Cient and Server

Projekt aplikacji

Wykonał:

Krzysztof Agaś-Lange

215668

1. **Cel projektu**
   Celem projektu jest stworzenie aplikacji klient – server.
2. **Wymagania co do aplikacji.**
   A. Należy opracować aplikacje kliencka i serwerowa przy podanych poniżej założeniach:

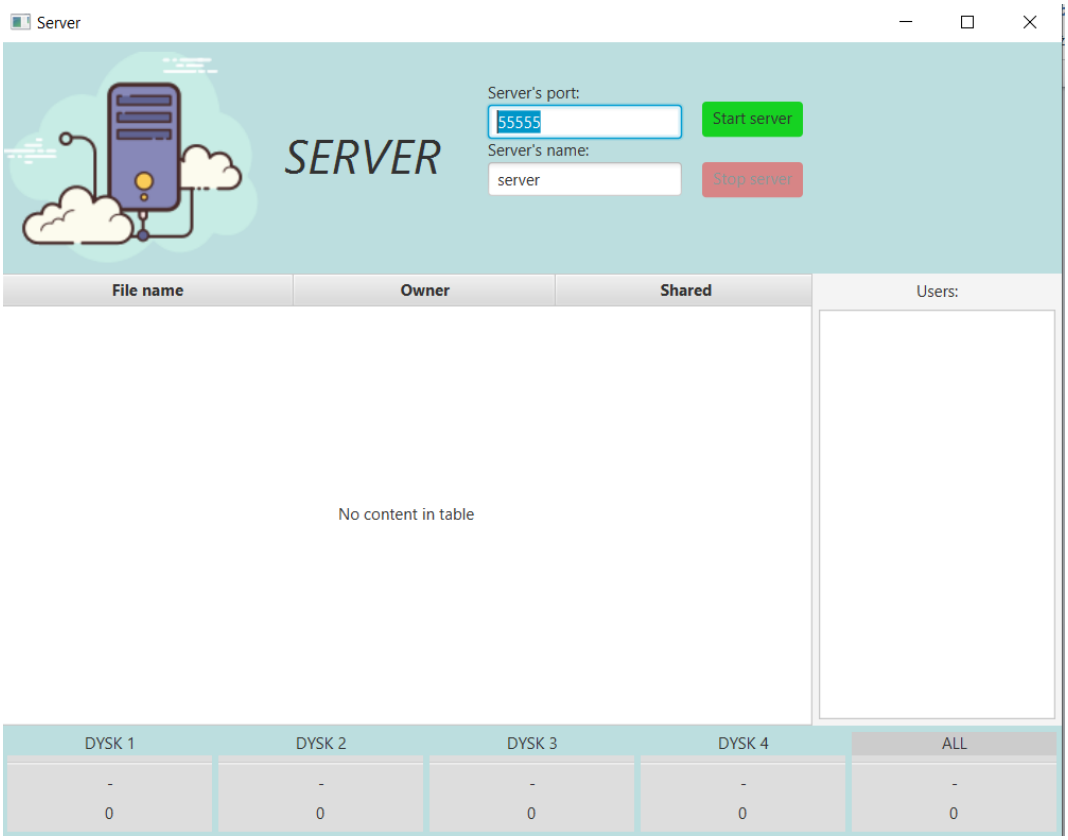
   B. Aplikacja kliencka: - Uruchamiana jest z dwoma parametrami: nazwa użytkownika i ścieżka do lokalnego folderu - Każdy klient ma swój lokalny folder z plikami - Aplikacja obserwuje lokalny folder i reaguje na zmiany. Jak pojawia sie tam nowe pliki, to wysyła je na serwer - Jak pojawi sie nowy plik dla danego użytkownika, to jest on pobierany do lokalnego folderu - Aplikacja po uruchomieniu odpytuje serwer o nowe pliki i je ściąga - Wysyłanie / odbieranie dzieje sie przy wykorzystaniu puli wątków - Aplikacja kliencka ma interfejs graficzny (np. Java FX) pokazujący w czasie rzeczywistym czym sie w danej chwili zajmuje klient ("Pobieram...", "Wysyłam ...", "Sprawdzam ....") oraz wyświetlający listę aktualnych plików w lokalnym folderze. Panel graficzny ma umożliwić także udostępnienie danego pliku innemu użytkownikowi. Listę dostępnych użytkowników należy pobrać z serwera.

