

Informatyka, Bezpieczeństwo systemów informatycznych  
Wydział Elektryczny  
Politechnika Poznańska

Telefonia IP

## **Komunikator głosowy**

**Wojciechowski Aleksander 126878**

**Nitschke Krzysztof 122094**

Poznań, 2018

# Spis treści

<b>Charakterystyka ogólna.</b>	<b>2</b>
<b>Wymagania.</b>	<b>2</b>
Funkcjonalne.	2
Niefunkcjonalne.	3
<b>Architektura systemu.</b>	<b>3</b>
<b>Baza danych.</b>	<b>4</b>
<b>Narzędzia, środowisko, biblioteki.</b>	<b>4</b>
<b>Opis najważniejszych protokołów.</b>	<b>5</b>
<b>Diagramy UML.</b>	<b>5</b>
<b>Widoki interfejsu.</b>	<b>9</b>
<b>Podział pracy.</b>	<b>13</b>

# 1. Charakterystyka ogólna.

Celem projektu jest zaimplementowanie komunikatora głosowego działającego w technologii VoIP.

Aby móc korzystać z aplikacji konsument zostanie poproszony o utworzenie konta. Następnie dwóch użytkowników będzie mogło ustanowić między sobą połączenie poprzez wyszukanie i wybranie odpowiedniego kontaktu z książki adresowej. Dodatkowo zostanie udostępniona opcja utworzenia i zarządzania listą znajomych oraz osób zablokowanych.

Dla zapewnienia bezpieczeństwa osoby korzystające z aplikacji otrzymają możliwość szyfrowania swoich rozmów.

## 2. Wymagania.

Aplikacja ma być prosta w użytkowaniu i wygodna. Powinna sprostać oczekiwaniom stawianym przez klienta. Wobec tego musi spełniać podane wymagania.

### 2.1. Funkcjonalne.

- użytkownik niezarejestrowany w celu utworzenia konta musi się zarejestrować podając login i hasło,
- użytkownik niezalogowany, a zarejestrowany ma możliwość zalogowania się przy pomocy swojego loginu i hasła,
- użytkownik zalogowany może:
  - wykonać połączenie,
  - zakończenie połączenia
  - odebrać bądź odrzucić połączenie wykonywane przez innego użytkownika,
  - wyszukać zarejestrowanego użytkownika na podstawie jego danych (nazwa, imię, nazwisko),
  - utworzyć listę kontaktów,
  - wyświetlić listę kontaktów wraz ze statusami użytkowników,
  - edytować listę kontaktów,
  - dodać innego użytkownika do czarnej listy,
  - wybrać i ustawić jeden z dostępnych statusów (dostępny / niedostępny),
  - zaktualizować swoje obecne dane (nazwa, imię, nazwisko, hasło),
  - wylogować się
- po wylogowaniu status użytkownika zalogowanego automatycznie zostanie ustawiony na niedostępny,
- administrator może:

- uzyskać dostęp do listy wszystkich zarejestrowanych użytkowników.

