Baza danych medium społecznościowego

Łukasz Fabia 272724 Mikołaj Kubś 272662 Martyna Łopianiak 272682 Piotr Schubert 272659 Projektowanie baz danych wt 18:55

11 listopada 2024

Spis treści

L	Etap 1: Faza konceptualna 1			
	1.1	Analiz	za świata rzeczywistego	1
		1.1.1	Streszczenie - Zarys wymagań projektu	1
		1.1.2	Potrzeby informacyjne	1
			Czynności, wyszukiwania	
			Cele projektu	
		1.1.5	Zakres projektu	2
		Wymagania funkcjonalne		
	1.3	ERD		3
2 Etap 2: Faza logiczna		3		
3	Etap 3: Faza fizyczna			4
4	Faza	a Fizyo	czna cd.	5

1 Etap 1: Faza konceptualna

1.1 Analiza świata rzeczywistego

1.1.1 Streszczenie - Zarys wymagań projektu

Celem projektu jest stworzenie bazy danych do obsługi medium społecznościowego. Ma ona przechowywać informacje o użytkownikach, ich treściach, relacjach i aktywnościach. Baza powinna być zaprojektowana w sposób wydajny i skalowalny.

1.1.2 Potrzeby informacyjne

- Rejestracja i logowanie użytkowników z różnymi poziomami dostępu.
- Publikowanie i interakcje z treściami użytkowników (posty, polubienia, komentarze).
- Zarządzanie relacjami społecznymi (znajomości).
- Przechowywanie wiadomości prywatnych i historii aktywności.
- Tworzenie konwersacji z innymi użytkownikami
- Reportowanie postów z nieodpowiednimi treśćmi
- Tworzenie i zarządzanie stronami organizacji, firm, fanpage itd.
- Tworzenie i zarządzanie wydarzeniami

1.1.3 Czynności, wyszukiwania

- Wyszukiwanie użytkowników.
- Wyszukiwanie treści po hasztagach lub słowach kluczowych.
- Filtrowanie aktywności użytkownika, np. przeglądanie polubień i komentarzy.
- Wyszukiwanie relacji (np. znajomi użytkownika, osoby obserwujące daną osobę).
- Dodawanie innych użytkowników do znajomych i interakcja z treściami dodawanie komentarzy i reakcji
- Tworzenie postów, wydarzeń, grup
- Konwersacja grupowa, pisanie wiadomości

1.1.4 Cele projektu

S (Specific): Zaprojektowanie bazy danych dla medium społecznościowego.

M (Measurable): Baza musi być wydajna, tzn. musi być w stanie obsługiwać dużą ilość użytkow-

ników, co najmniej 20 000.

A (Achievable): Projekt zostanie zrealizowany przy użyciu PostgreSQL. Do stworzenia struktu-

ry tabel wykorzystany zostanie mechanizm ORM (Object Relational Mapping). Na koniec baza zostanie wypełniona danymi, aby przetestować jej wydajność.

R (Relevant): Przechowywanie profili użytkowników oraz interakcje między nimi są kluczowe

dla funkcjonowania medium społecznościowego.

T (Time-bound): Praca nad projektem powinna zająć 2 miesiące.

1.1.5 Zakres projektu

Multimedia: W bazie przechowywane będą wyłącznie linki do plików na ze-

wnętrznym serwerze.

Obsługa haseł: Wszystkie hasła w bazie będą hashowane.

1.2 Wymagania funkcjonalne

Streszczenie

Użytkownikom przypisany jest jeden z tych poziomów dostępu: admin, user lub guest.

Guest(Gość)

• Może przeglądać wybrane dane.

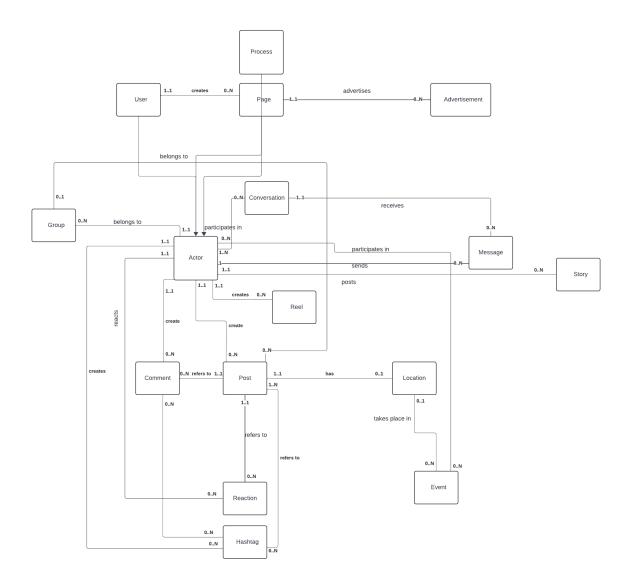
Admin

• Może przeglądać, edytować, usuwać, dodawać i przeglądać wszystkie treści, zrządza bazą i nadaje uprawnienia

