

Sprawozdanie sk2

Temat 7- komunikator internetowy

1. Opis projektu

reComm - komunikator internetowy w architekturze klient-serwer. Funkcjonalności: rejestracja/logowanie z hashowaniem haseł i JWT, system znajomych (zaproszenia, akceptacja/odrzucenie), wiadomości prywatne, grupy konwersacyjne (bez limitu użytkowników), powiadomienia real-time, graficzny interfejs desktopowy, trwałe przechowywanie danych.

Technologie:

Backend (C++20) **Frontend (Python 3.x)**

TCP sockets (POSIX) PyQt6

JSON (nlohmann/json) TCP Client z auto-reconnect

JWT (jwt-cpp) QThread (powiadomienia)

CMake 3.28+ Kolejki dla async I/O

Architektura Backend: Repository Pattern, Service Layer, Dependency Injection, Hexagonal Architecture, DDD.

Architektura Frontend: warstwa komunikacji (TCPClient, ApiService), warstwa prezentacji (okna, dialogi, widgety), cache lokalne z timerami QTimer.

2. Opis komunikacji pomiędzy serwerem i klientem

Protokół: TCP + JSON (newline-delimited), port 8080.

Request: {"method": "NAZWA", "body": {...}, "token": "JWT"}

Response: {"code": 200, "message": "...", "data": {...}}

Metody API:

Autoryzacja: REGISTER, AUTH

Znajomi: SEND/ACCEPT/REJECT_FRIEND_REQUEST, GET_FRIENDS,

GET_PENDING_REQUESTS

Grupy: CREATE_GROUP, ADD_MEMBER_TO_GROUP, UPDATE_GROUP_NAME, LEAVE/DELETE_GROUP, GET_USER_GROUPS, GET_GROUP_DETAILS/MEMBERS

Wiadomości: SEND/GET_GROUP_MESSAGE, SEND/GET_PRIVATE_MESSAGE

Powiadomienia push: trwałe połączenie TCP, ConnectionManager (userId → socket), typy:

NEW_FRIEND_REQUEST, NEW_GROUP_MESSAGE, NEW_PRIVATE_MESSAGE. Offline → kolejka.

3. Podsumowanie

Kluczowe aspekty techniczne:

- **Backend:** wielowątkowość (każdy klient osobny wątek), thread-safety (mutex'y), trwałe połączenia TCP dla push notifications
- **Frontend:** NotificationWorker (QThread), PyQt signals, auto-reconnect z exponential backoff, cache z wykrywaniem zmian

Trudności:

1. **Wielowątkowość** → mutex'y, separacja odpowiedzialności
2. **Trwale połączenia TCP** → auto-reconnect, heartbeat, obsługa błędów
3. **Błędy biznesowe** → hierarchia wyjątków (already_friends_error, group_not_found_error)
4. **Migracja Tkinter→PyQt6** → Tkinter wymagał python3-tk na openSUSE i powodował problemy, PyQt6 było łatwo instalowane przez pip + lepsze wsparcie wielowątkowości

Wnioski: Projekt dostarczył doświadczenia w programowaniu sieciowym, wielowątkowości C++/Python, integracji komponentów wielojęzycznych oraz architekturze warstowej. Największe wyzwanie: zapewnienie bezpieczeństwa wątkowego przy dostępie do współdzielonych zasobów.