## 2.2. Niefunkcjonalne.

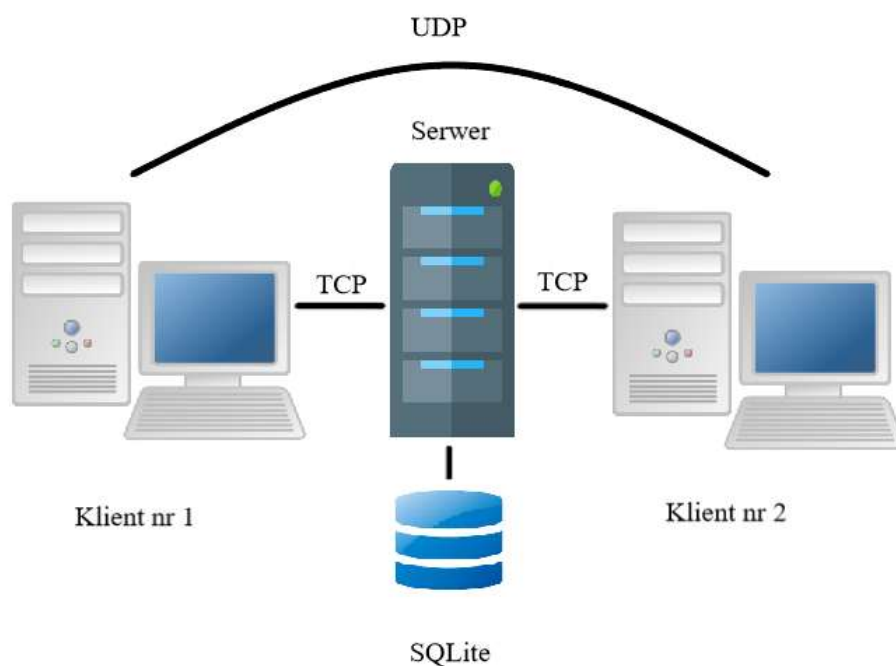
- język programowania: C#,
- działanie w obrębie sieci lokalnej,
- przeznaczone dla systemu operacyjnego: Windows,
- interfejs w języku polskim,
- brak możliwości wykonania więcej niż jednego połączenia w tym samym momencie.

## 3. Architektura systemu.

Na *rysunku 1* przedstawiony został ogólny schemat architektury systemu. System pracuje w architekturze klient - serwer. Za pośrednictwem serwera klient może ustanowić połączenie z innym klientem i rozpocząć rozmowę.

Aplikacja klienta umożliwia użytkownikom zarządzanie swoim kontem oraz wykonywanie połączeń. W celu nawiązania połączenia klient za pomocą protokołu komunikacyjnego TCP/IP wysyła zapytanie do serwera o IP innego klienta. Po jego otrzymaniu za pomocą protokołu UDP ustanawia połączenie.

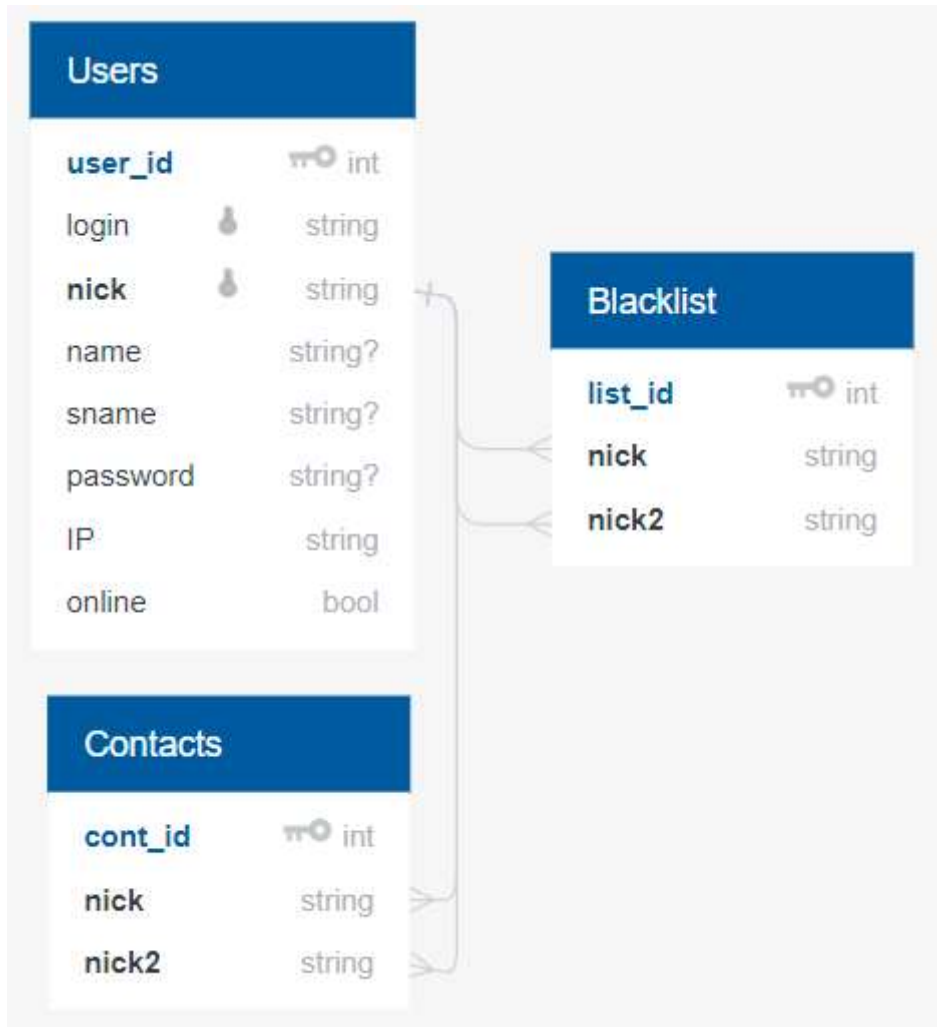
Aplikacja serwera służy do zarządzania bazą danych oraz pośredniczy przy ustanowieniu połączenia między klientami. Pozwala dodawać, wyszukiwać i usuwać konta użytkowników znajdujące się w bazie danych.



