

# Serwer Echo –analiza wymagań

Anna Foltyniewicz  
Łukasz Jakubowski  
Marta Elik  
Eryk Lewandowski

<https://github.com/Ongakute/SerwerTCP>

## Wprowadzenie

W tym dokumencie zostaną omówione wymagania funkcjonalne oraz pozafunkcjonalne asynchronicznego serwera szyfrującego. Głównym zadaniem serwera jest odsyłanie zaszyfrowanej wiadomości, którą serwer otrzyma od klienta lub jej deszyfracja za pomocą znanego klucza. Jednak aby tego dokonać klient najpierw musi zalogować się do serwera. Serwer jest w stanie obsługiwać kilku klientów na raz. Implementacja wszystkich aktywności serwera jest uproszczona do absolutnego minimum (baza użytkowników przechowywana jest w zmiennej typu Dictionary).

## Wymagania funkcjonalne

1. Administrator uruchamia aplikację serwera.
2. Serwer rozpoczyna działanie od razu po uruchomieniu aplikacji.
3. Administrator kończy działanie aplikacji serwera.
4. Klient łączy się z aplikacją serwera za pomocą aplikacji klienckiej.
5. Klient loguje się do serwera za pomocą loginu hasła.
6. Po zalogowaniu klient wybiera tryb pracy serwera – szyfrowanie lub deszyfrowanie wiadomości za pomocą podanego słowa klucza.
7. Klient kończy połączenie z aplikacją serwera.
8. Serwer łączy się z bazą danych
9. Klient może zmienić hasło do logowania

## Wymagania pozafunkcjonalne

1. Aplikacja serwera jest dostarczona w postaci aplikacji konsolowej przeznaczonej na system Windows.
2. Aplikacja kliencka jest dostarczona w postaci aplikacji konsolowej przeznaczonej na system Windows.
3. W komunikacji klient-serwer wykorzystywany jest protokół komunikacyjny Raw – wiadomości przesyłane są bezpośrednio.
4. W ramach serwera nie jest implementowana obsługa rozłączającego się klienta
5. W ramach serwera nie jest implementowana informacja o wyłączeniu serwera przysłana do klienta.
6. Serwer utrzymuje wiele połączeń z klientem na raz.
7. Implementacja serwera nie jest wersjonowana.
8. Serwer po zakończeniu jednego połączenia nie kończy swojej działalności – oczekuje na innych klientów.
9. Hasło jest przesyłane w formie zaszyfowanej
10. Serwer generuje ziarno dla każdej transmisji

