POLITECHNIKA POZNAŃSKA WYDZIAŁ ELEKTRYCZNY Instytut Automatyki, Robotyki i Inżynierii Informatycznej

Michał Ściborski

Bartosz Więckowski

Jakub Smierzchalski

Podstawy Teleinformatyki - dokumentacja projektu

Konferencje audio-wideo

12 czerwca 2019

Spis treści

1. Ogólne założenia projektu

Aplikacja służąca do prowadzenia konferencji audio-wideo (również z możliwością udostępniania ekranu) przez dwóch użytkowników, przy jak najmniejszym czasie początkowej konfiguracji z ich strony.

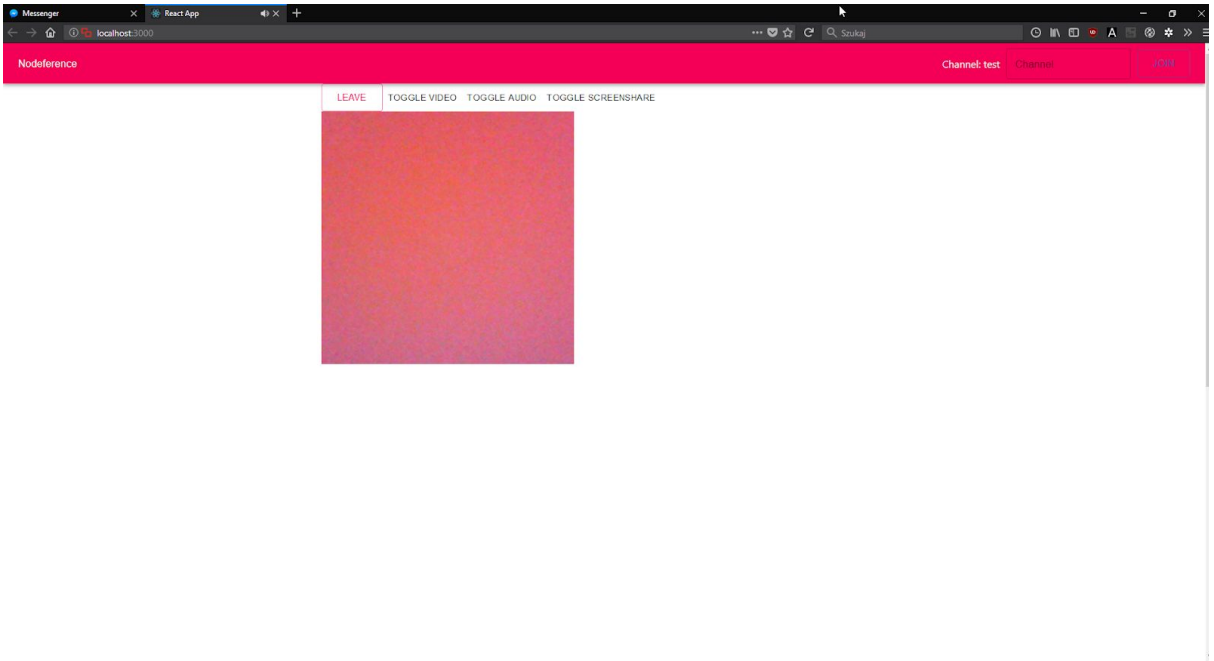
2. Podział prac

Z uwagi na tematykę projektu i dynamikę prowadzonych prac, nie zdecydowaliśmy się na jasny podział na zadania wykonywane od początku do końca przez danego członka zespołu. Większość prac była wykonywana wspólnie, poszczególne elementy poprawiane przez różne osoby, bezpośrednio zależne od siebie elementy tworzone przez kilka osób na raz. Jasne określenie podziału zadań było tym trudniejsze, że była to nasza pierwsza styczność z taką technologią, przynajmniej od strony developerskiej. W trakcie prowadzenia projektu, niejednokrotnie zmienialiśmy podejście do niektórych problemów, nieraz drastycznie. Z uwagi na powyższe, nie prezentujemy tu dokładnego podziału prac między członków zespołu.

