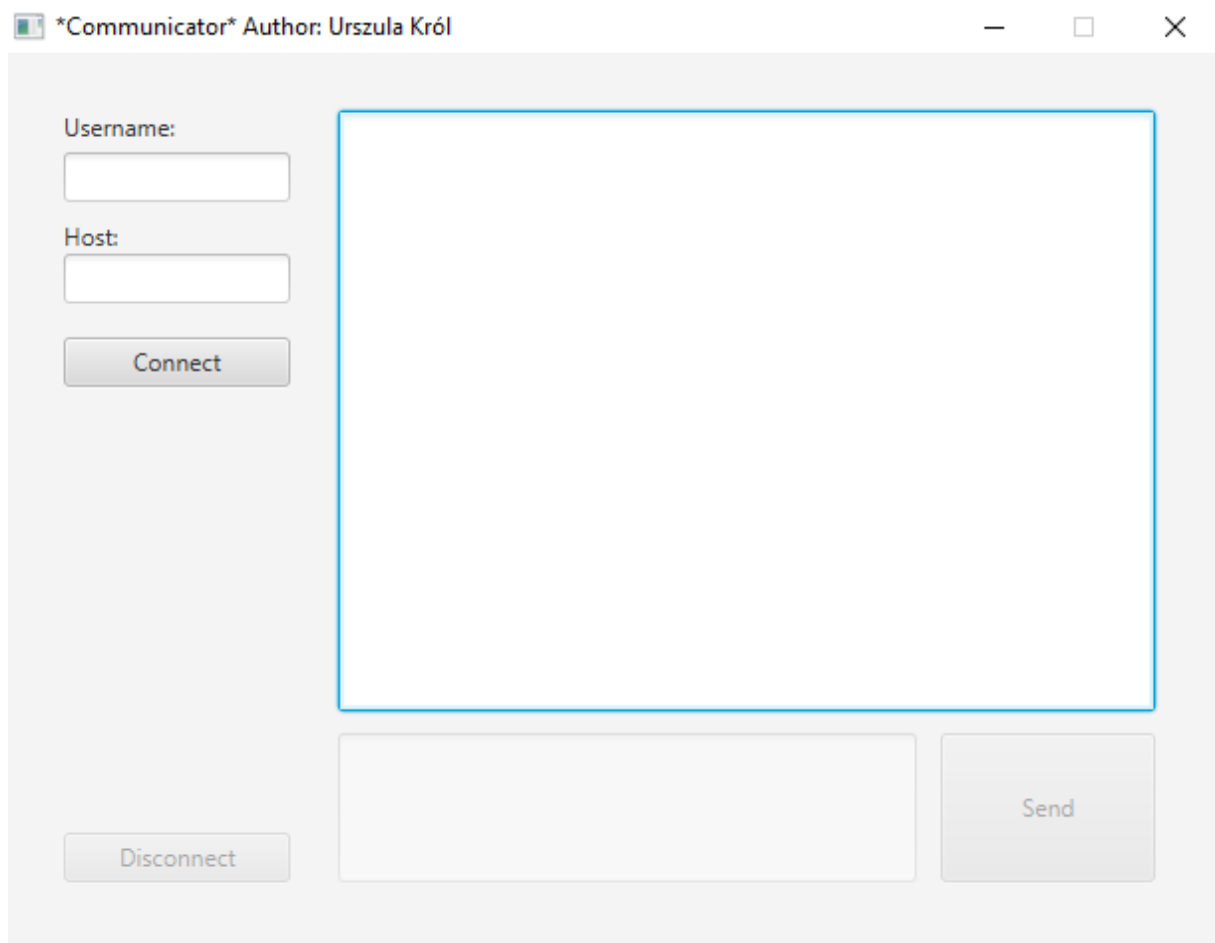
-kiedy do konwersacji dołączy inny użytkownik każdy z obecnych otrzymuje o tym stosowny komunikat,

- pole tekstowe służy do wpisywania wiadomości, która ma zostać wysłana,
- po wpisaniu wiadomości i kliknięciu przycisku „Send” wiadomość zostaje wysłana do serwera, który następnie rozsyła ją do reszty podłączonych użytkowników,
- naciśnięcie przycisku „Disconnect” powoduje zakończenie nawiązanego połączenia, na strumień wysyłana jest wiadomość „disconnected”, która po odczytaniu zostaje rozpoznana i powoduje, że rozpoczęty zostaje proces zakończenia połączenia, co objawia się komunikatem o opuszczeniu konwersacji w obszarze tekstowym wszystkich pozostałych użytkowników.