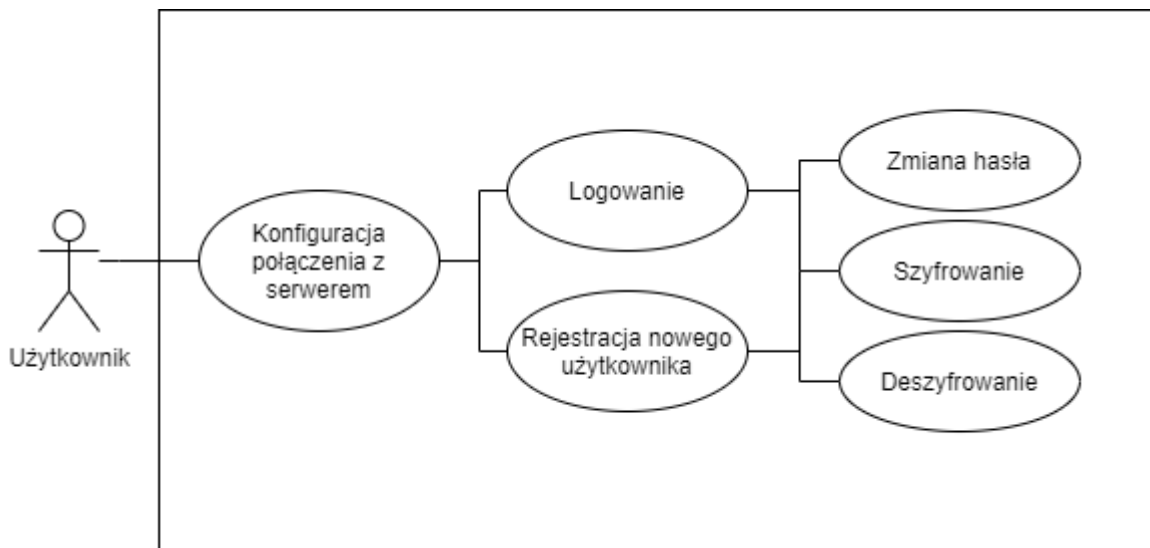


Diagram przypadków użycia

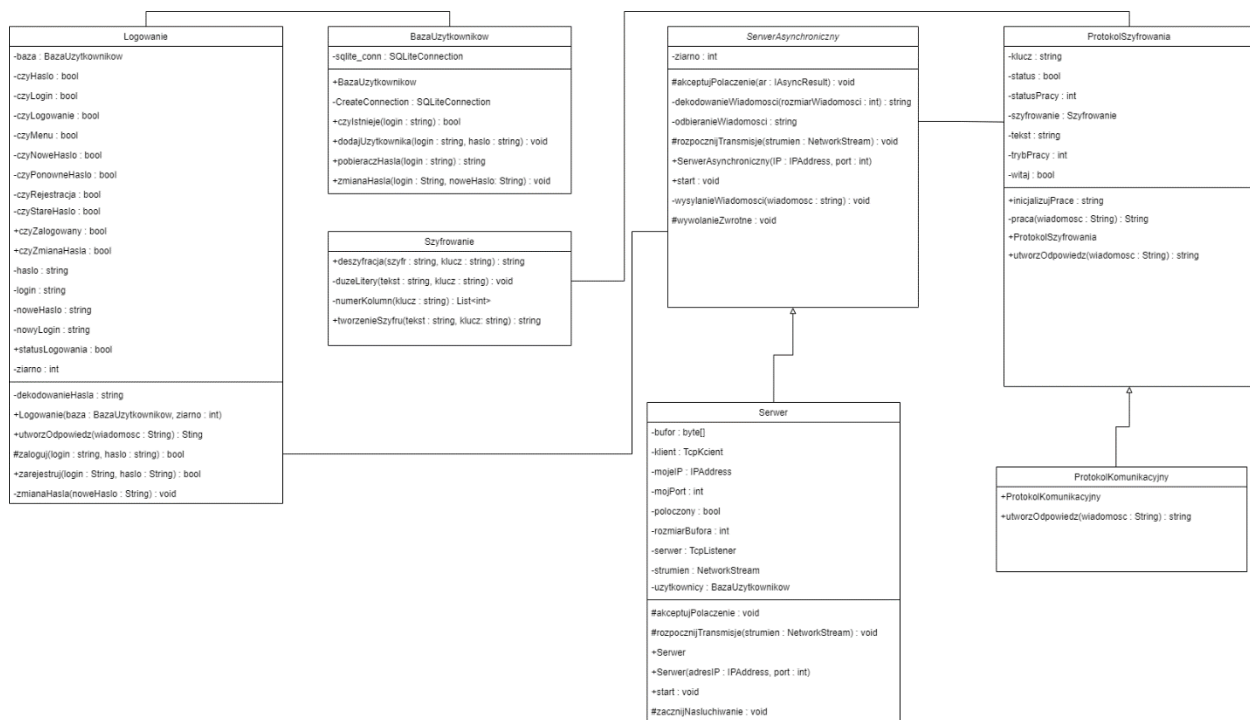


