

Politechnika Poznańska

# Telefonia IP

Dokumentacja projektowa

*No Title Call*

Marcin Złotek 136334

marcin.zlotek@student.put.poznan.pl

Poznań, 2020

# Spis treści

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Ogólna charakterystyka</b>                        | <b>2</b>  |
| <b>2</b> | <b>Architektura systemu</b>                          | <b>2</b>  |
| <b>3</b> | <b>Wymagania</b>                                     | <b>2</b>  |
| 3.1      | Funkcjonalne . . . . .                               | 2         |
| 3.1.1    | Użytkownik niezalogowany/niezarejestrowany . . . . . | 2         |
| 3.1.2    | Użytkownik zalogowany . . . . .                      | 2         |
| 3.1.3    | Serwer . . . . .                                     | 3         |
| 3.1.4    | Administrator serwera . . . . .                      | 3         |
| 3.2      | Pozafunkcjonalne . . . . .                           | 3         |
| <b>4</b> | <b>Narzędzia, środowisko, biblioteki</b>             | <b>4</b>  |
| <b>5</b> | <b>Diagramy UML</b>                                  | <b>5</b>  |
| 5.1      | Przypadków użycia . . . . .                          | 5         |
| 5.2      | Stanów . . . . .                                     | 6         |
| 5.3      | Klas . . . . .                                       | 7         |
| 5.4      | Sekwencji . . . . .                                  | 8         |
| <b>6</b> | <b>GUI</b>   | <b>10</b> |
| <b>7</b> | <b>Testy i przebiegi</b>                             | <b>13</b> |
| <b>8</b> | <b>Podsumowanie</b>                                  | <b>15</b> |
| 8.1      | Cele zrealizowane . . . . .                          | 15        |
| 8.2      | Problemy . . . . .                                   | 15        |

# 1 Ogólna charakterystyka

Tematem projektu jest system komunikacji głosowej poprzez protokół IP. System pełni rolę skrzynki głosowej, na którą mogą nagrać się użytkownicy. Aplikacja klienta będzie posiadała interfejs tekstowy.

Projekt nosi nazwę *NoTitleCall*, sloganem jest: „*listen a lot, talk more*”.

## 2 Architektura systemu

Architektura systemu jest w postaci klient-serwer. Użytkownik będzie się łączył z serwerem w celu nagrania lub odebrania wiadomości.

Do komunikacji z serwerem służy protokół TCP oraz własna implementacja protokołów do nawiązywania połączenia. Do przechowywania danych o użytkownikach serwer wykorzystuje pliki *XML*. Poufność przekazywanych danych jest zapewniona przez protokół *TLS*.

## 3 Wymagania

### 3.1 Funkcjonalne

Wymagania funkcjonalne dla aplikacji klienta i serwera pośredniczącego.

#### 3.1.1 Użytkownik niezalogowany/niezarejestrowany

Wymagania dla aplikacji klienta niezalogowanego/niezarejestrowanego:

- utworzenie konta użytkownika,
- utworzenie hasła do konta,

#### 3.1.2 Użytkownik zalogowany

Wymagania dla aplikacji klienta zalogowanego:

- nagranie wiadomości głosowej,
- wybór osoby do której chcemy wysłać wiadomość,
- odsłuchanie wiadomości głosowej,
- sprawdzenie czy są dostępne nowe wiadomości głosowe.

### 3.1.3 Serwer

Wymagania dla serwera pośredniczącego w transmisji:

- informowanie użytkownika o błędach w komunikacji,
- przechowywanie informacji dotyczącej użytkowników (hasła, pseudonimy),
- rejestrowanie nowych użytkowników,
- logowanie istniejących użytkowników,
- weryfikacja nadawcy danych przy pomocy jednorazowych tokenów,
- możliwość zalogowania się administratora systemu,
- usuwanie odsłuchanych wiadomości z pamięci.

### 3.1.4 Administrator serwera

Wymagania dla administratora zarządzającego serwerem:

- zalogowanie się na serwer,
- sprawdzenia aktualnie zalogowanych użytkowników,
- usuwanie kont użytkowników,
- sprawdzenie logów z powiadomieniami serwera,
- wyłączenia serwera.

## 3.2 Pozafunkcjonalne

Wymagania pozafunkcjonalne odnoszące się do całego systemu. Są to wymagania dotyczące wydajności, bezpieczeństwa i użyteczności systemu.

- system musi posiadać serwer wielowątkowy,
- serwer posiada stały, znany aplikacji klienckiej, adres IP,
- hasło służące do zalogowania administratora na serwerze musi być niewidoczne podczas wpisywania,
- aplikacja użytkownika posiada tekstowy interfejs użytkownika,
- komunikacja klient-serwer jest szyfrowana przy pomocy SSL/TLSv1.2,

- system musi działać na systemie operacyjnym Windows 10 lub nowszym,
- system powinien przechowywać hasła w postaci skrótu utworzonego funkcją SHA-256,
- system wymaga połączenia internetowego o przepustowości 100kB/s (kilobajtów na sekundę) i większej,
- nazwą identyfikacyjną użytkownika jest jego adres e-mail,
- nazwa identyfikacyjna użytkownika musi być unikalna,
- dane dotyczące użytkowników powinny być przechowywane w plikach XML.

