

Projekt JAVA
Politechnika Świętokrzyska w Kielcach
Wydział Elektrotechniki, Automatyki i Informatyki

Studia: Stacjonarne I stopnia

Kierunek: Informatyka

Data wykonania:
02.06.2020

**Grupa
dziekańska:**
2ID13B

Wykonał:
Mateusz Krząstek
Kamil Kasprzyk
Daniel Wojnowski

Temat projektu: Odtwarzacz muzyki

Projekt „odtwarzacza muzyki” został zbudowany przez nas w oparciu o technologię klient serwer. Założeniem, które udało nam się zrealizować było stworzenie aplikacji do przesyłania muzyki. W ten oto sposób po uruchomieniu naszej aplikacji serwer udostępnia utwory muzyczne, jest on naszym „szefem” natomiast klient odbiera je w sposób, w jaki zostały mu udostępnione. Ponadto zaprojektowaliśmy możliwość komunikacji tekstowej pomiędzy klientem a serwerem co jest naszą dodatkową funkcjonalnością w tym projekcie. Po stronie klienta jest również możliwość nadania dowolnej nazwy użytkownika oraz w dowolnej chwili użytkownik może łączyć się jak również rozłączać z serwerem. Więcej funkcjonalności posiada nasz serwer, który również ma możliwość łączenia się oraz rozłączania z klientami lecz ponad to może dodawać, usuwać, czyścić listę z piosenkami do odtworzenia. Serwer może również ładować gotowe playlisty oraz je zapisywać. W celu wizualizacji naszych dokonań w kodzie postanowiliśmy użyć plików FXML. Pliki FXML to nic innego jak XML z odpowiednimi znacznikami wewnątrz. Inicjalizacją obiektów naszego widoku, ustawieniem ich właściwości oraz stworzeniem zależności między nimi zajmuje się wówczas za nas klasa *FXMLoader*. Pozostała funkcjonalność została stworzona w Java-ie. Do pisania całej aplikacji użyliśmy narzędzia programistycznego IntelliJ.

Nasz projekt funkcjonuje zgodnie z początkowymi założeniami po uruchomieniu programu po stronie serwera posiadamy wszystkie niezbędne funkcjonalności to przesyłania muzyki. Po stronie klienta zostały zaprojektowane niezbędne funkcje do odbioru muzyki wysyłanej przez serwer.

Uruchomienie projektu oraz jego obsługa jest bardzo intuicyjna. Uruchamiamy po przez IntelliJ dwa osobne pliki w dwóch osobnych oknach *AppClient* oraz *AppServer*. Po ich uruchomieniu należy postępować wg własnego uznania i tego co chcemy słuchać. Możemy wybrać poprzednią playlistę i ją słuchać jak również istnieje możliwość stworzenia własnej klient musi słuchać utworów, które narzuca mu serwer.

Klasa *ClientApp* odpowiada za uruchomienie całej aplikacji po stronie klienta posiada metody do pobrania nicku od użytkownika, pobiera port na którym następuje połączenie z serwerem

Klasa *Protocol* jest klasą finalną posiada finalne statyczne metody odpowiadające za właściwe połączenie na właściwych portach. Po dokonaniu połączenia klienta z serwerem zwraca taką informację.

Interface *WritableGUI* - interfejs zapewniający, że każde nowe GUI będzie miało klasę piszącą na czacie

Klasa *MessageListener*- obsługuje transmisję Serwer -> Klient posiada kilka metod np. *stopPlayer* służąca do zatrzymania odtwarzania, *run* odpowiada za połączenie z serwerem, inicjalizacja obiektów, rozpoczęcie wątku audio

Klasa `AudioPlayerThread`- odpowiada za tworzenie nowych wątków jest to klasa dziedzicząca po `Thread`, znajduje się w niej przysłonięcie metody `run`

Po stronie serwera z uwagi na jego większą funkcjonalność znajduje się więcej klas.

Główną klasą jest tutaj `ServerApp` posiadająca kilka metod związanych z zarządzaniem playlistami.

Klasa `Song` posiada kilka finalnych metod związanych z odtwarzaniem muzyki jej przesyłaniem, czasem trwania, pobraniem ścieżki gdzie dany utwór się znajduje.

Klasa `TextWriter` służy do zapewniania komunikacji tekstowej między klientem a serwerem znajduję się w niej przysłonięcie metody `run`.

Podczas pracy nad projektem wszyscy członkowie zespołu w równy sposób dokładali swoją wiedzę oraz umiejętności programistyczne do tworzenia kodu źródłowego naszego programu, którym jest Odtwarzacz muzyki. Pomimo trudnych warunków związanych z pracą na odległość oraz komunikacją wyłącznie online udało nam się podzielić pracą w równym zakresie. Po wykonaniu pewnych etapów przez każdego z członków zespołu łączyliśmy efekty prac w jeden projekt. Po każdym etapie staraliśmy się łączyć na różnych platformach całym zespołem w celu omówienia wykonanych prac oraz wytyczenia dalszego etapu prac. W efekcie wspólnej pracy udało nam się wykonać w pełni działającą aplikację.