*Rysunek 1. Schemat architektury systemu.*

## 4. Baza danych.

Baza danych serwera stworzona po to, aby przechowywać dane klientów. Została stworzona w systemie SQLite.



Rysunek 2. Schemat bazy danych.

## 5. Narzędzia, środowisko, biblioteki.

Do realizacji poszczególnych zadań wykorzystane zostaną następujące narzędzia:

- Narzędzia:
  - Wireshark,
  - Git,
  - Inkscape,
  - GIMP,
  - WPF,
  - C#,
  - SQLite,

- Visual Paradigm.
- Środowisko:
  - Visual Studio.
- Biblioteki:
  - Naudio.

## 6. Opis najważniejszych protokołów.

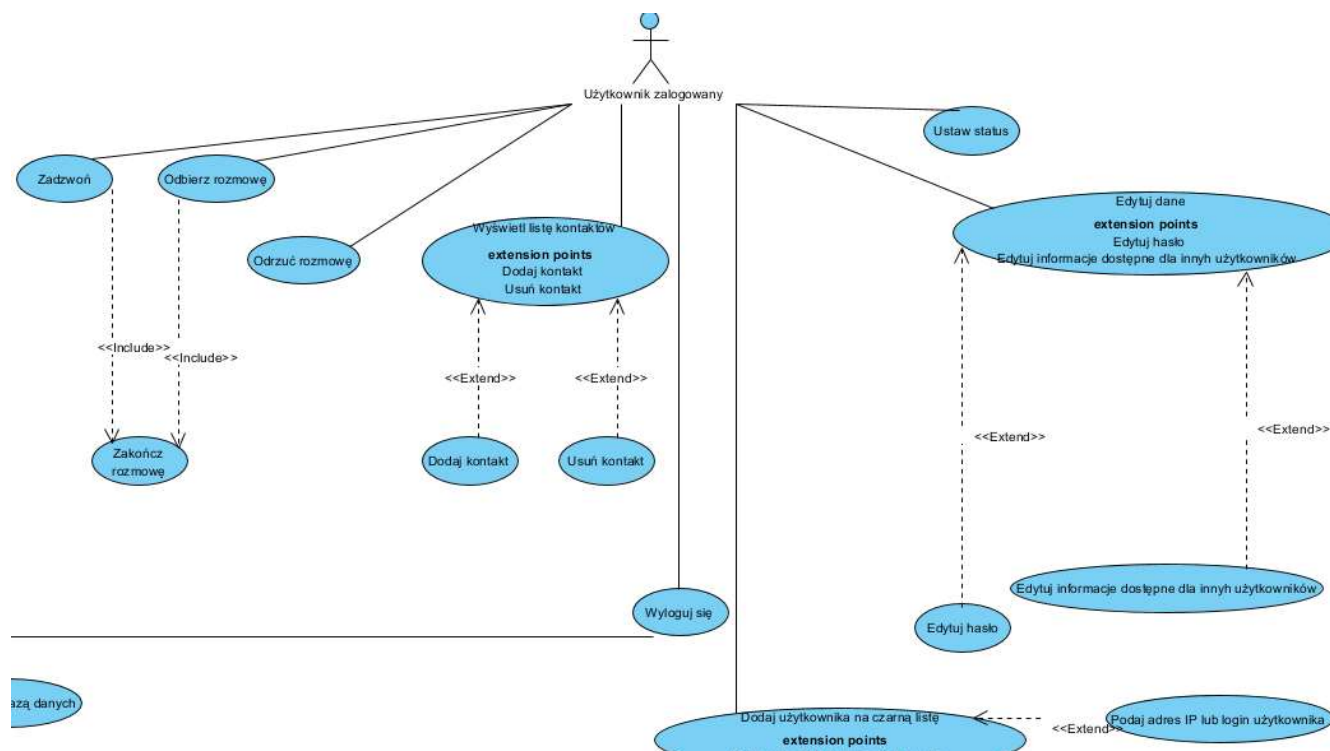
W punkcie tym przedstawiono najważniejsze protokoły wykorzystywane w aplikacji.