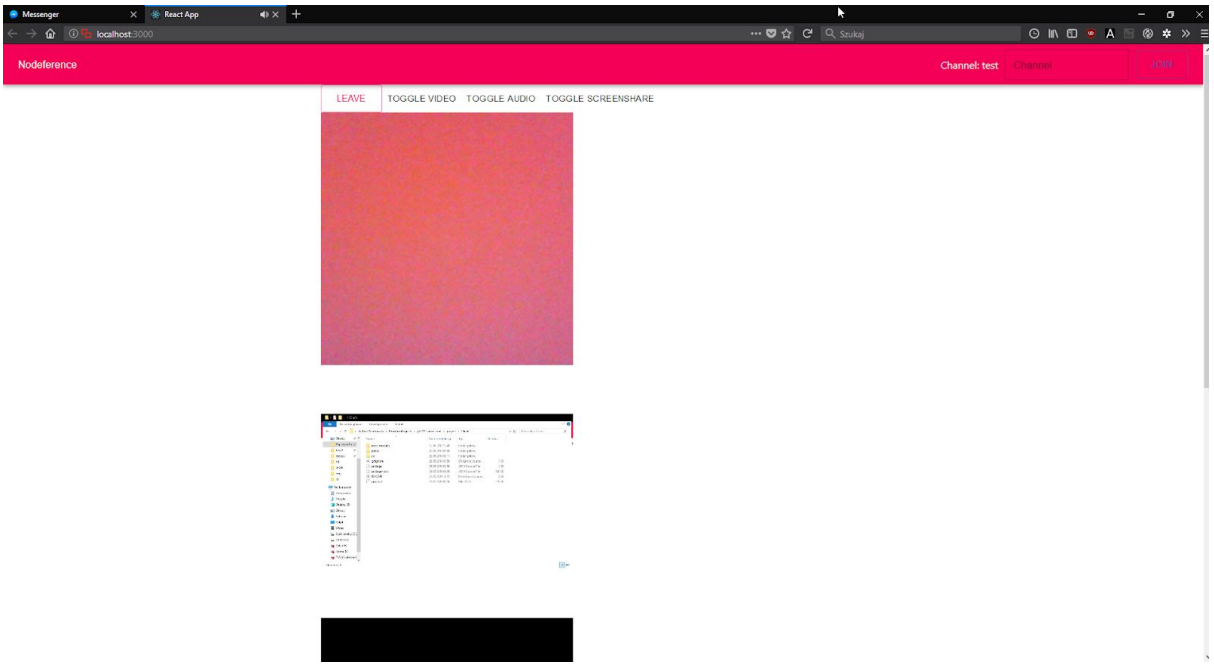
3. Opis funkcjonalności aplikacji

Aplikacja (docelowo dostępna online) umożliwia prowadzenie rozmów audio-wideo z funkcją udostępniania ekranu, przy praktycznie zerowym czasie konfiguracji wstępnej, bez żadnego logowania czy pobierania oprogramowania.

Rys. 1 Podstawowy widok aplikacji przed dołączeniem do kanału



Rys. 2 Podstawowy widok aplikacji przy jednym aktywnym użytkowniku



Rys. 3 Podstawowy widok aplikacji podczas udostępniania ekranu

4. Przebieg prac

Pierwotnie chcieliśmy zaimplementować streaming audio-wideo z wykorzystaniem Socket.io, Node.js i Simple-Peer, z uwagi na teoretyczną łatwość połączenia tych technologii i ich skuteczność w komunikacji w czasie rzeczywistym, jednak to rozwiązanie okazało się zbyt nieopłacalne, przez ilość czasu i pracy, którą musielibyśmy w nie zainwestować. Ostatecznie zdecydowaliśmy się wykorzystać Agora.io - płatną implementację WebRTC - oraz React.js.

5. Główne wykorzystane technologie

a. WebRTC

WebRTC, czyli komunikacja w sieci Web w czasie rzeczywistym, to otwarty standard promowany m.in. przez Google oraz Mozillę, który umożliwia komunikację w czasie rzeczywistym bez użycia wtyczek, korzystając z interfejsów JavaScript API. Dzięki temu możliwe jest wykonywanie połączeń głosowych, wideokonferencji oraz przesyłanie plików przy pomocy przeglądarki. WebRTC korzysta z serwera zwanego Web Conferencing Server, który wraz z Serwerem STUN ma za zadanie zapewnić stronę początkową oraz synchronizację połączeń pomiędzy dwoma punktami końcowymi WebRTC. Swoją przełomową szybkość, WebRTC zawdzięcza głównie bezpośredniemu połączeniu między klientami, przy użyciu przeglądarki. Niestety jest niekompatybilny z niektórymi przeglądarkami (np Safari), a do współpracy z niektórymi, potrzebuje zewnętrznych pluginów (np Edge). Dodatkowo przez fakt używania protokołu UDP, WebRTC nie nadaje się najlepiej do przesyłania plików, z uwagi na gubione pakiety.

b. Agora.io

Agora.io to płatna implementacja WebRTC, umożliwiająca proste i szybkie wykorzystanie technologii WebRTC bez dalszego marnowania czasu na, jak się okazało bardziej problematyczną, niż pierwotnie zakładaliśmy, implementację streamingu audio-wideo od podstaw.

