

Uniwersytet Kardynała Stefana Wyszyńskiego

Wydział Matematyczno-Przyrodniczy

Szkoła Nauk Ścisłych

Projekt MiniTwitter „X”

Autorzy:

- Patryk Mizerski
- Jakub Żak
- Mateusz Tucki
- Paweł Romańczuk

Spis treści:

1. Wstęp.
2. Funkcjonalność.
3. Diagram Gantt.
4. Korki pośrednie.
5. Diagram sekwencji.

1. Wstęp

W projekcie X celem jest stworzenie rozproszonej aplikacji działającej na klientach oraz serwerze, która operuje na zasadach podobnych do Twittera. Klient wysyła do serwera wiadomości o maksymalnej długości 80 znaków oraz może prosić o wyświetlenie ostatnich n wiadomości. Projekt ten wykorzystuje Protocol Buffers i gRPC do komunikacji między klientem a serwerem, zapewniając efektywną transmisję danych oraz interakcję za pomocą interfejsu zdalnego wywołania procedury (RPC).

2. Funkcjonalność

Kluczowe aspekty, które oferuje projekt X:

1. Rejestracja i logowanie użytkowników:

- Użytkownicy mogą się rejestrować i logować, co zapewnia bezpieczeństwo i personalizację usług.

2. Wysyłanie wiadomości:

- Klienci mogą wysyłać wiadomości o maksymalnej długości 80 znaków z własnych urządzeń do serwera, który odpowiednio je przechowuje.

3. Pobieranie wiadomości:

- Klienci mogą żądać od serwera wyświetlenia ostatnich n wiadomości, które są przechowywane w określonej kolejności.

4. Lajkowanie i komentowanie wiadomości:

- Użytkownicy mogą lajkować oraz komentować wiadomości innych użytkowników.

5. Śledzenie użytkowników:

- Użytkownicy mogą śledzić innych użytkowników, co pozwala im na bieżąco obserwować aktywność śledzonych osób.

6. Edytowanie i usuwanie wiadomości:

- Użytkownicy mogą edytować oraz usuwać swoje wiadomości.

7. Zarządzanie kontem:

- Użytkownicy mogą zmieniać swoje hasło oraz usuwać konto z możliwością usunięcia wszystkich swoich wiadomości lub ich pozostawienia.

8. Efektywna komunikacja:

- Dzięki wykorzystaniu Protocol Buffers i gRPC, komunikacja między klientami a serwerem jest zoptymalizowana, co pozwala na szybką transmisję danych.

9. Interfejs wiersza poleceń:

- Interakcja z klientem odbywa się za pomocą intuicyjnego interfejsu wiersza poleceń, co sprawia, że korzystanie z aplikacji jest łatwe i przyjazne dla użytkownika.

3. Diagram Gantt

| | Patryk Mizerski | Jakub Żak | Mateusz Tucki | Paweł Romańczuk | | |
|------------|--|-----------|---------------|-----------------|--|--|
| Tydzień 1 | 1 godzina | | | | | |
| Tydzień 2 | 2 godziny | | | | | |
| Tydzień 3 | 2 godziny | | | | | |
| Tydzień 4 | 10 godzin | 5 godzin | 5 godzin | 5 godzin | | |
| Tydzień 5 | 8 godzin | 5 godzin | 5 godzin | 5 godzin | | |
| Tydzień 6 | 8 godzin | 5 godzin | 5 godzin | 5 godzin | | |
| Tydzień 7 | 6 godzin | 5 godzin | 5 godzin | 5 godzin | | |
| Tydzień 8 | 11 godzin | 4 godziny | 2 godziny | 2 godziny | | |
| Tydzień 9 | 12 godzin | 2 godziny | 4 godziny | 2 godziny | | |
| Tydzień 10 | 16 godzin | 2 godziny | 2 godziny | 4 godziny | | |
| Tydzień 11 | 3 godziny | 1 godzina | 1 godzina | 1 godzina | | |
| LEGENDA | | | | | | |
| | Wybór projektu. | | | | | |
| | Przegląd projektu. | | | | | |
| | Określenie obowiązków, zaplanowanie następnych spotkań oraz zoorganizowanie pracy. | | | | | |
| | Projektowanie i kodowanie. | | | | | |
| | Tworzenie dokumentacji. | | | | | |
| | Sprawdzanie działania oraz dodawanie nowych funkcji. | | | | | |
| | Prezentacja. | | | | | |

4. Kroki pośrednie

1. Utworzenie repozytorium GitHub.
2. Utworzenie bazy danych do przechowywania wiadomości.
3. Utworzenie frameworka gRPC do komunikacji serwer–klient.
4. Napisanie w gRPC kodu proto, który służy do komunikacji serwer–klient.
5. Wygenerowanie pliku proto.
6. Utworzenie serwera.
7. Napisanie kodu serwera i połączenie go z bazą danych.
8. Sprawdzenie działania kodu serwera.
9. Napisanie kodu klienta.
10. Testowanie kodu klienta.
11. Testowanie połączenia serwera z klientem.
12. Poprawienie błędów i optymalizacja kodu.

5. Diagram sekwencji