- UDP - protokołów warstwy transportowej modelu OSI. Wykorzystywany do transmisji głosu między użytkownikami aplikacji. Protokół ten jest bezpołączeniowy - nie posiada on mechanizmów kontroli przepływu i retransmisji, dzięki czemu jest on szybszy i nadaje się do komunikacji użytkowników w czasie rzeczywistym.
- TCP - protokół połączeniowy - zapewnia retransmisję. Wykorzystywany do przesyłania komend między klientem a serwerem.
- SIP - protokół inicjowania sesji pomiędzy klientami. Jego zamierzeniem jest dostarczenie funkcji obsługi połączenia i innych cech obecnych w publicznej sieci telefonicznej.
- RTP - protokół transmisji danych w czasie rzeczywistym. Pakiet protokołu zawiera informację o typie przesyłanych danych, numer sekwencyjny oraz znacznik czasu. RTP nie gwarantuje jakości usługi oraz jako protokół warstwy transportowej wykorzystuje UDP.

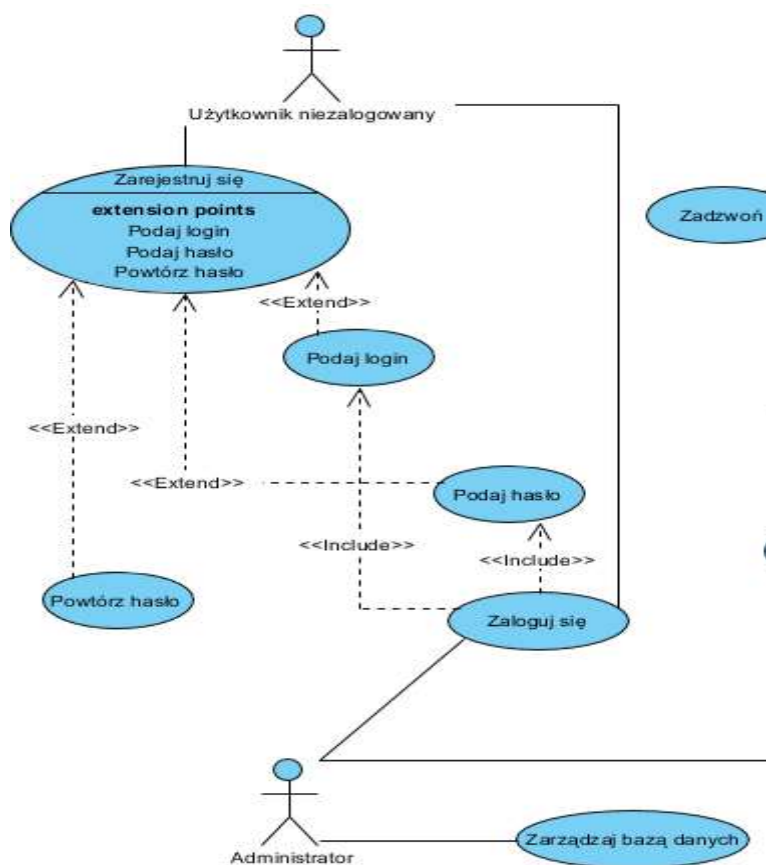
## 6. Diagramy UML.

