**PROJEKT\_CHAT1\_SERVER – DOKUMENTACJA**

Autor: Krzysztof Rudzki

Opis serwera chat, który łączy klientów chat do wspólnej wymiany informacji.

Repozytorium: <https://github.com/krzysieker/PROJEKT_CHAT1_SERVER>

Spis treści

**Wpisz tytuł rozdziału (poziom 1)1**

Wpisz tytuł rozdziału (poziom 2)2

Wpisz tytuł rozdziału (poziom 3)3

**Wpisz tytuł rozdziału (poziom 1)4**

Wpisz tytuł rozdziału (poziom 2)5

Wpisz tytuł rozdziału (poziom 3)6

1. **UŻYTY PROTOKÓŁ**

Do połączenia użyto protokołu TCP na socketach.

Obraz zawierający wewnątrz

Opis wygenerowany przy bardzo wysokim poziomie pewności

Maksymalna liczba klientów: 100

*socketServer.Listen(100);*

1. **OKNO APLIKACJI**

Po uruchomieniu serwera pojawi się główne okno aplikacji. Z racji iż jest to wersja 1.0 to jest to jedyne okno dostępne na ten czas. W przyszłości można rozbudowywać projekt np. dodając formularze logowania czy zaawansowane zarządzanie klientami.

Okno po uruchomieniu wygląda następująco:

Obraz zawierający zrzut ekranu

Opis wygenerowany przy bardzo wysokim poziomie pewności

**TextBox: Adres IP:** służy do wprowadzenia adresu IP naszego serwera. Domyślnie ustawiono wartość 127.0.0.1, gdyż klienci byli uruchamiani na tym samym urządzeniu co serwer. W przypadku lokalnej sieci można ustawić IP serwera w tej sieci.

**TextBox:** **Port:** analogicznie do adres IP. Domyślnie zarówno na serwerze jak i kliencie ustawiono jego wartość na 5000,

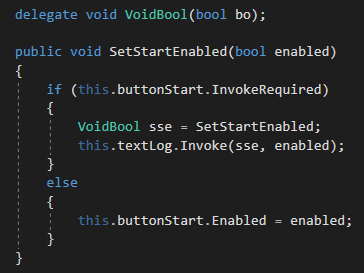
**Button: Uruchom serwer**: tak jak nazwa wskazuje służy do uruchomienia serwera.

Po uruchomieniu serwera button : Uruchom serwer zostaje zablokowany.

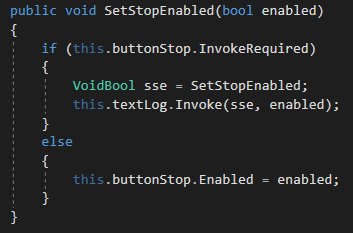
Obraz zawierający zrzut ekranu

Opis wygenerowany przy bardzo wysokim poziomie pewności

Kod programu odpowiadający za tą funkcję. Ponieważ funkcje głównego formularza wywoływane są w głównej klasie Program.cs, użyto delegatów.



**Button: Wyłącz serwer:** tak jak nazwa wskazuje służy do uruchomienia serwera. Analogicznie jak w przypadku „Uruchom serwer”



**ListBox (pod buttonami):** Wyświetla IPEndPoint (adres IP + port) aktualnie połączonych z serwerem klientów.

Dla przykładu:



**TextBox: Logs** (największy): Wyświetla wszelkie komunikaty, wiadomości,

Obraz zawierający zrzut ekranu, wewnątrz

Opis wygenerowany przy bardzo wysokim poziomie pewności

**TextBox: Message** (na dole): Służy do wprowadzania tekstu do wysłania.

\*Ograniczono liczbę znaków do 1024, gdyż taki bufor danych został zadeklarowany.

**Button: Wyślij wszystkim:** Służy do wysłania wiadomości z textBox: Message do wszystkich podłączonych klientów.

1. POŁĄCZENIE

Tak jak wspomniano przy omawianiu protokołu użyto TCP na socketach.

SZYFROWANIE WIADOMOŚCI

ODBIERANIE WIAODMOŚCI I PRZEKAZYWANIE DO INNYCH KLIENTÓW

WYSYŁANIE WIADOMOŚCI

ZAPISYWANIE LOGÓW I WYŚWIETLANIE WIADOMOSCI