## 4 Narzędzia, środowisko, biblioteki

Zbiór używanych do stworzenia projektu narzędzi i bibliotek. Wymienione zostały także używane środowiska programistyczne (*IDE*), które umożliwiły stworzenie całego systemu.

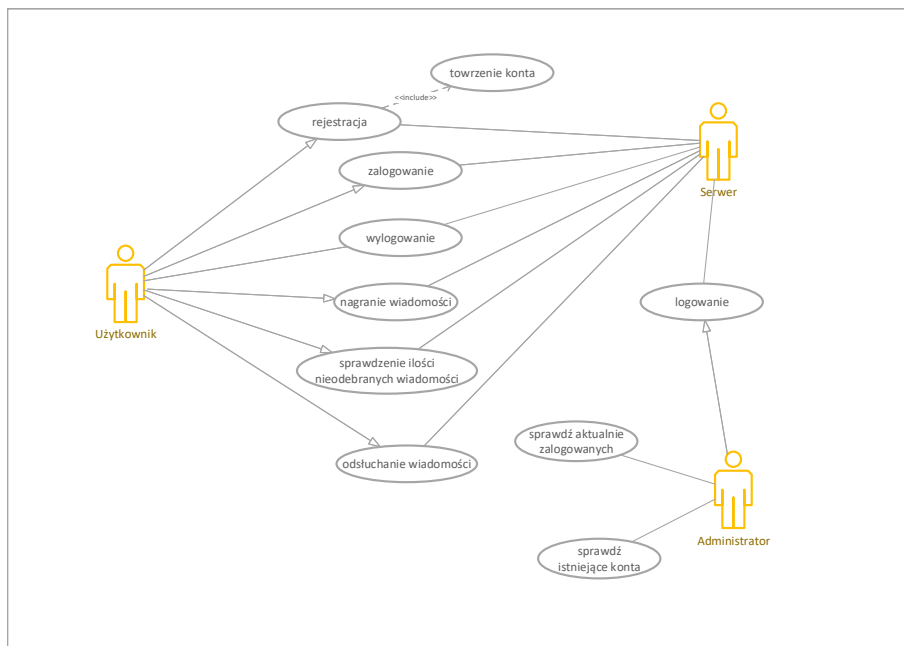
- Narzędzia
  - C#,
  - XML.
- Środowisko
  - MS Visual Studio 2015 oraz 2019,
  - RawCap,
  - Wireshark
  - MS Visio 2016
  - TeXStudio.
- Biblioteki / Standardy
  - X.509,
  - SSL/TLSv1.2,
  - NAudio v10.

## 5 Diagramy UML

Diagramy UML przedstawiające budowę i działanie systemu.

### 5.1 Przypadków użycia

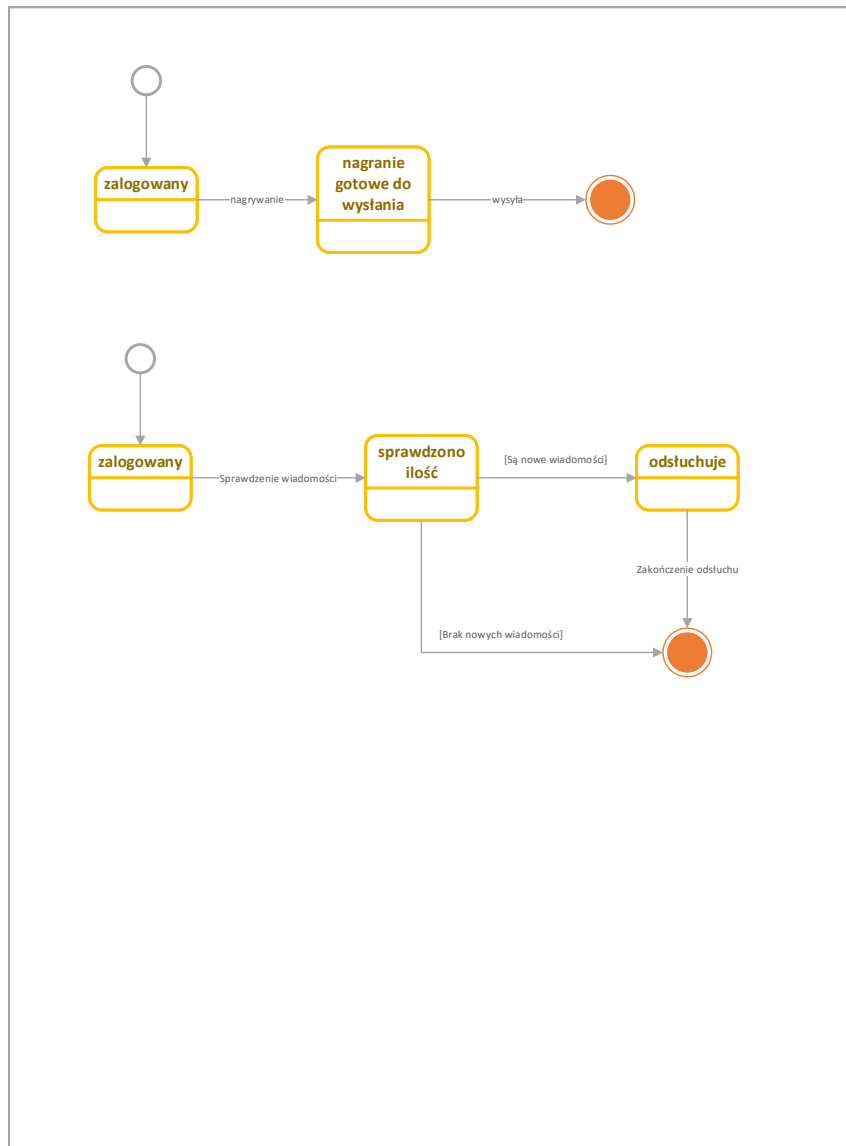
Diagram przypadków użycia z podziałem na aktorów.