   C. Serwer: - 5 folderów, które symulują 5 serwerów lub 5 dysków - Klient wysyła np. 10 plików, wiec serwer uruchamia ileś wątków na których równolegle kopiuje pliki do tych dysków (folderów) - Wymagany jest kontroler, który tak rozłoży ruch, ze do każdego z dysków (folderów) jednocześnie jest kopiowana taka sama liczba plików - Jeżeli podłączy sie drugi klient, który zacznie wysyłać pliki, nie może on czekać Az skończą sie zadania pierwszego klienta. Lista żądań na serwerze musi ulec reorganizacji, tak aby obydwaj klienci mieli wrażenie natychmiastowej obsługi (zaproponuj stosowny algorytm) - Na każdym dysku serwera znajduje sie plik tekstowy (np. csv), w którym jest opisana zawartość danego dysku i kto jest jego właścicielem. Zauważ, ze plik Bedzie uaktualniany przez wiele wątków. Rozwiąż ten problem. - W celu wizualizacji symulacji na niewielkiej liczbie użytkowników, czas kopiowania ma być sztucznie wydłużony poprzez usypianie wątku na losowa liczbę sekund. - Serwer posiada panel graficzny (np. Java FX) pokazujący zawartość 5ciu dysków (serwerów) oraz aktualnie wykonywane operacje.

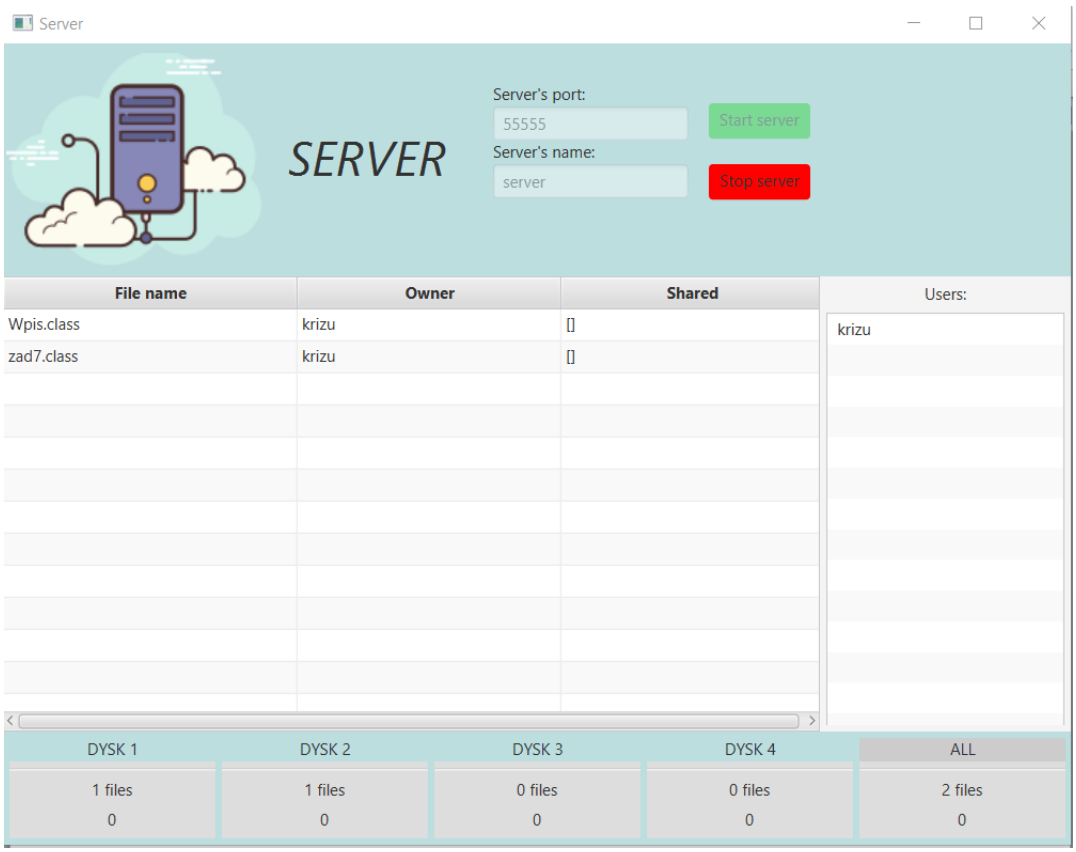
   D. Zaproponuj liste testow jednostkowych opracowanych w JUnit, osobno dla klienta i serwera obejmujacych kluczowe operacje wykonywane przez kazdego z nich.