User(Użytkownik)

- System umożliwia rejestrację oraz logowanie.
- Rejestracja wymaga imienia, nazwiska, daty urodzenia, hasła oraz maila.
- Logowanie wymaga maila i hasła.

1.3 ERD



Rysunek 1: Diagram obiektowo-relacyjny

2 Etap 2: Faza logiczna

Do tabel zostały dodane atrybuty, tabele **Page** oraz **User** zostały uogólnione przez **Author**, który bierze udział w innych czynnościach. Baza została także sprowadzona do *III postaci normalniej*, przez co wydzielono klika nowych tabel.



Rysunek 2: Diagram relacji

3 Etap 3: Faza fizyczna

Encje, które uległy zmianie:

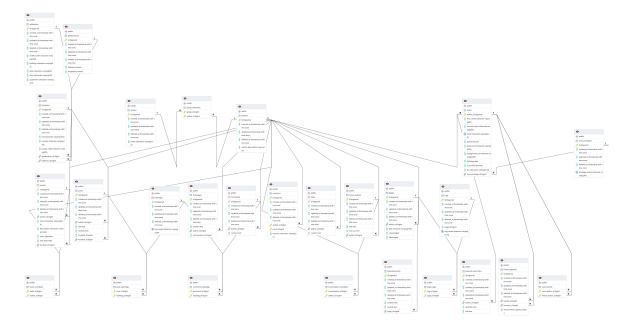
- Rozbicie lokalizacji na pomniejsze tabele
- Zamiana na enumy: typu autora oraz statusu zaproszenia do znajomych
- Dodanie do encji Conversation pole Members, które przechowuje id użytkowników biorących udział w konwersacji

Wykorzystano instrukcję CHECK do potwierdzenia poprawności danych w paru encjach. Przykładowo:

- 'A' nie moze być przyjacielem 'A'
- 'A' nie moze wysłać zaproszenia do znajomych do 'A'
- Czas rozpoczęcia wydarzenia musi być wcześniejszy niż czas zakończenia wydarzenia
- Grupa musi mieć od 1 do 10000000 członków.

Do stworzenia struktury bazy wykorzystano mechanizm ORM - (gorm). Sama baza wymagała dopracowania jeśli chodzi o enumeracje oraz reguły usuwania już bezpośrednio w systemie PostgreSQL (pgadmin).

Usuwanie: Każdy model posiada pole *DeletedAt* z indeksem. Podczas usuwania danego wiersza pole *DeletedAt* jest ustawiane na znacznik czasu (timestamp) wskazujący moment, w którym dane zostały usunięte. Dzięki temu baza obsługuje soft deleting. Oznacza to, że gdy użytkownik usunie swoje konto, będzie można je przywrócić. Taką operację obsługują na przykład serwisy takie jak Facebook.



Rysunek 3: Diagram relacji z PostgreSQL

4 Faza Fizyczna cd.

Standardowo kod można zobaczyć w repoztorium na gicie: social media db.

Mając napisany ORM poprzenio, w tym etapie pozostało napisać funkcje generujące dane do bazy. Rozwiązanie można podzielić na 3 cześci (od szczegółu do ogółu):

- 1. Dekorator, który będzie wykonywać okreśolny blok count razy.
- 2. Funkcja, generująca dany typ danych np. generator użytkowników.
- 3. Odpowienie ułożenie wywołań.

Zapełnianie bazy danych dużą ilością danych zajmuje stosunkowo dużo czasu, może to być spowodowane przez unikalność niektórych atrybutów(biblioteka "męczy" się z generowaniem unikalnych sensownych danych), ale także przez wywołania które tworzą listy autorów dla np. eventów, ostatnim podejrzeniem może być nieoptymalnie napisay kod.

Język: Go

Generowanie sztucznych danych: gofakeit Object relational mapping (ORM): gorm

Reszta rzeczy, które zostały wykonane:

- Funkcja usuwająca dane z tabel
- Własne implementacje niektórych danych np. Title, Birthday

- Konfiguracja loggera
- Funkcja haszująca hasło