Rysunek 1: Diagram UML przypadków użycia

## 5.2 Stanów

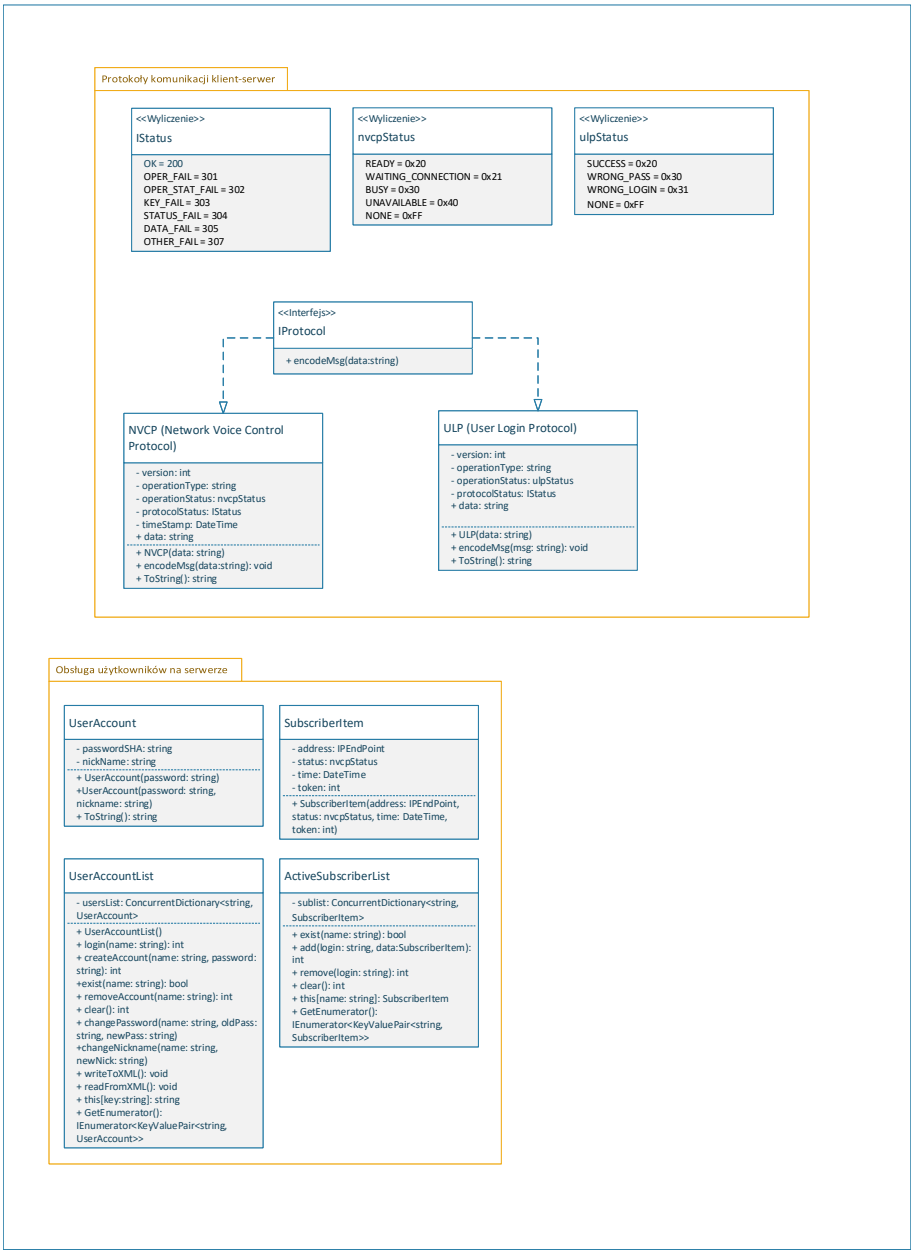
Diagramy stanów: nagrywanie i odbieranie wiadomości.



Rysunek 2: Diagram UML stanów - nagrywanie wiadomości (u góry) i odbieranie (u dołu)

### 5.3 Klas

Diagramy klas protokołów, aplikacji serwera i klienta.