3. **Server**

   Server jest włączany z dwoma parametrami do podania: **Portem servera** oraz **Nazwą servera**. Po wprowadzeniu danych i zatwierdzeniu ich przyciskiem **Stert server** włączany jest server.

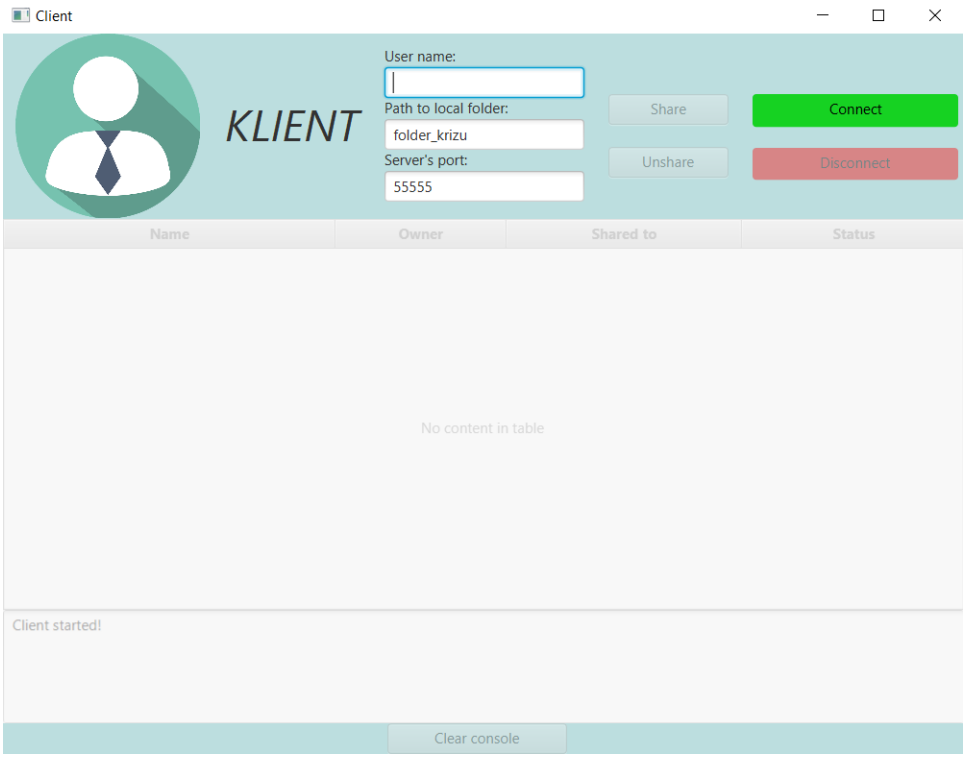


Pod główną tablicą można zobaczyć aktualnie zapisane pliki w serverze, ich właścicieli oraz do kogo są udostępnione. Po prawej stronie w tabeli **Users** można zobaczyć aktualnie połączonych użytkowników z serverem. Na dole znajduje się status 5 dysków oraz to, jaką ilość plików aktualnie posiadają i ile aktualnie jest do nich przesyłanych plików.



4. Klient

Po uruchomieniu aplikacji klienta, aplikacja prosi o podanie **User name** oraz **Path to local folder** oraz **Server's port**, które są już domyślnie wypełnione, ale można je zmienić. Przycisk Connect powoduje uruchomienie aplikacje wraz z podanymi parametrami.



Pod główną tablicą można zobaczyć pliki u danego użytkowniaka, kto jest ich właścicielem, do kogo są udostępnione oraz ich status wysłania. Na dole znajduje się konsola, na której pokazywane są najważniejsze informacje zwiazane z danymi akcjami. Za pomocą przycisków **Share** i **Unshare** można udostępnić plik innemu użytkownikowi jak i to cofnąć. **Przycisk Clear** console służy do czyszczenia konsoli.