Mateusz Puk 155864
Magdalena Hotaj 155991

Sprawozdanie sk2

Temat: Komunikator internetowy

1. Opis projektu (0.5 strony)

- Projekt polega na stworzeniu komunikatora internetowego umożliwiającego rejestrację i logowanie użytkowników, dodawanie znajomych oraz prowadzenie rozmów indywidualnych i grupowych. Użytkownicy mogą wysyłać wiadomości tekstowe do znajomych oraz do grup, a historia rozmów jest przechowywana w bazie danych. Komunikacja pomiędzy klientami jest obsługiwana za pomocą serwera wykorzystującego architekturę klient-serwer. Klienci łączą się do serwera poprzez protokół TCP. Serwer obsługuje komunikację za pomocą gniazd BSD w środowisku GNU/Linux.
- Projekt składa się z kilku modułów zaimplementowanych w różnych technologiach:
  - **Serwer:** Język C, biblioteki: *pthread* do wielowątkowości, *sys/socket.h*, *netinet/in.h*, *arpa/inet.h* do komunikacji sieciowej, *sqlite3* do obsługi bazy danych, , *cJSON* do formatowania wiadomości.
  - **Klient:** Język Python, biblioteka *socket* do komunikacji sieciowej.
  - **GUI:** Tkinter (Python) do stworzenia interfejsu graficznego.
  - **Baza danych:** SQLite, przechowująca dane użytkowników, znajomości, wiadomości i grupy.

2. Opis komunikacji pomiędzy serwerem i klientem (0.5 strony, może być schemat/rysunek)

   Komunikacja pomiędzy klientem a serwerem odbywa się za pomocą protokołu TCP/IP. Klient inicjuje połączenie z serwerem, podając jego adres IP oraz numer portu 1100. Po nawiązaniu połączenia serwer akceptuje klienta i tworzy osobny wątek do jego obsługi. Klient wysyła zapytania w formie tekstowej, na przykład *LOG;username;password* w celu logowania lub *MSG;sender;receiver;message* do wysyłania wiadomości. Serwer odbiera dane, analizuje je i wykonuje odpowiednie operacje na bazie danych SQLite, takie jak weryfikacja użytkownika, zapis wiadomości lub pobranie historii rozmów. W przypadku poprawnej weryfikacji serwer odsyła komunikat potwierdzający, a w przeciwnym razie wysyła komunikat o niepowodzeniu.
   W przypadku pobierania danych, klient może wysłać zapytanie *GET_USER_GROUPS;username*, na co serwer odpowiada listą znajomych w formacie JSON. Analogicznie, polecenie *GET_MESSAGES;user1;user2* pozwala uzyskać historię rozmów między dwoma użytkownikami. Wiadomości mogą być również wysyłane do grup poprzez *CREATE_GROUP;group_name*, co powoduje utworzenie nowej grupy w bazie danych i zwrócenie potwierdzenia do klienta. Po zakończeniu komunikacji klient może zamknąć połączenie, a serwer odpowiednio zarządza zamknięciem sesji.

3. Podsumowanie (0.5-1 strona)

- Najważniejsze informacje o implementacji

- Projekt obejmuje kompletne rozwiązanie komunikatora z obsługą dodawania znajomych i tworzenia grup.
- Wykorzystano prostą bazę danych SQLite do przechowywania danych użytkowników, wiadomości i relacji.
- Wielowątkowość na serwerze pozwala obsługiwać wielu użytkowników jednocześnie.
- Klient posiada przyjazny interfejs graficzny, ułatwiający zarządzanie kontaktami i wiadomościami.
- Implementacja obejmuje mechanizmy obsługi błędów oraz zabezpieczenia przed błędnym użyciem systemu.

- Co sprawiło trudność
  - Obsługa formatowania wiadomości i zapewnienie ich prawidłowego odbioru oraz dekodowania JSON w komunikacji między klientem a serwerem.
  - Zapewnienie poprawnego działania komunikacji w warunkach dużego ruchu użytkowników.

  Mimo tych wyzwań udało się stworzyć funkcjonalną aplikację pozwalającą na skuteczną komunikację między użytkownikami. Projekt może być rozwijany o dodatkowe funkcje, takie jak szyfrowanie wiadomości, wsparcie dla załączników oraz aplikację mobilną.