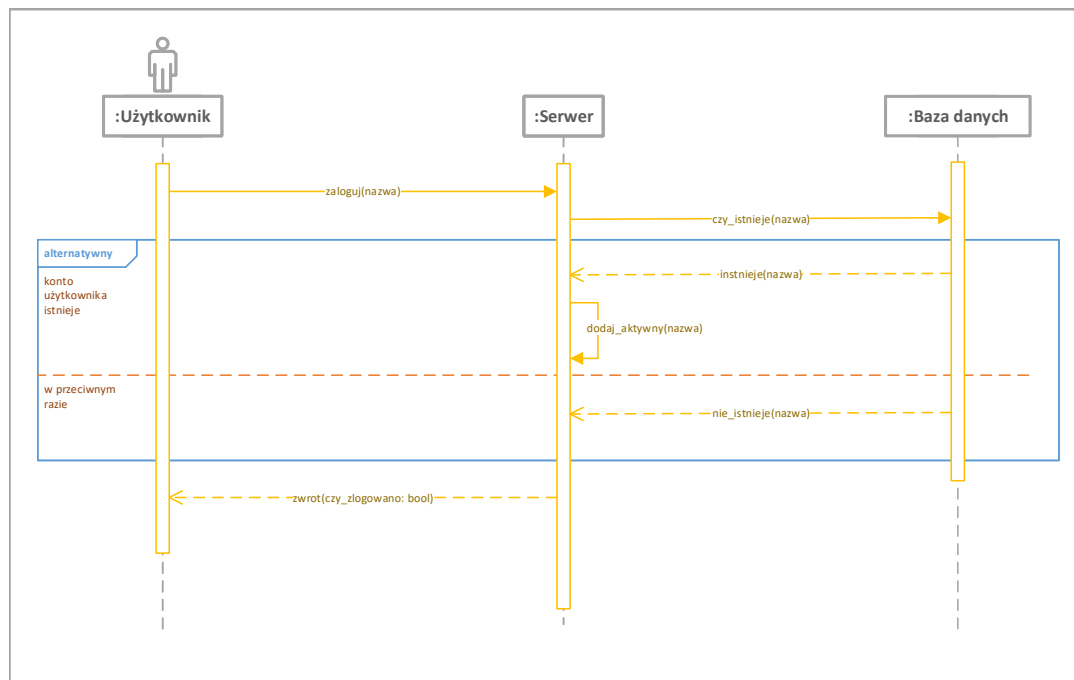


Rysunek 3: Diagram UML klas

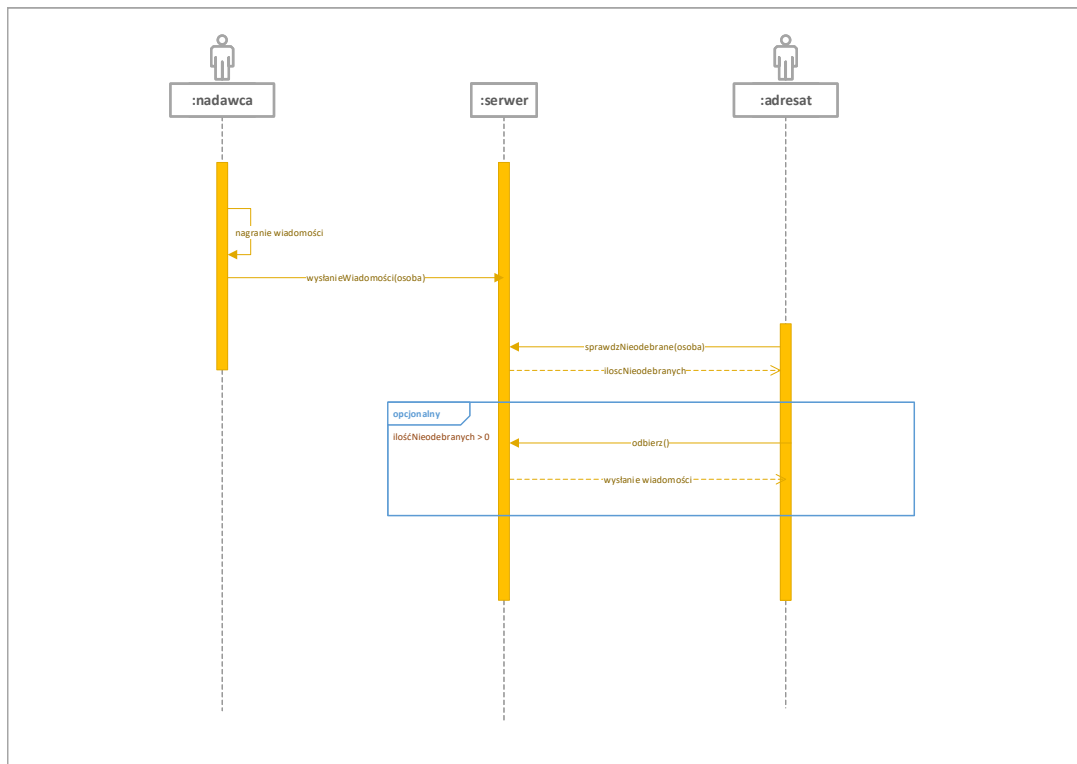


## 5.4 Sekwencji

Diagramy sekwencji: próby nawiązania połączenia oraz logowania od systemu.



