

Laboratorium z przedmiotu Sieci komputerowe 2			
Sprawozdanie			
Temat projektu: <i>Komunikator internetowy</i>			
Prowadzący: <i>mgr inż. Adam Godziński</i>	Autorzy: Mikołaj Simon 145347 Daniel Szkopek 145216	Grupa dziekańska:	17.1
		Ocena:	

1. Opis projektu

1.1 Krótki opis

Celem projektu było stworzenie komunikatora internetowego. Aplikacja pozwala na komunikację między klientami. Komunikator pozwala na tworzenie konta, logowanie z hasłem, wyświetlenie znajomych, prowadzenie wielu konwersacji jednocześnie. Całość jest widoczna z poziomu graficznego interfejsu użytkownika.

1.2 Użyta technologia

- serwer został napisany w języku **C** z wykorzystaniem biblioteki **bsd-sockets**
- klient został napisany w języku **Python** przy wykorzystaniu bibliotek **Threading** oraz **socket**
- GUI jest stworzone jako okno dialogowe w prostej bibliotece **Tkinter** w języku Python

2. Opis komunikacji pomiędzy serwerem i klientem

W pierwszej kolejności, aby możliwa była komunikacja między użytkownikami muszą oni się zarejestrować. Klient wysyła do serwera polecenie "rejestracja" wraz z nazwą użytkownika i hasłem pobranych z formularza interfejsu. Serwer dokonuje walidacji nowego użytkownika, sprawdzając czy dany użytkownik już istnieje. Po poprawnej walidacji serwer tworzy konto i informuje klienta, że rejestracja przebiegła pomyślnie.

Następnie, użytkownik loguje się poprzez wysłanie polecenia "logowanie" wraz z nazwą użytkownika i hasłem pobranym z formularza interfejsu. Serwer dokonuje autoryzacji danych użytkownika i jeśli jest ona poprawna, to odsyła odpowiedź, dzięki której użytkownik "wejdzie" do systemu. Serwer oznacza użytkownika jako zalogowany.

Po zalogowaniu użytkownik ma możliwość otwarcia chatu (jeśli posiada znajomych) lub dodania nowego użytkownika do znajomych. Dodanie użytkownika następuje po odebraniu przez serwer polecenia "dod_znaj" wraz z nazwą dodawanego użytkownika. W przypadku gdy użytkownik nie istnieje serwer nie doda znajomego.

Po otwarciu chatu ze znajomym klient uruchamia dwa wątki: odbierania i wysyłania wiadomości. Wątek odbierania co sekundę wysyła do serwera polecenie "reload" w celu odświeżenia historii chatu. Wątek wysyłania wiadomości przesyła do serwera polecenie "wysylanie" wraz z treścią wiadomości i jej adresatem. Po odebraniu serwer aktualizuje historię wysyłanych wiadomości i odsyła odpowiedź o odebraniu wiadomości.

3. Podsumowanie

Serwer w momencie uruchomienia tworzy cały system, który inicjuje działanie aplikacji. Od tego momentu klienci mogą łączyć się z komunikatorem wykorzystując adres serwera oraz przypisany port. Dane przechowywane są w strukturach serwera, użytkownicy nie mają do nich dostępu z poziomu ich aplikacji. Po zerwaniu połączenia użytkownik zostaje automatycznie wylogowany, a wątek z nim związany zostaje zniszczony. Wraz z wyłączeniem serwera zostaną utracone wszystkie dane aktualnej sesji. Dane między serwerem a klientem wysyłane są w konwencji plików JSON w celu ułatwienia odczytu danych.

Największą trudność sprawiła nam implementacja serwera w języku C oraz obsługiwanie w nim wielu wątków. Mieliśmy również trochę problemów z odpowiednim przesyłaniem danych o obiektach użytkowników i ich wiadomości, gdyż były one skomplikowane. Dobrym rozwiązaniem okazało się być ich przesyłanie w formacie JSON. Mankamentem naszej aplikacji jest toporne GUI napisane w bibliotece Tkinter. Okazuje się, że każdy system wyświetla okna dialogowe na swój własny nieprzewidziany sposób.