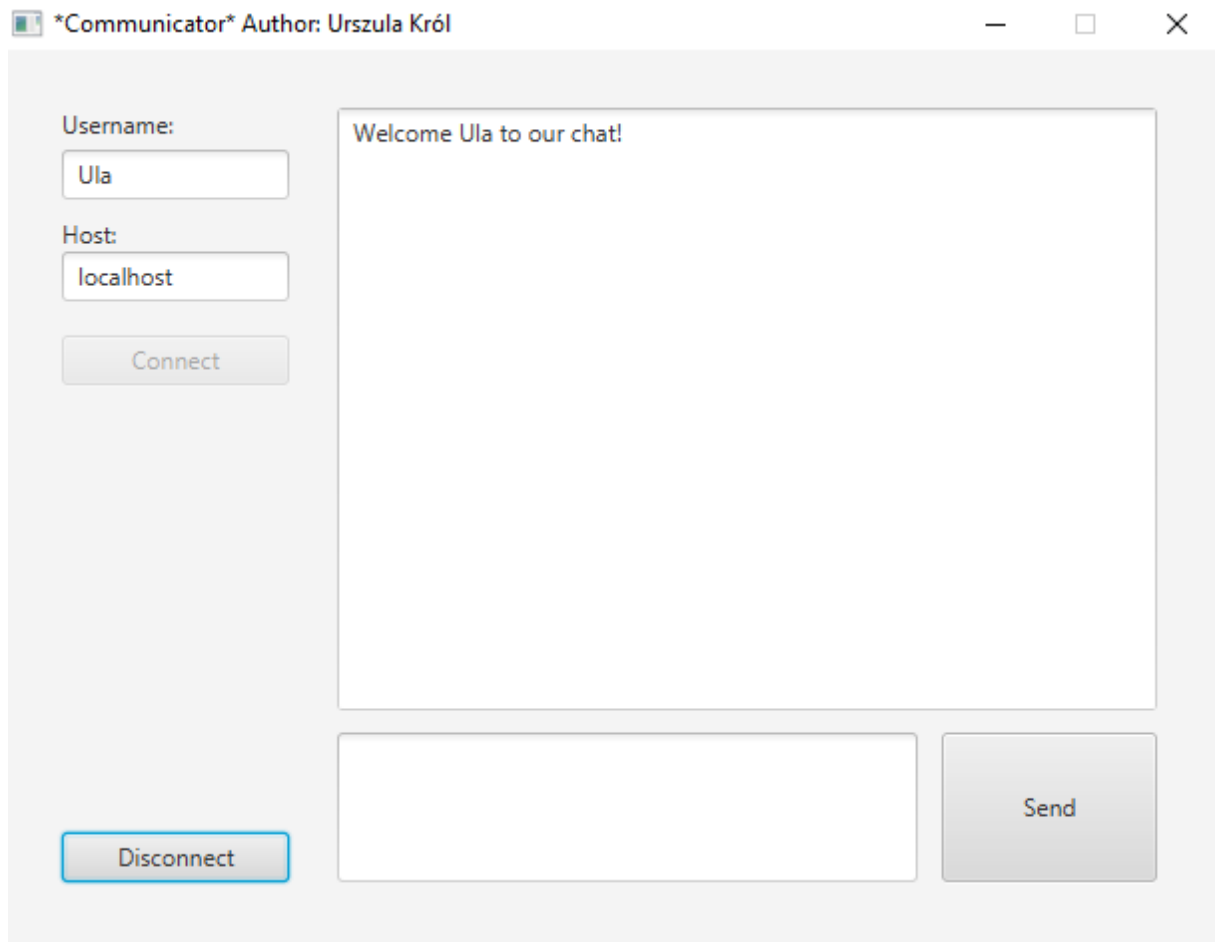
### Zrzuty ekranu z działania programu:



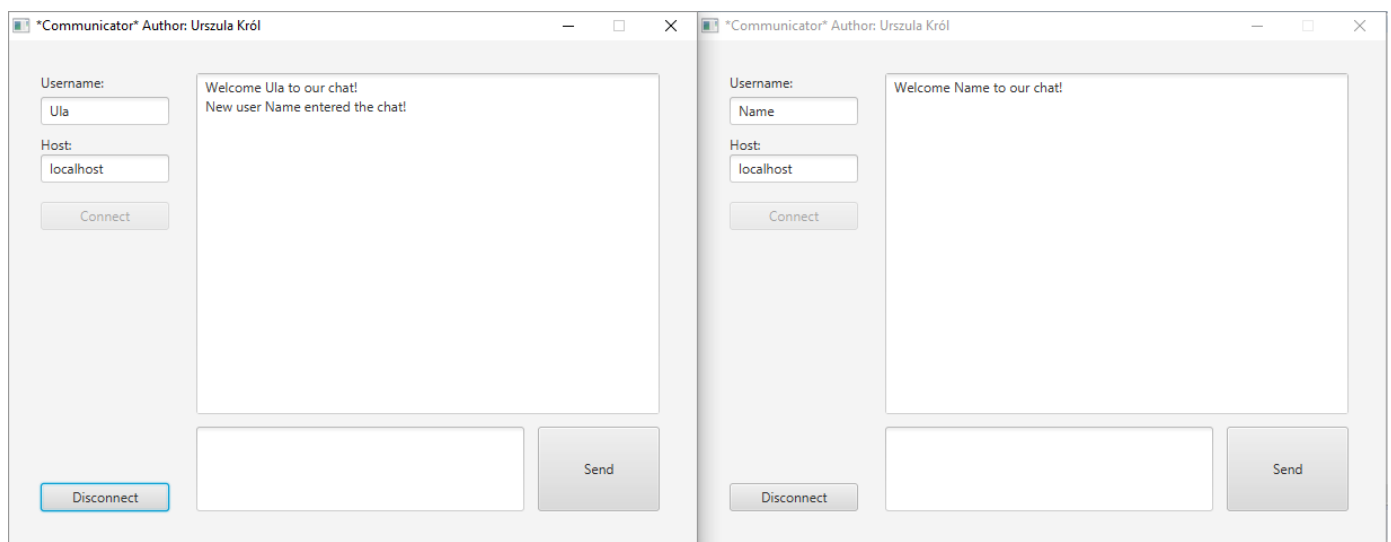
1. Po uruchomieniu serwera przyciskiem „Start”