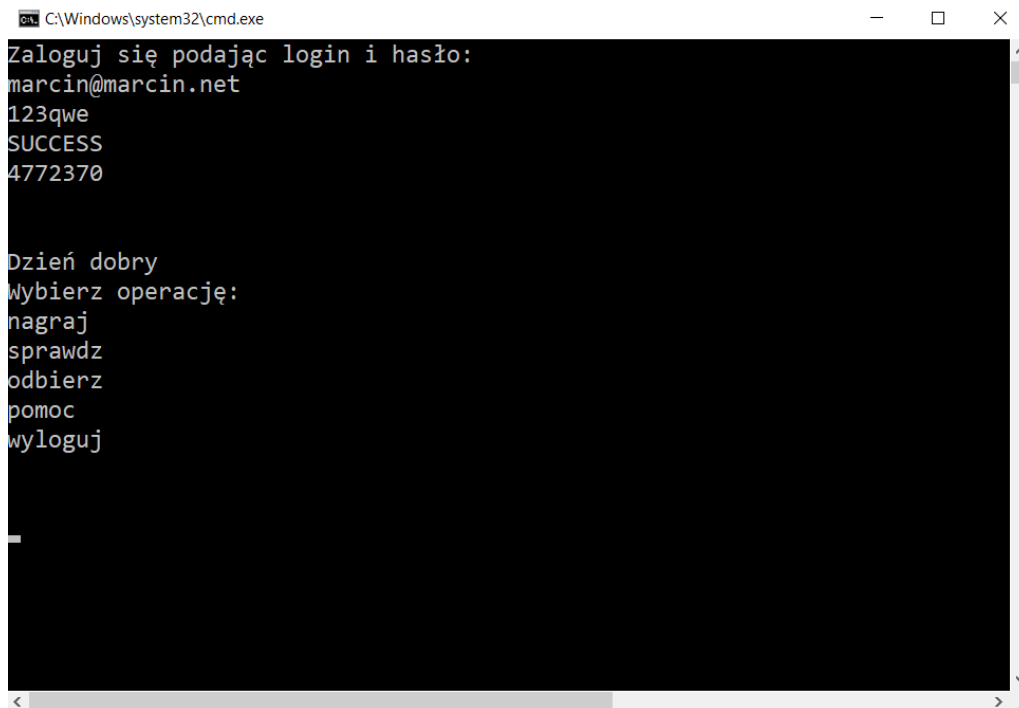
Rysunek 4: Diagram UML sekwencji - logowanie



Rysunek 5: Diagram UML sekwencji - przesłanie wiadomości głosowej

## 6 GUI

Projekt interfejsu aplikacji klienta i serwera.

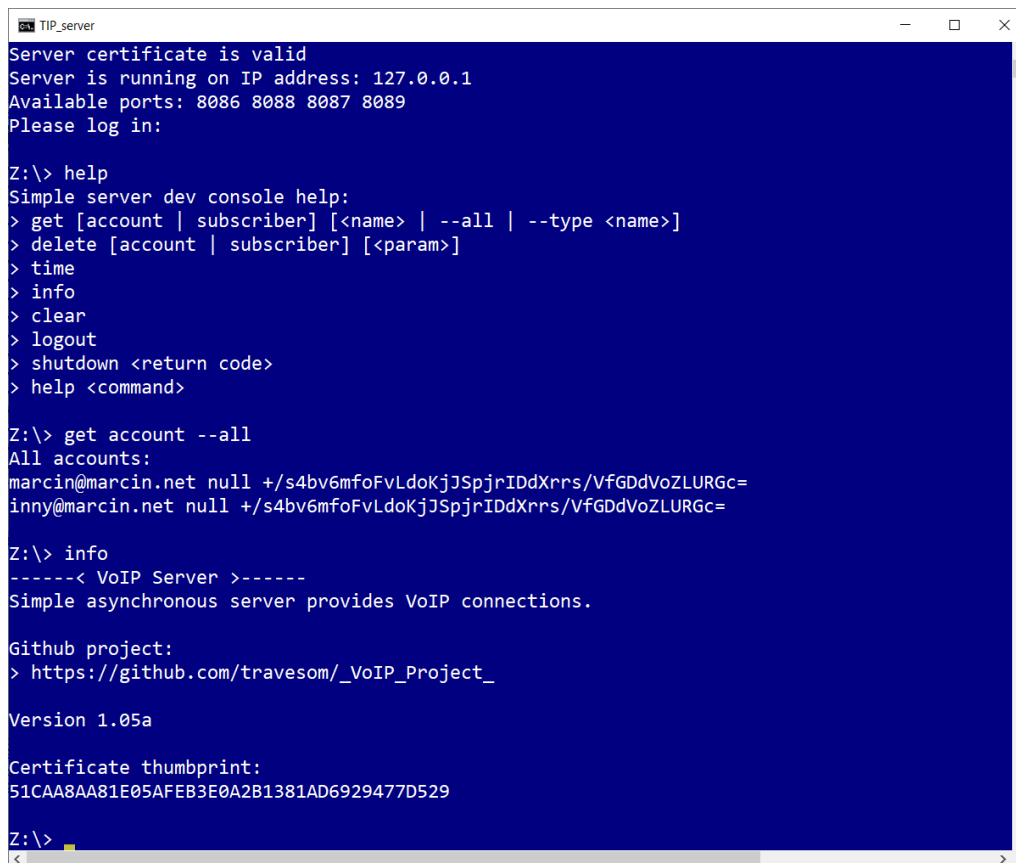


```
C:\Windows\system32\cmd.exe
Zaloguj się podając login i hasło:
marcin@marcin.net
123qwe
SUCCESS
4772370

Dzień dobry
Wybierz operację:
nagraj
sprawdz
odbierz
pomoc
wyloguj
```