6. Instrukcja użytkowania aplikacji

a. Pobranie projektu z repozytorium https://github.com/msciborski/pt2019-video-conf

b. Utworzenie w folderze project/Client/src pliku o nazwie *config.js* o treści

```
const config = {
    appId: 'bce20bf93f8749daa3f139e7623fc614',
};
export default config;
```

c. Otworzenie okna terminalu w folderze project/Client

d. Wpisanie poniższych komend:

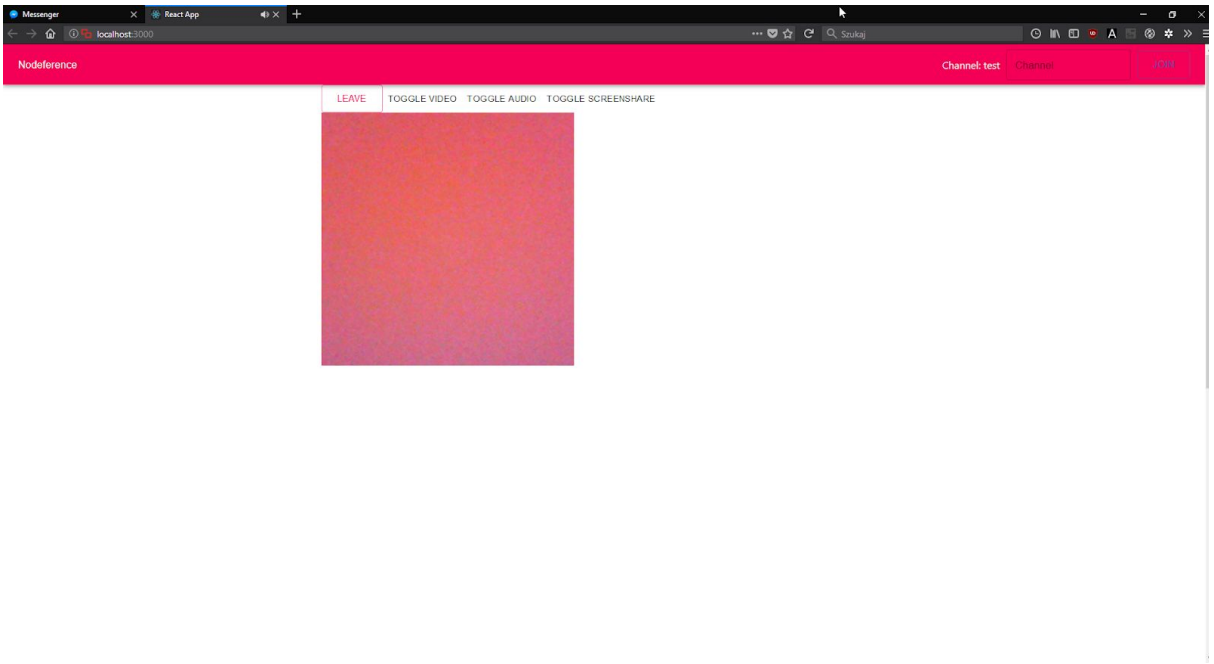
```
npm install
npm start
```

e. Wpisanie w oknie przeglądarki internetowej adresu *localhost:3000*

f. W widocznym w prawym górnym rogu polu *Channel* wpisujemy ID kanału, do którego chcemy dołączyć (np *1*), po czym naciskamy przycisk *JOIN*

Rys. 4 Podstawowy widok aplikacji przed dołączeniem do kanału

g. Uzyskujemy następujący widok. Kolejna osoba może dołączyć do naszej konferencji, wybierając ten sam kanał i naciskając przycisk *JOIN*. Teraz możemy dowolnie włączać/wyłączać wideo, audio oraz udostępnianie ekranu, przy pomocy odpowiednich przycisków nad widokiem z naszej kamery. Do opuszczenia pokoju służy przycisk *LEAVE*.



Rys. 5 Podstawowy widok aplikacji po dołączeniu do kanału