

# Projekt zaliczeniowy – Sieci Komputerowe 2 – projekt komunikator typu GG

Wykonali: Kacper Grzelak 154042 oraz Bartosz Korszun 154030

## 1) Protokół komunikacyjny

komunikacja oparta jest na wymianie między klientem a serwerem zaprojektowanych przez nas „pakietów” składających się z nagłówka (typu pakietu), dodatkowych pól oraz treści. Liczba pól uzależniona jest od typu komunikatu. Wszystkie pola są oddzielone znakiem „\n” a komunikat kończony jest znakiem „\n\n”

## 2) Sposób implementacji

- a. Serwer korzysta z języka C oraz biblioteki SQLite do obsługi bazy danych. W kodzie zdefiniowane są stałe reprezentujące różne rodzaje komunikatów, struktury danych do przechowywania informacji o użytkownikach oraz funkcje do obsługi operacji na bazie danych i zarządzania listą zalogowanych użytkowników.

W głównej funkcji main, serwer inicjalizuje gniazdo, nasłuchuje na połączenia i dla każdego klienta tworzy nowy wątek do obsługi komunikacji. Wątek obsługi klienta (cthread) odbiera komunikaty od klienta, identyfikuje ich rodzaj na podstawie kodu komunikatu, a następnie wykonuje odpowiednie akcje.

Obsługiwane komunikaty obejmują logowanie, wysyłanie wiadomości, pobieranie listy użytkowników, wylogowywanie, sprawdzanie statusu użytkownika, sprawdzanie istnienia użytkownika, itp.

Serwer uruchamiany jest na adresie IN\_ADDRANY oraz porcie 1234.

- b. Kod klienta jest zaimplementowany przy użyciu modułów `tkinter` i `socket` do obsługi interfejsu graficznego i komunikacji sieciowej. Struktura obejmuje klasy `Message` i `Chat` do reprezentacji wiadomości i czatu. Proces logowania i inicjalizacji interfejsu użytkownika jest obsługiwany w funkcji `main()`, gdzie tworzone są także wątki do odbierania wiadomości i monitorowania kolejki komunikatów systemowych. Kod korzysta z blokad wątkowych, takich jak `threading.Lock`, do synchronizacji dostępu do struktur danych. Interfejs graficzny obejmuje okno główne z listą kontaktów, przyciskiem do rozpoczęcia nowego czatu, oraz obszar czatu z możliwością wysyłania wiadomości.

do zalogowania potrzebny jest adres ip serwera (port jest wpisany na stałe w kodzie)

3 podstawowe konta stworzone w aplikacji podane w formacie login, hasło

- i. 123456789, admin

- ii. 987654321, user1

- iii. 123123231, user2

## 3) Sposób Kompilacji

- a. Serwer – do skompilowania serwera wymagane jest zainstalowanie sqlite3 oraz libsqlite3-dev na komputerze.

do skompilowania wystarczy użycie komendy:

```
`gcc serv.c -lpthread -lsqlite3`
```

następnie uruchomienie pliku wykonywalnego

- b. Klient – istnieją dwie możliwości uruchomienia klienta:

- i. Uruchomienie klienta z pliku wykonywalnego – rozpakowujemy archiwum zawarte w kodzie i przechodzimy do folderu `dist` a następnie uruchamiamy `client.exe`

- ii. Uruchomienie klienta przez interpreter pythona – potrzebujemy pythona w wersji co najmniej 3.10 oraz pobranych paczek `tkinter` oraz `ttkbootstrap`. Jeżeli mamy wszystkie wymagane paczki przechodzimy do folderu `client\_gui` i uruchamiamy klienta komendą:

```
`python client.py`
```