Poniższe diagramy wygenerowane zostały w programie **Visual Paradigm**.

- Diagram przypadku użycia(ang. use case) z podziałem na aktorów(użytkownik niezalogowany/zalogowany/administrator) przedstawia podstawowe funkcjonalności systemu wraz z otoczeniem, które są widoczne z zewnątrz systemu.

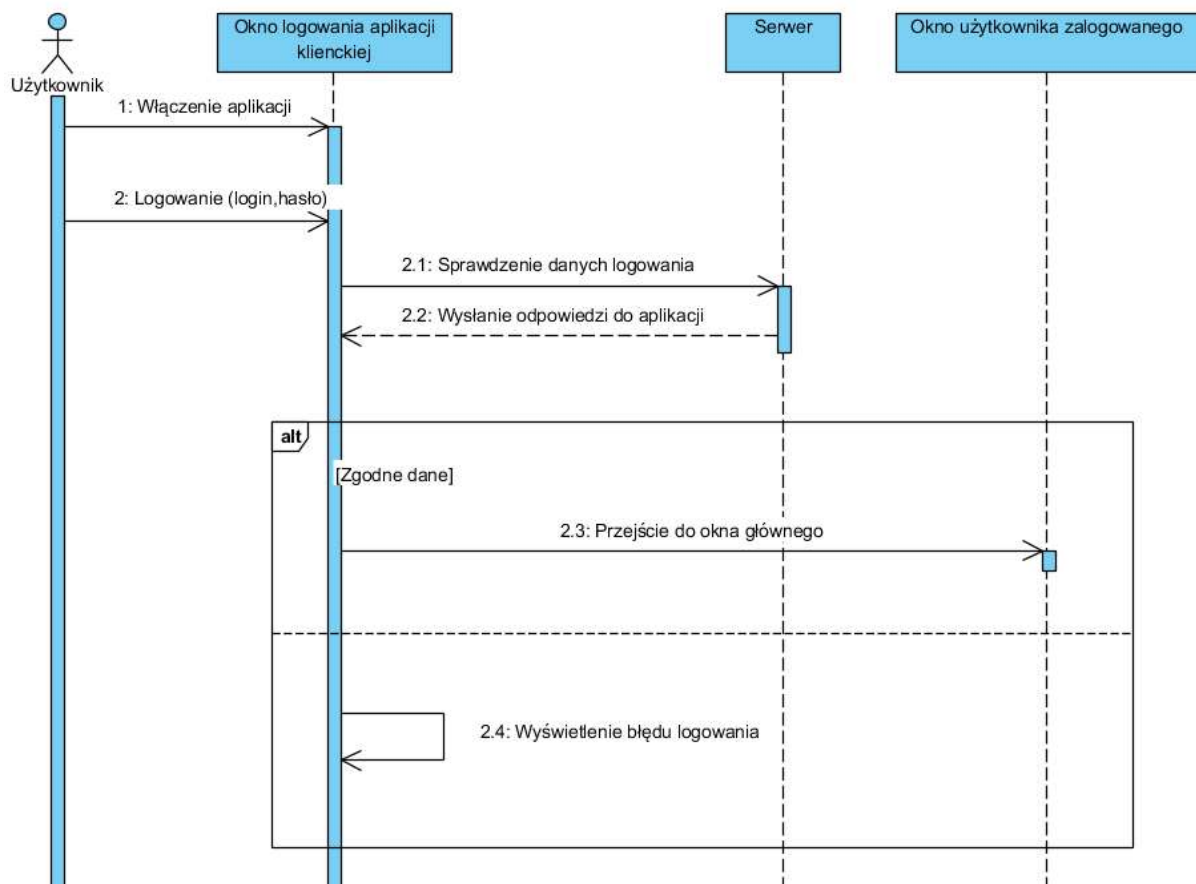


Rysunek 3.1. Diagram UML przypadku użycia z podziałem na aktorów cz.1

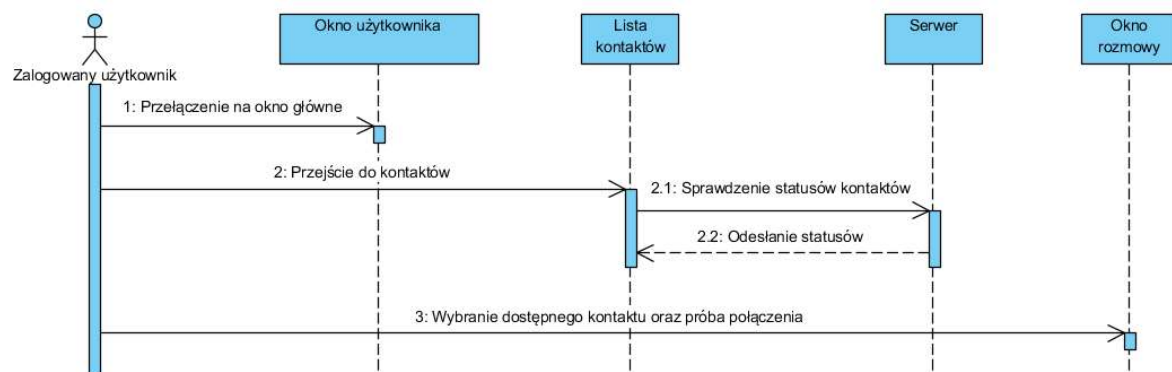