Rysunek 6: Ekran główny - aplikacja klienta

Poniżej umieszczono zrzuty ekranu konsoli serwera. Przedstawiono przykładowe komendy dostępne dla administratora po zalogowaniu się. Hasło logowania nie pojawia się przy wpisywaniu, jest to zabieg celowy.



```
TIP_server
Server certificate is valid
Server is running on IP address: 127.0.0.1
Available ports: 8086 8088 8087 8089
Please log in:

Z:\> help
Simple server dev console help:
> get [account | subscriber] [<name> | --all | --type <name>]
> delete [account | subscriber] [<param>]
> time
> info
> clear
> logout
> shutdown <return code>
> help <command>

Z:\> get account --all
All accounts:
marcin@marcin.net null +/s4bv6mfoFvLdoKjJSpjrIDdXrrs/VfGDdVoZLURGc=
inny@marcin.net null +/s4bv6mfoFvLdoKjJSpjrIDdXrrs/VfGDdVoZLURGc=

Z:\> info
-----< VoIP Server >-----
Simple asynchronous server provides VoIP connections.

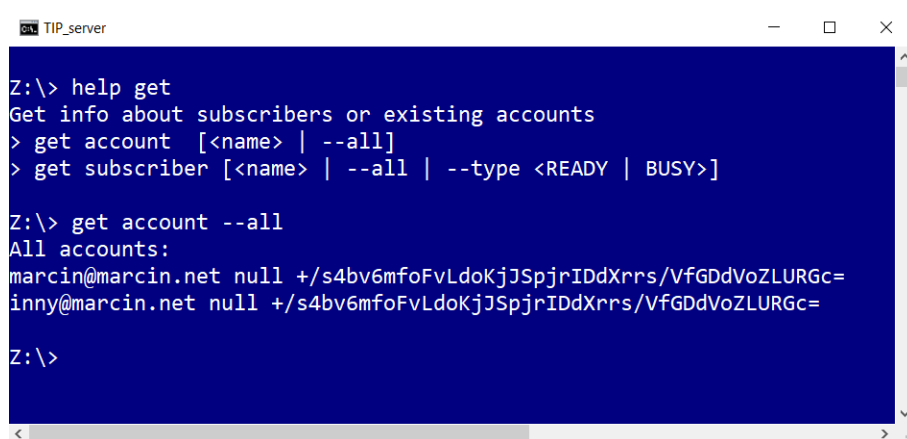
Github project:
> https://github.com/travesom/_VoIP_Project_

Version 1.05a

Certificate thumbprint:
51CAA8AA81E05AFEB3E0A2B1381AD6929477D529

Z:\>
```