Diagram klas

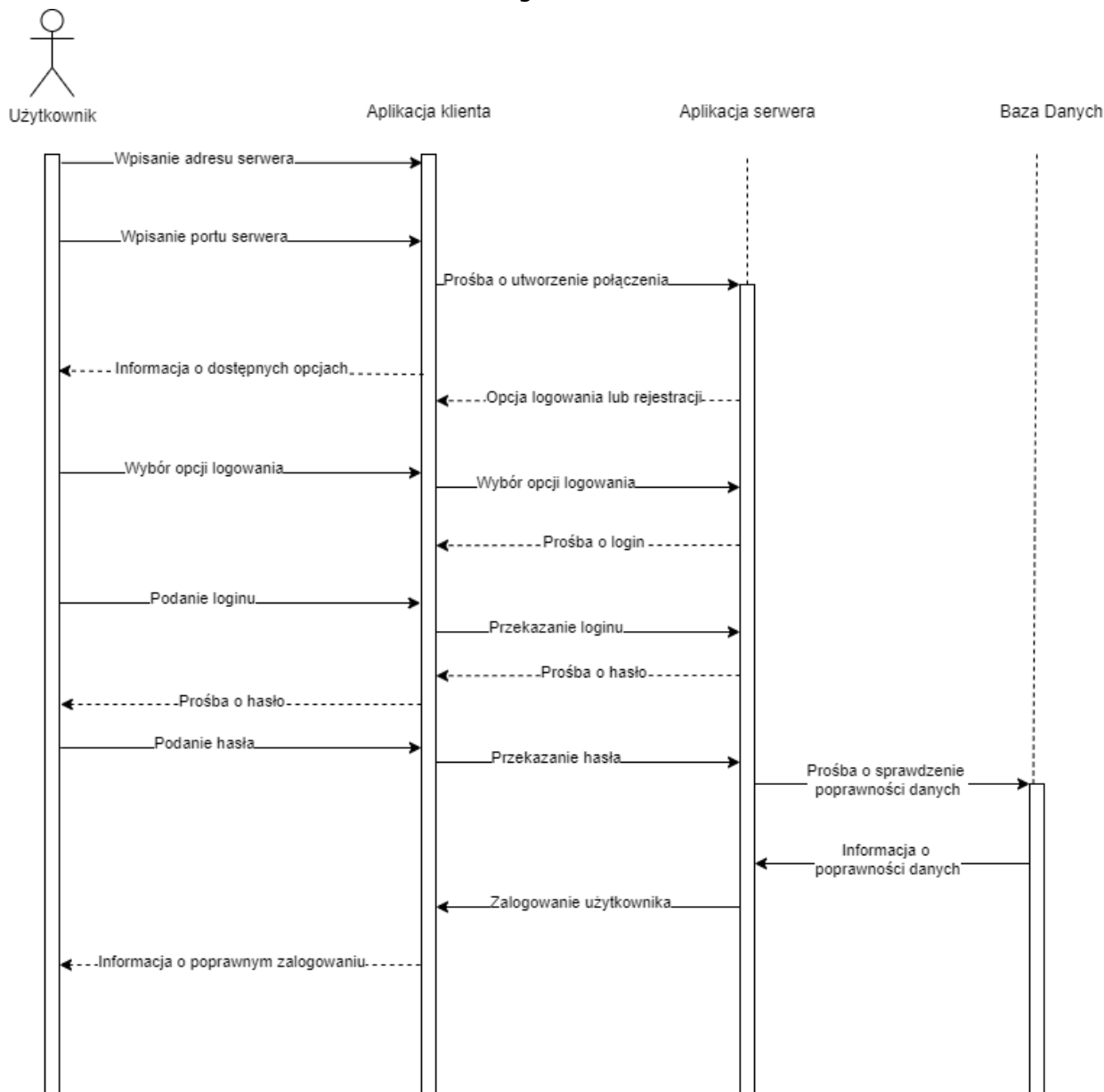


Diagram sekwencji