Rysunek 3.2. Diagram UML przypadku użycia z podziałem na aktorów cz.2

- Diagram sekwencji służy do prezentowania interakcji pomiędzy obiektami wraz z uwzględnieniem w czasie komunikatów, jakie są przesyłane pomiędzy nimi. Stosowany jest do przedstawienia zachowania systemu w kontekście różnych scenariuszy.



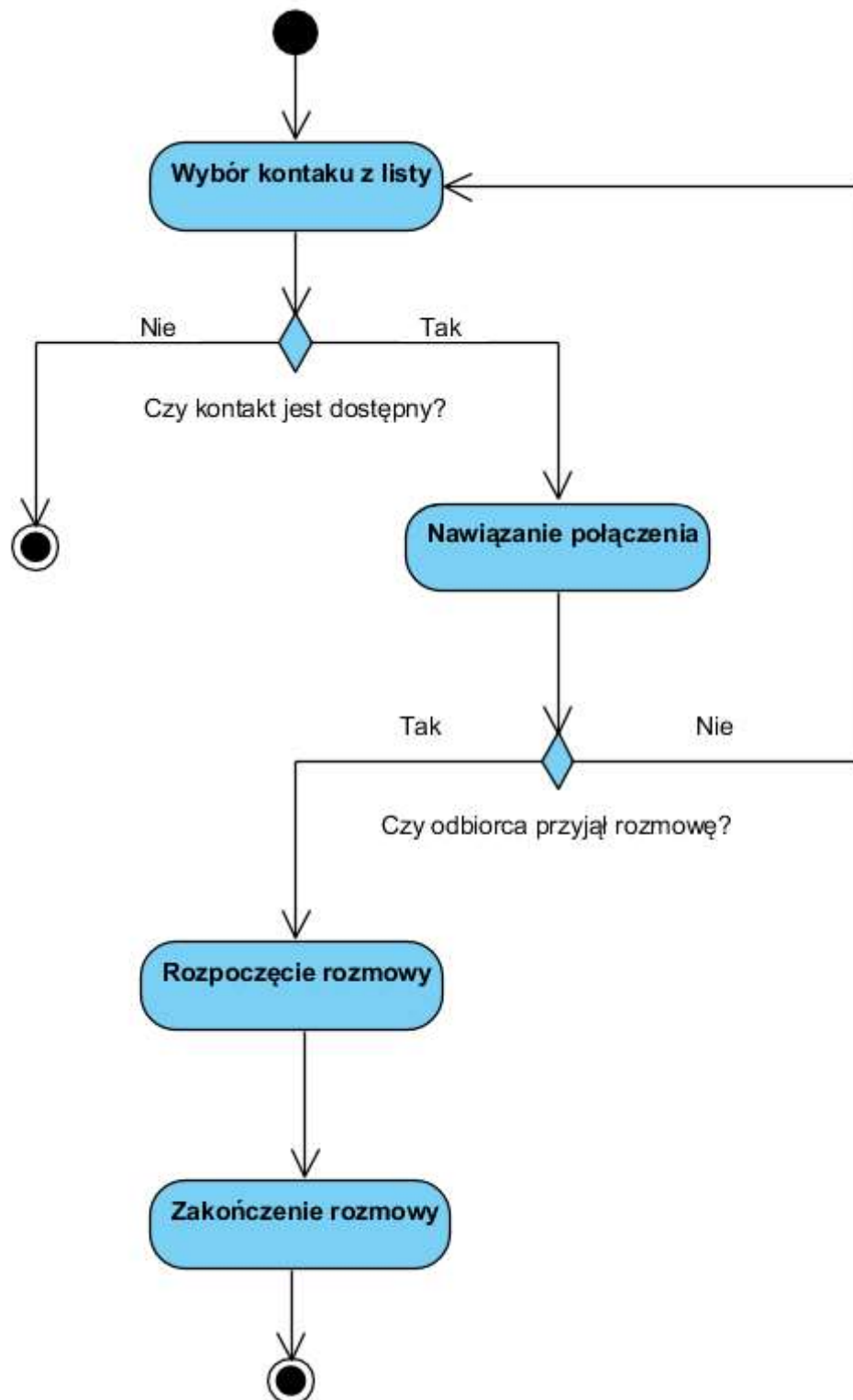
Rysunek 4. Diagram sekwencji - logowanie użytkownika