Rysunek 7: Konsola serwera

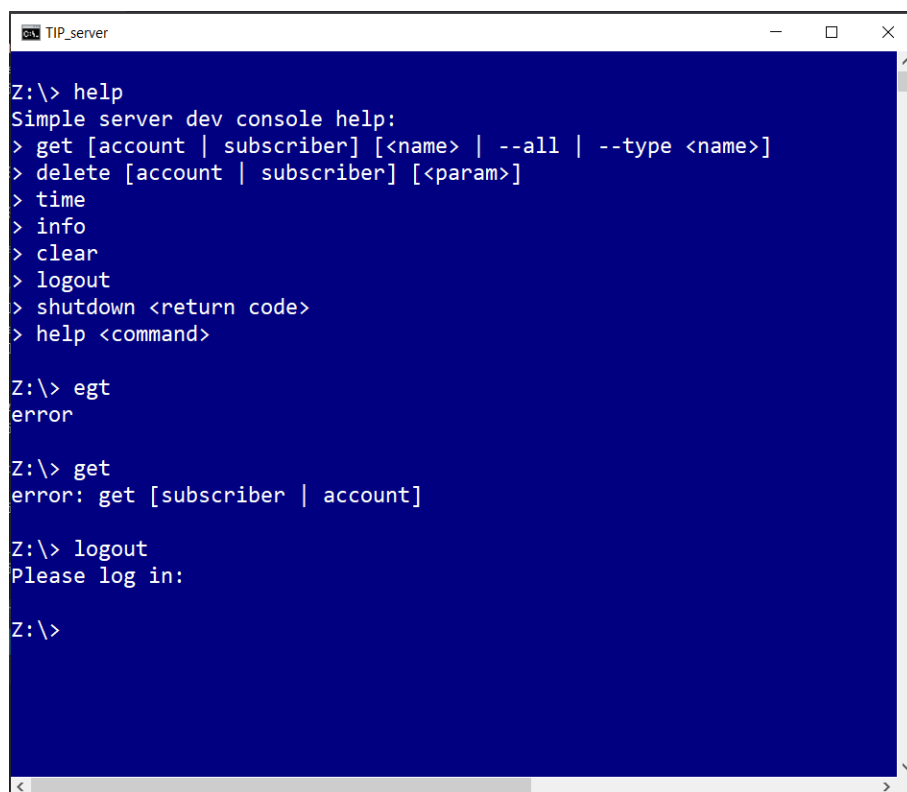


```
Z:\> help get
Get info about subscribers or existing accounts
> get account [<name> | --all]
> get subscriber [<name> | --all | --type <READY | BUSY>]

Z:\> get account --all
All accounts:
marcin@marcin.net null +/s4bv6mfoFvLdoKjJSpjrIDdXrrs/VfGDdVoZLURGc=
inny@marcin.net null +/s4bv6mfoFvLdoKjJSpjrIDdXrrs/VfGDdVoZLURGc=

Z:\>
```

Rysunek 8: Konsola serwera



```
Z:\> help
Simple server dev console help:
> get [account | subscriber] [<name> | --all | --type <name>]
> delete [account | subscriber] [<param>]
> time
> info
> clear
> logout
> shutdown <return code>
> help <command>

Z:\> egt
error

Z:\> get
error: get [subscriber | account]

Z:\> logout
Please log in:

Z:\>
```

Rysunek 9: Konsola serwera

## 7 Testy i przebiegi

Poniżej przedstawiono fragmenty transmisji klient-serwer. Całość przygotowanej transmisji znajduje się w pliku *tip\_spr.pcap*. Transmisja (wysłanie i odebranie) przykładowej wiadomości głosowej znajduje się w pliku *tip\_voice.pcap*.

```
C:\Windows\system32\cmd.exe
Uwierzytelniono
Wysłano: ver#1' oper#REGISTER' status#NONE' ulp#OK' data#marcin@marcin.net'
Odebrano: ver#1' oper#REGISTER' status#NONE' ulp#DATA_FAIL' data#''

Uwierzytelniono
Wysłano: ver#1' oper#REGISTER' status#NONE' ulp#OK' data#marcin@marcin.1234 123qwe'
Odebrano: ver#1' oper#REGISTER' status#NONE' ulp#DATA_FAIL' data#''

Uwierzytelniono
Wysłano: ver#1' oper#REGISTER' status#NONE' ulp#OK' data#marcin@marcin.net 123qwe'
Odebrano: ver#1' oper#REGISTER' status#SUCCESS' ulp#OK' data#''

Uwierzytelniono
Wysłano: ver#1' oper#LOGIN' status#NONE' ulp#OK' data#marcin@marcin.net'
Odebrano: ver#1' oper#LOGIN' status#NONE' ulp#DATA_FAIL' data#''

Uwierzytelniono
Wysłano: ver#1' oper#LOGIN' status#NONE' ulp#OK' data#marcin@marcin.net 1qaz2wsx'
Odebrano: ver#1' oper#LOGIN' status#WRONG_PASS' ulp#OK' data#''

Uwierzytelniono
Wysłano: ver#1' oper#LOGIN' status#NONE' ulp#OK' data#marcin@marcin.net 123qwe'
Odebrano: ver#1' oper#LOGIN' status#SUCCESS' ulp#OK' data#9854416'

Uwierzytelniono
Wysłano: ver#1' oper#MY_STATUS' status#READY' nvcp#OK' time#21.09.2020 14:48:22' data#marcin@marcin.net 9854416'
Odebrano: ver#1' oper#MY_STATUS' status#READY' nvcp#OK' time#21.09.2020 14:48:22' data#7312906'

Uwierzytelniono
Wysłano: ver#1' oper#MY_STATUS' status#BUSY' nvcp#OK' time#21.09.2020 14:48:22' data#marcin@marcin.net 7312906'
Odebrano: ver#1' oper#MY_STATUS' status#BUSY' nvcp#OK' time#21.09.2020 14:48:22' data#5840324'

Uwierzytelniono
Wysłano: ver#1' oper#REGISTER' status#NONE' ulp#OK' data#inny@marcin.net 123qwe'
```

