

POLITECHNIKA POZNAŃSKA

WYDZIAŁ INFORMATYKI I TELEKOMUNIKACJI

TELEFONIA IP

**Dokumentacja aplikacji
internetowego komunikatora
głosowego**

ANDRZEJ SKROBAK, 136310

ANDRZEJ.SKROBAK@STUDENT.PUT.POZNAN.PL

BARTOSZ POREBSKI, 136303

BARTOSZ.POREBSKI@STUDENT.PUT.POZNAN.PL

30 Września 2020



Spis treści

1	Cel i założenia projektowe	2
2	Architektura systemu	2
3	Wymagania	3
3.1	Funkcjonalne	3
3.2	Pozafunkcjonalne	4
4	Technologie	5

1 Cel i założenia projektowe

Celem projektu jest wykonanie aplikacji umożliwiającej nawiązanie rozmowy głosowej przez Internet pomiędzy dwoma aplikacjami klienckimi. Aplikacja ma spełniać kryteria bezpiecznego systemu informatycznego, to znaczy:

- Poufność - transmitowane dane będą szyfrowane.
- Autentyczność - ustanowienie sesji będzie wymagać od klientów uwierzytelnienia certyfikatem lub hasłem przez serwer.
- Niezaprzeczalność - serwer będzie prowadził rejestr połączeń, którego rekordy będą zawierać ID rozmówców oraz sygnaturę czasową utworzenia pokoju do rozmowy stanowiące dowód rozmowy.
- Dostępność - wirtualizacja środowiska pozwala na dynamiczne przypisywanie zasobów, co zapewni jakość dostarczania usługi.

2 Architektura systemu

System opiera się na architekturze klient-serwer (n:1). Rozmową określa się sesję, w której bierze udział dokładnie dwóch użytkowników. System oparty jest na standardowych protokołach wykorzystywanych w komunikacji internetowej np. IP (ang. *Internet Protocol*), UDP (ang. *User Datagram Protocol*). Komunikacja przy użyciu mechanizmu WebRTC, jest w całości szyfrowana i w transmisji pośredniczy serwer TURN (Traversal Using Relays around NAT), umożliwia to uwierzytelnienie użytkowników w sieci peer-to-peer i rozwiązuje problem NAT.

- Oparty na frameworku Django, serwuje użytkownikom aplikację sieciową umożliwiającą rejestrację i rezerwowanie wirtualnych pokoi do rozmów. Korzysta z lokalnej bazy danych SQLite.
- Serwer Node.js, umożliwia użytkownikom nawiązanie sesji, pośredniczy w początkowej fazie negocjacji, sygnalizuje stworzenie pokoju, podłączenie się do niego użytkownika i pośredniczy w wysyłaniu komunikatów SDP (Session Description Protocol), przy pomocy którego użytkownicy negocjują rozpoczęcie sesji.
- Serwer TURN, odróżnieniu do serwera STUN, pośredniczy w całości transmisji po nawiązaniu sesji przez użytkowników. Mechanizm uwierzytelniania klientów opiera się na długoterminowych kluczach, przechowywanych w lokalnej bazie danych.

Aplikacja kliencka składa się z dwóch serwisów:

- serwowany na porcie 8000 serwis umożliwiający rezerwację pokoi.
- serwowany na porcie 3000 serwis umożliwiający podanie danych uwierzytelniających i nawiązanie sesji z drugim użytkownikiem. Logika i komunikacja z serwerem opiera się na skrypcie NodeJS działającym w przeglądarce klienta.

3 Wymagania

Funkcje oraz wymagania pozwalające na prawidłowe działanie programu oraz określające kroki wymagane do nawiązania rozmowy.

3.1 Funkcjonalne

Wymagania co do funkcji jakie spełniać musi dany system

Po stronie serwera:

- Rejestracja użytkownika - obsługa rejestracji użytkowników poprzez tworzenie profili o określonych nazwach, adresach e-mail i hasłach, dodatkowo użytkownikowi zostaje nadany unikalny ośmio cyfrowy identyfikator.
- Zarządzanie bazą danych - obsługa danych zawartych w lokalnej bazie danych, aktualizacja uprawnień do pokoi.
- Uwierzytelnienie użytkownika - użytkownik podaje swój login i hasło w celu uwierzytelnienia.
- Pośredniczenie w transmisji - przesyłanie pakietów bez ingerencji w ich zawartość, pakiet musi zostać przetworzony aby poprawnie pośredniczyć w transmisji.
- Prowadzenie rejestru rozmów - zapisywanie faktu nawiązania połączenia pomiędzy użytkownikami.
- Zarządzanie użytkownikami i dostępami - administrator systemu ma możliwość usuwania użytkowników i ograniczania uprawnień.

Po stronie klienta:

- Rejestracja konta użytkownika - użytkownik podaje nazwę konta, adres e-mail i hasło w celu utworzenia konta zapewniającego dostęp do funkcjonalności serwisu.
- Zaproszenie użytkownika do wirtualnego pokoju - klient podaje identyfikator użytkownika, którego chce zaprosić do sesji.
- Dołączenie do wirtualnego pokoju - wysłanie danych umożliwiających uwierzytelnienie tożsamości poprzez podanie wygenerowanego identyfikatora pokoju, tokenu użytkownika i tokenu uwierzytelniającego.
- Zakończenie sesji - wysłanie do serwera komunikatu o prośbie zakończenia sesji.

3.2 Pozafunkcjonalne

- System operacyjny z zainstalowanym interpreterem Python w wersji 3.8 lub wyższej i dostępem do Internetu
- Zainstalowane urządzenie do nagrywania (mikrofon) oraz odtwarzania audio
- Interfejs użytkownika (graficzny lub konsolowy)
- Zszyfrowanie transmisji zgodne ze standardami DTLS (Datagram Transport Layer Security) określonymi w RFC 6347, 5238, 6083, 5764.

4 Technologie

Język: Python 3.8, NodeJS, HTML

Biblioteki: Django, SocketIO, WebRTC

Baza danych: SQLite

Środowisko wirtualizacyjne: Docker

Kontrola wersji: Git, GitHub

Dokumentacja: LaTeX

Link do wideoprezentacji: <https://drive.google.com/file/d/157Zv1Wh70REL-ZIZ42eX13R9xyAAiZXp/view?usp=sharing>

Link do projektu GitHub: https://github.com/SomeonePL/Web_video_chat_WebRTC