Rysunek 5. Diagram sekwencji - wybranie kontaktu z listy



- Diagram czynności służy do modelowania czynności i zakresu odpowiedzialności elementów bądź użytkowników systemu.



Rysunek 6. Diagram czynności - rozpoczęcie rozmowy

## 7. Widoki interfejsu.

Interfejs został stworzony z wykorzystaniem Windows Form C# w Visual Studio.

- Ekran startowy zawiera okno logowania użytkownika, możliwość przejścia do okna rejestracji.

The image shows a screenshot of a Windows application window with a blue background. The window has a standard Windows title bar with a red close button in the top right corner. The main content area is blue and contains the following elements: the word "Login" in black text, a white rectangular input field, the word "Hasło" in black text, another white rectangular input field, a light blue button with the text "Zaloguj", and a light blue button with the text "Zarejestruj". All elements are centered vertically and horizontally.

*Rysunek 7. Ekran startowy logowanie/rejestracja użytkownika*

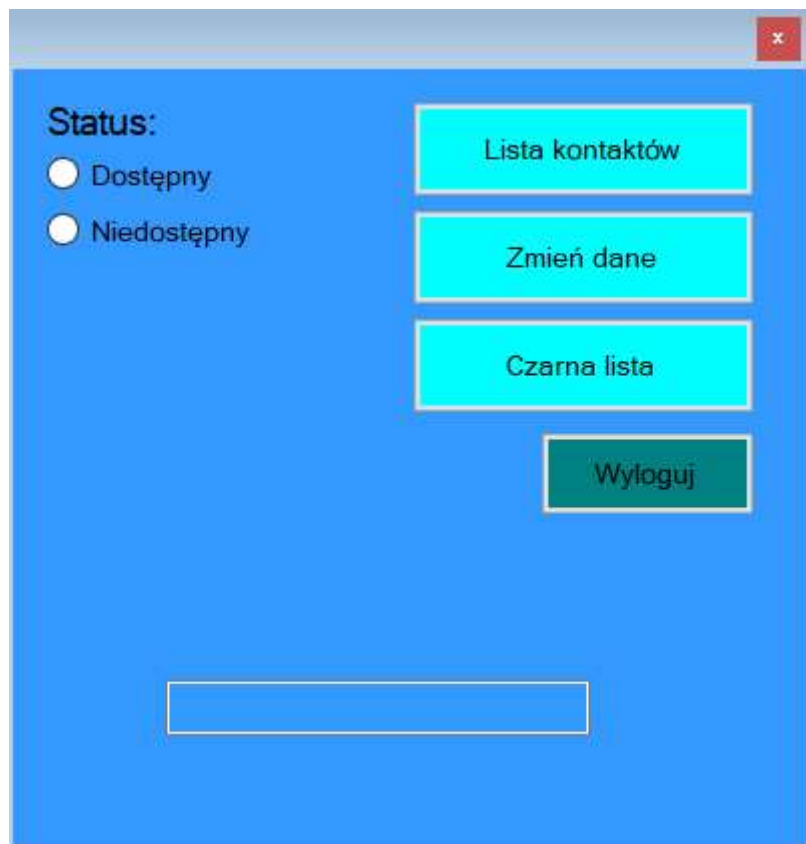
- Ekran rejestracji zawiera wszystkie informacje zawarte w wymaganiach funkcjonalnych, przycisk do zatwierdzenia rejestracji oraz przycisk powrotu do widoku logowania.