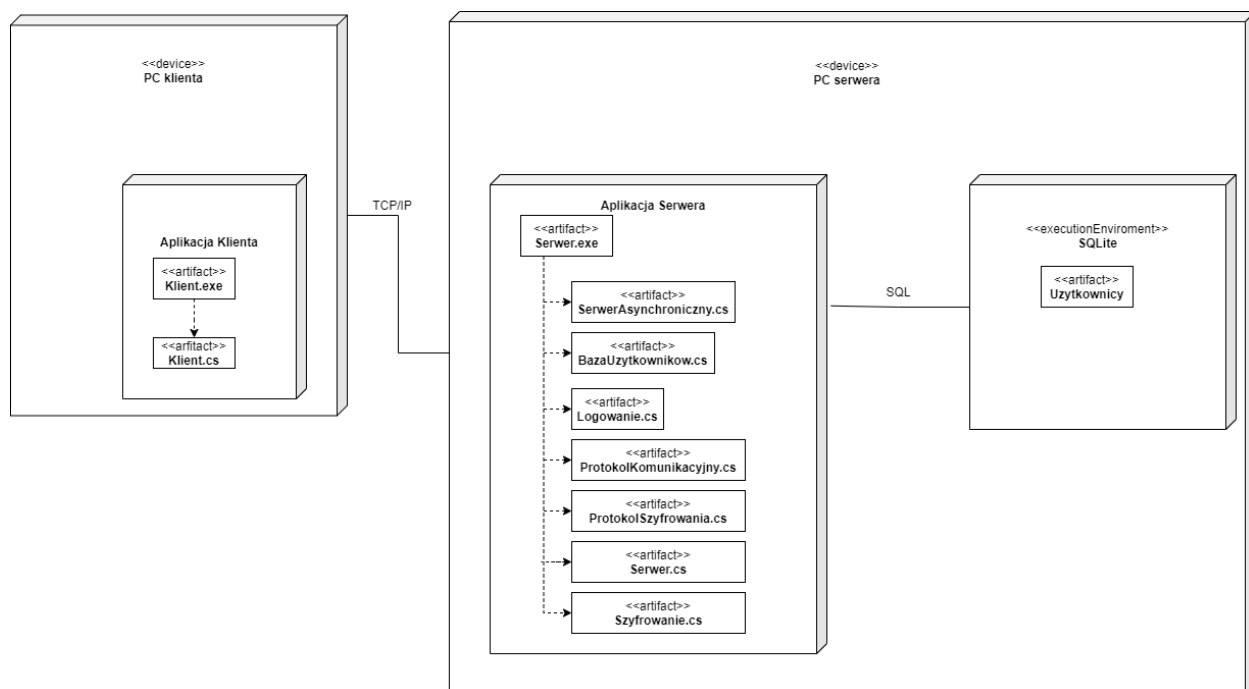


Diagram wdrożenia

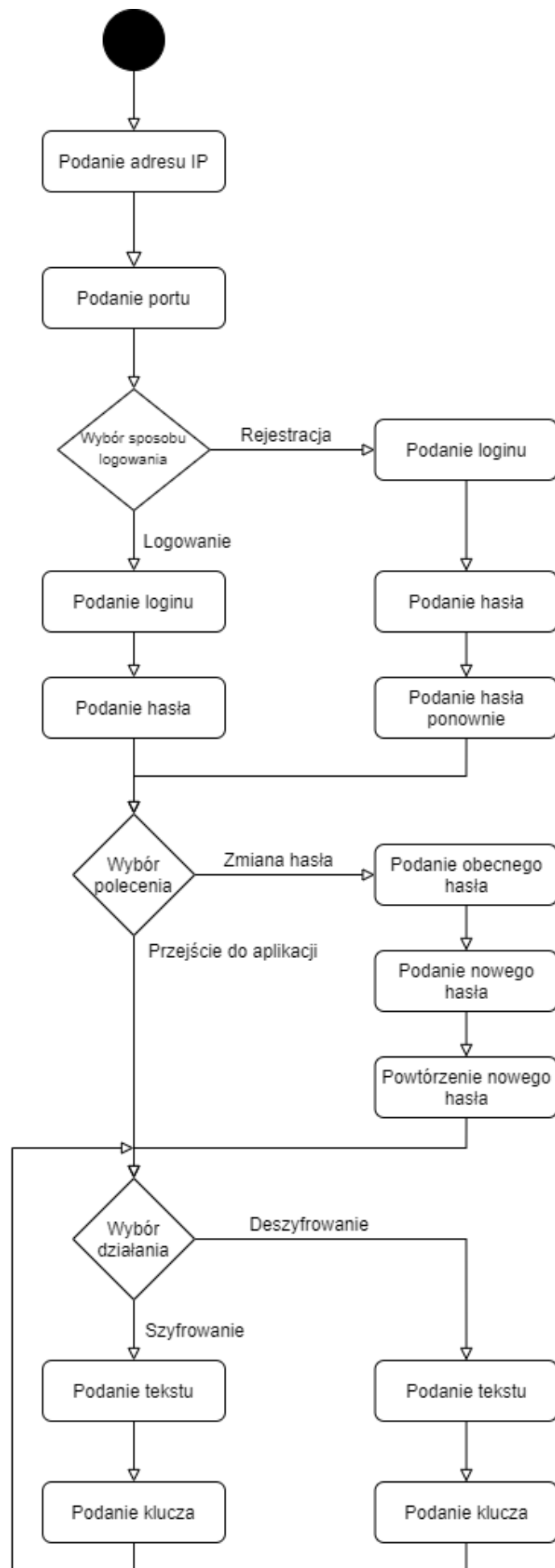


Diagram aktywności