Rysunek 10: Przebieg w programie testującym serwer

```
C:\Windows\system32\cmd.exe
Wysłano: ver#1' oper#REGISTER' status#NONE' ulp#OK' data#inny@marcin.net 123qwe'
Odebrano: ver#1' oper#REGISTER' status#SUCCESS' ulp#OK' data#''

Uwierzytelniono
Wysłano: ver#1' oper#LOGIN' status#NONE' ulp#OK' data#inny@marcin.net 123qwe'
Odebrano: ver#1' oper#LOGIN' status#SUCCESS' ulp#OK' data#1826232'

Uwierzytelniono
Wysłano: ver#1' oper#MY_STATUS' status#READY' nvcp#OK' time#21.09.2020 14:48:22' data#inny@marcin.net 1826232'
Odebrano: ver#1' oper#MY_STATUS' status#READY' nvcp#OK' time#21.09.2020 14:48:22' data#3582895'

Uwierzytelniono
Wysłano: ver#1' oper#AVABILITY' status#NONE' nvcp#OK' time#21.09.2020 14:48:22' data#inny@marcin.net marcin@marcin.net'
Odebrano: ver#1' oper#AVABILITY' status#BUSY' nvcp#OK' time#21.09.2020 14:48:22' data#''

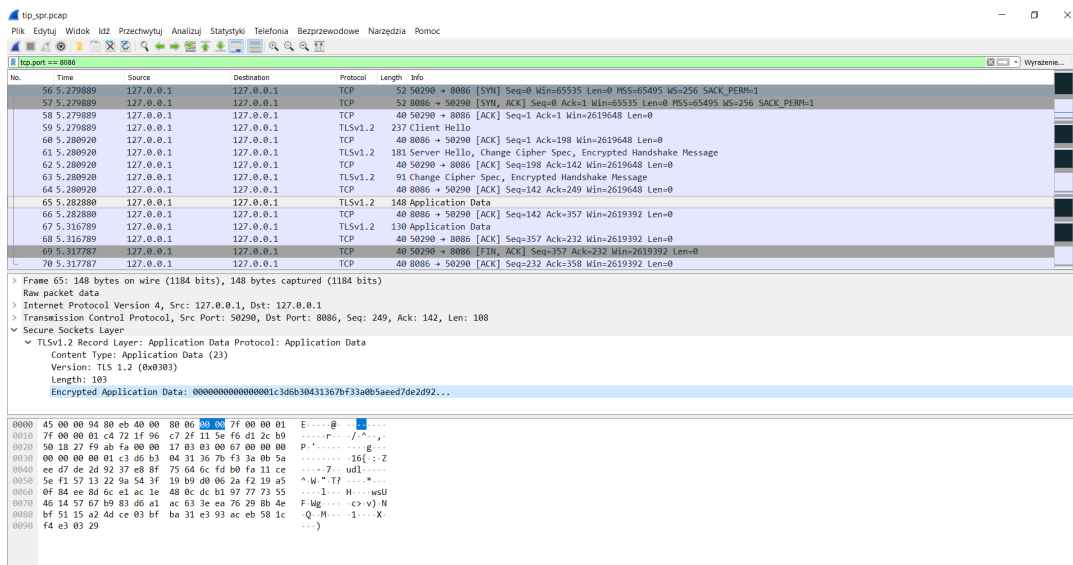
Uwierzytelniono
Wysłano: ver#1' oper#MY_STATUS' status#READY' nvcp#OK' time#21.09.2020 14:48:22' data#marcin@marcin.net 5840324'
Odebrano: ver#1' oper#MY_STATUS' status#READY' nvcp#OK' time#21.09.2020 14:48:22' data#1041386'

Uwierzytelniono
Wysłano: ver#1' oper#AVABILITY' status#NONE' nvcp#OK' time#21.09.2020 14:48:22' data#inny@marcin.net marcin@marcin.net'
Odebrano: ver#1' oper#AVABILITY' status#READY' nvcp#OK' time#21.09.2020 14:48:22' data#''

Uwierzytelniono
Wysłano: ver#1' oper#MY_STATUS' status#UNAVAILABLE' nvcp#OK' time#21.09.2020 14:48:22' data#marcin@marcin.net 1041386'
Odebrano: ver#1' oper#MY_STATUS' status#UNAVAILABLE' nvcp#OK' time#21.09.2020 14:48:22' data#2798048'

Uwierzytelniono
Wysłano: ver#1' oper#MY_STATUS' status#UNAVAILABLE' nvcp#OK' time#21.09.2020 14:48:22' data#inny@marcin.net 3582895'
Odebrano: ver#1' oper#MY_STATUS' status#UNAVAILABLE' nvcp#OK' time#21.09.2020 14:48:22' data#6027293'
```

Rysunek 11: Przebieg w programie testującym serwer



Rysunek 12: Fragment przebiegu w programie Wireshark

## 8 Podsumowanie

Udało się zrealizować projekt w kształcie opisanym w rozdziale 1. System umożliwia nagranie wiadomości głosowej dla innej osoby, do późniejszego odtworzenia.

### 8.1 Cele zrealizowane

Udało się zrealizować własne protokoły realizujące zadania obsługi użytkowników. Aplikacja klienta umożliwia nagrywanie i odsłuchanie wiadomości głosowych.

### 8.2 Problemy

Podczas tworzenia aplikacji napotkałem na trudności:

- problem z implementacją szyfrowanej komunikacji z serwerem przy pomocy protokołu SSL,
- problemy z działaniem biblioteki NAudio.