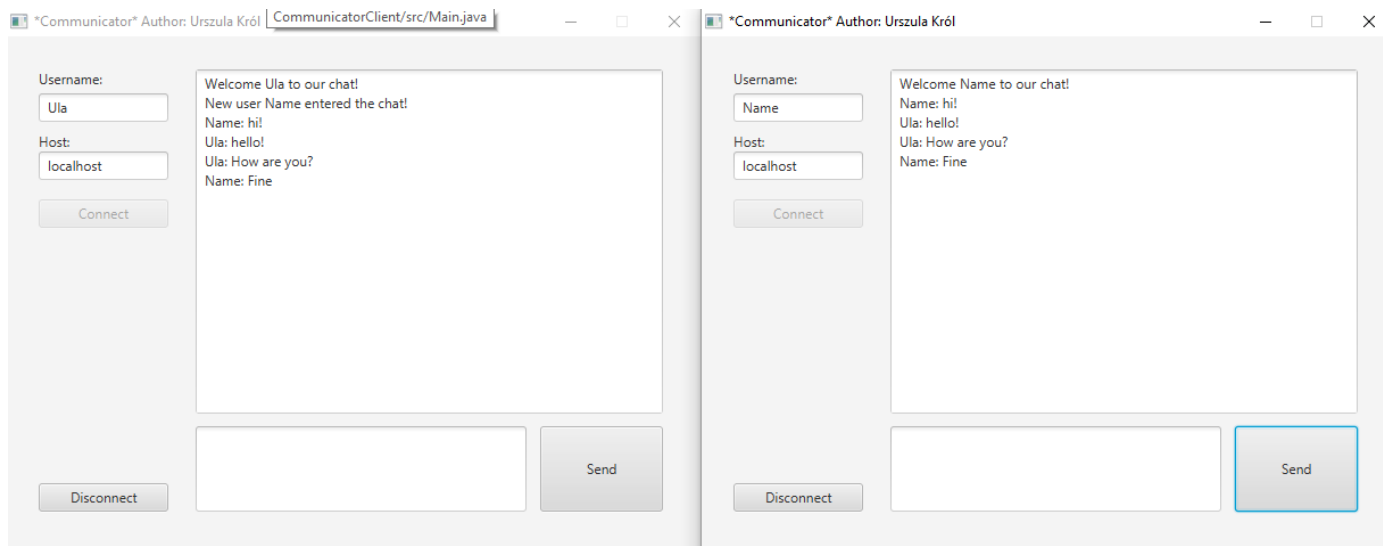
2. Głównie okno aplikacji



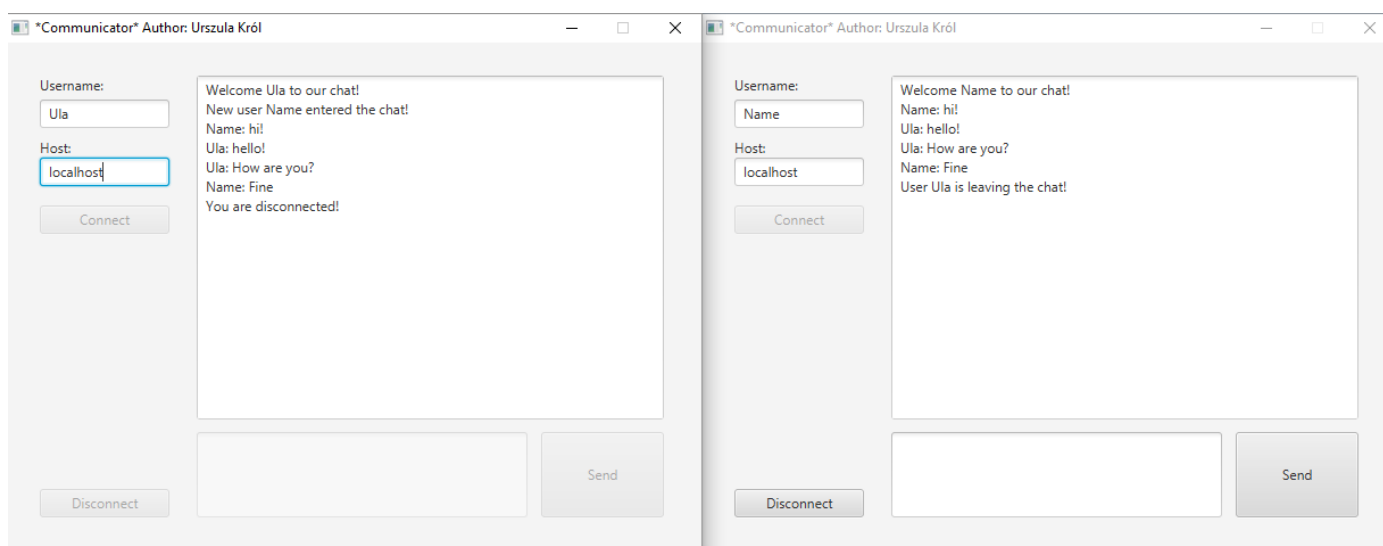
3. Po wpisaniu nazwy użytkownika i hosta



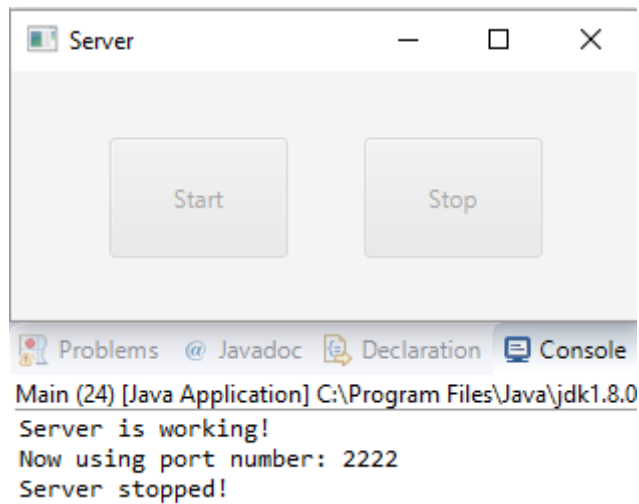
4. Po podłączeniu drugiego klienta



## 5.Wymiana wiadomości między połączonymi użytkownikami



## 6.Po rozłączeniu jednego użytkownika



7. Po zatrzymaniu serwera przyciskiem „Stop”

## Podsumowanie

Projekt komunikatora został wykonany zgodnie z architekturą Klient-Serwer w oparciu o sockety. Praca nad projektem umożliwiła zapoznanie się z modelem Klient-Serwer oraz dała możliwość zrozumienia tej komunikacji. Wykonanie aplikacji dostarczyło nowej wiedzy i rozwinęło umiejętności programistyczne. Wykonany komunikator działa poprawnie. Umożliwia wymianę wiadomości pomiędzy podłączonymi do serwera użytkownikami w prosty i wygodny sposób.