

Rzeszów, 19.06.2018r.

Politechnika Rzeszowska im. Ignacego Łukasiewicza

Wydział Elektrotechniki i Informatyki

Techniki multimedialne

Projekt

Komunikator P2P w sieci lokalnej

Opracował

Konrad Snopkowski

Spis treści

Spis treści	2
1. Założenia projektu	3
2. Wykonanie zadania projektowego	3
2.1. Aplikacja	3
2.1.1. Informacje podstawowe	3
2.1.2. Licencja	3
2.1.3. Changelog	3
2.1.4. Działanie aplikacji	3
2.2. Dokumentacja kodu	4
3. Wnioski	4

1. Założenia projektu

Według wstępnych założeń, aplikacja powstała w czasie realizacji projektu powinna spełniać następujące wymagania:

- Możliwość komunikacji dwóch dowolnych użytkowników będących jednocześnie w tej samej sieci lokalnej,
- Wyszukiwanie aktywnych użytkowników wewnątrz sieci,
- Zapisywanie stanu rozmowy i odtwarzanie w przypadku błędu.

2. Wykonanie zadania projektowego

2.1. Aplikacja

2.1.1. Informacje podstawowe

Aplikacja została wykonana w technologii JavaFX, korzystając jedynie z podstawowych pakietów dostępnych w JDK1.8.

2.1.2. Licencja

Aplikacja udostępniana jest na Powszechnej Licencji Publicznej GNU (GPLv3). Kod źródłowy dostępny jest w publicznym repozytorium kodu GitHub pod adresem <https://github.com/nenovike/LanTalk>.

2.1.3. Changelog

Changelog opiera się na historii wersji w repozytorium kodu:

23 kwietnia: wstępna wersja konsolowej aplikacji z podziałem na klienta i serwer. (3h)

25 kwietnia: połączenie klienta z serwerem, komunikacja dwukierunkowa. (2h)

16 maja: utworzenie okienkowej aplikacji w javafx, budowa głównego okna aplikacji, multicastowe wysyłanie pakietów „hello” (4h)

30 maja: synchronizacja kontaktów z wykorzystaniem multicastowo rozsyłanych pakietów „hello” (3h)

2 czerwca: uporządkowanie kodu, przepisanie klas serwerów socketów (4h)

17 czerwca: budowa listy użytkowników i okna rozmowy, działająca komunikacja w oknie rozmowy (5h)

19 czerwca: powiadomienie o nieprzeczytanej wiadomości, generacja dokumentacji javadoc (3h)

łącznie czas pracy: 24h.

2.1.4. Działanie aplikacji

Aplikacja pozwala na:

- Niezależną rozmowę z wieloma użytkownikami jednocześnie,
- Wybór użytkownika do rozmowy z automatycznie aktualizowanej listy aktywnych klientów,
- Wybór interfejsu sieciowego w przypadku przynależności do kilku sieci jednocześnie,
- Możliwość zmiany nazwy użytkownika.

2.2. Dokumentacja kodu

Dokumentacja kodu wykonana przy pomocy oprogramowania javadoc jest dostępna w repozytorium kodu.

3. Wnioski

Nie udało się wykonać części zakładanej funkcjonalności:

- Szyfrowanie komunikacji,
- Zapamiętywanie rozmowy pomiędzy uruchomieniami aplikacji,
- Uzupełnianie rozmowy po problemach z połączeniem.

Jest to głównie związane z brakiem czasu na wykonanie niektórych zadań. Komunikacja jest w pełni oparta na protokole UDP, co sprawia, że informacje są przesyłane jawnym tekstem, lecz próba wykorzystania protokołu TCP nie powiodła się z kilku przyczyn. Przede wszystkim wymagałoby to założenia osobnego wątku dla każdej rozmowy i przy większej liczbie użytkowników doszłoby do spadku wydajności i wzrostu zapotrzebowania na zasoby. Innym problemem jest brak możliwości nadawania z użyciem TCP w trybie innym niż unicast, co uniemożliwiło rozgłaszanie pakietów powitalnych.

Zapamiętywanie rozmów mogłoby być zaimplementowane z wykorzystaniem plików tymczasowych lub cache. Byłaby wtedy możliwość zapisu jawnego bądź zaszyfrowanego tekstu rozmowy w plikach lokalnych komputera użytkownika.

Uzupełnianie rozmowy po utracie połączenia nie jest łatwą funkcjonalnością do zaimplementowania, zważywszy na to, że wykorzystywany jest protokół UDP. Nie ma możliwości kontroli stanu połączenia, bardzo trudne byłoby śledzenie stanu pakietów. Wymagałoby to potwierdzenia dostarczenia każdego pakietu.