The image shows a registration form window titled "Rejestracja". The window has a blue background and a light blue border. It contains five input fields for registration: "Login:", "Nazwa:", "Imię:", "Nazwisko:", and "Hasło:". Below the input fields are two buttons: a green "Zarejestruj" button and a grey "Powrót" button. The window also has a standard Windows-style title bar with a close button in the top right corner.


*Rysunek 8. Widok formularza widoku rejestracji*

- Widok główny po zalogowaniu użytkownika. Zawiera przejścia do wszystkich opcji aplikacji oraz możliwość zmieniania statusu.



*Rysunek 9. Widok po zalogowaniu użytkownika*

- Okno zmiany danych pozwala użytkownikowi na zmianę danych ustalonych przy rejestracji.



The image shows a graphical user interface window for changing user data. The window has a blue background and a standard Windows-style title bar at the top with a red close button. The form contains four text input fields, each preceded by a label: 'Nazwa:', 'Imię:', 'Nazwisko:', and 'Hasło:'. Below the input fields are two buttons: a green button labeled 'Zmień dane' and a cyan button labeled 'Powrót'.

*Rysunek 10. Widok formularza do zmiany danych użytkownika*

- Okno z listą kontaktów pozwala użytkownikowi na dodanie, usunięcie oraz na rozpoczęcie rozmowy z wybranym kontaktem.

The screenshot shows a window with a blue background. At the top left is a white text input field. Below it, the text "Dostępni znajomi:" is displayed. Underneath this text is a list box containing two items: "olo" and "xaxa". To the right of the list box is a cyan button labeled "Dodaj kontakt". Below the list box, the text "Niedostępni znajomi:" is displayed, followed by a list box containing the item "krz". To the right of this list box is a cyan button labeled "Usuń kontakt". At the bottom right of the window is a green button labeled "Zadzwoń".

*Rysunek 11. Widok okna z listą kontaktów*

- Okno wyskakujące podczas otrzymywania połączenia.

The screenshot shows a dialog window with a blue background. At the top left is a green button labeled "Odbierz". At the top right is a red button labeled "Odrzuć". In the center of the window is a white rectangular text input field. At the bottom center is a green button labeled "Rozłącz".

*Rysunek 12. Widok okna przychodzącego połączenia*

- Aplikacja serwera wyświetla odebrane odebrane od klientów komunikaty, listę dostępnych użytkowników.

```
Waiting for a connection... Connected!
Received: = k r z 1 0 :
Name: www      Sname: eee
Name:  Sname:
Name: xaxa     Sname: xaxa
Name: olo      Sname: woj
Name: krz      Sname: krz
Name: krz10    Sname: krz10
Waiting for a connection... Connected!
Received: 2 k r z 1 0 :
Name: www      Sname: eee
Name:  Sname:
Name: xaxa     Sname: xaxa
Name: olo      Sname: woj
Name: krz      Sname: krz
Waiting for a connection...
```

*Rysunek 13. Widok okna serwera*

## 8. Podział pracy.

Krzysztof Nitschke:

- baza danych
- aplikacja serwera
- połączenie TCP klient - serwer

Aleksander Wojciechowski:

- komunikaty aplikacji
- przesyłanie dźwięku
- połączenie modułów