

a) ExpressApp

Opis: Reprezentuje główną aplikację serwerową opartą na frameworku Express.js.

Atrybuty: Brak (klasa skupia się na metodach).

Metody:

+trackEndpoint(): void – Obsługuje endpoint /track, który rejestruje dane o wizytach.

+statsAPI(): void – Obsługuje endpoint /api/stats, dostarczający statystyki.

+serveDashboard(): void – Służy do wyświetlania interfejsu frontendowego (np. stats.html).

Rola: Punkt wejścia dla żądań HTTP, integruje się z innymi klasami.

b) Session

Opis: Reprezentuje sesję użytkownika, która grupuje wizyty.

Atrybuty:

-session\_id: String – Unikalny identyfikator sesji.

-start\_time: DateTime – Czas rozpoczęcia sesji.

-end\_time: DateTime – Czas zakończenia sesji.

Metody:

+trackSession(): void – Metoda do śledzenia sesji (np. aktualizacja czasu).

Rola: Łączy wiele wizyt w jedną sesję użytkownika.

c) Visit

Opis: Reprezentuje pojedynczą wizytę na stronie internetowej.

Atrybuty:

-page\_url: String – Adres URL odwiedzonej strony.

-referrer: String – Źródło odwiedzin (np. inny adres URL).

-user\_agent: String – Nagłówek user-agent przeglądarki.

-ip\_address: String – Adres IP użytkownika.

-browser: String – Nazwa przeglądarki (np. Chrome).

-os: String – System operacyjny (np. Windows).

-device: String – Model urządzenia (np. Desktop).

-country: String – Kraj użytkownika.

-is\_entry\_page: boolean – Czy strona jest stroną wejściową sesji.

-is\_exit\_page: boolean – Czy strona jest stroną wyjściową sesji.

Metody:

+trackVisit(): void – Rejestruje nową wizytę w systemie.

+updateExitStatus(): void – Aktualizuje status strony jako wyjściowej.

Rola: Główna encja przechowująca dane o wizytach.

d) StatsService

Opis: Usługa odpowiedzialna za generowanie statystyk na podstawie danych wizyt.

Atrybuty: Brak (klasa skupia się na metodach).

Metody:

+getStats(): Object – Zwraca ogólne statystyki (np. liczba wizyt).

+getTopPages(): Array – Zwraca 5 najpopularniejszych stron.

+getReferrers(): Array – Zwraca 5 głównych źródeł odwiedzin.

+getBrowsers(): Array – Zwraca statystyki przeglądarek.

+getEntryPages(): Array – Zwraca 5 najczęstszych stron wejściowych.

+getExitPages(): Array – Zwraca 5 najczęstszych stron wyjściowych.

+getUserPaths(): Array – Zwraca ścieżki użytkowników (np. 5 ostatnich sesji).

Rola: Analizuje dane z bazy i dostarcza statystyki przez API.

e) Database

Opis: Abstrakcja bazy danych (np. SQLite), która przechowuje dane wizyt.

Atrybuty: Brak (klasa skupia się na metodach).

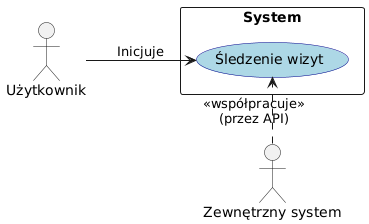
Metody:

+run(query: String, params: Array): void – Wykonuje zapytanie modyfikujące dane (np. INSERT, UPDATE).

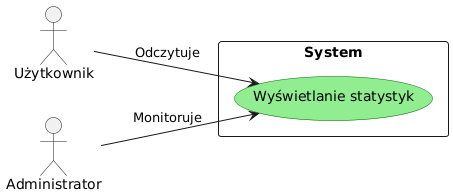
+get(query: String, params: Array): Object – Pobiera jeden rekord.

+all(query: String, params: Array): Array – Pobiera wiele rekordów.

Rola: Persystencja danych, obsługa zapytań SQL.

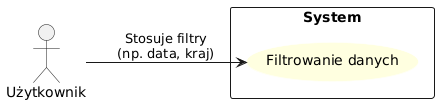


Opis:

* Aktorzy :
  + Użytkownik : Osoba, która odwiedza stronę internetową i inicjuje śledzenie wizyt.
  + Zewnętrzny system : System zewnętrzny (np. Google Analytics) może dostarczać dodatkowe dane do śledzenia wizyt.
* Przypadek użycia :
  + Śledzenie wizyt : Rejestrowanie danych o wizytach użytkowników, takich jak adres IP, przeglądarka, urządzenie, kraj itp.
  + Relacja **<<współpracuje>>** wskazuje integrację zewnętrznego systemu z głównym systemem śledzenia wizyt.

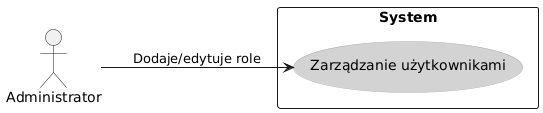
Opis:

* Aktorzy :
  + Użytkownik : Wyświetla podstawowe statystyki dotyczące ruchu na stronie.
  + Administrator : Ma dostęp do bardziej szczegółowych statystyk i monitoruje ogólną wydajność systemu.
* Przypadek użycia :
  + Wyświetlanie statystyk : Funkcjonalność umożliwiająca przeglądanie danych, takich jak liczba odwiedzin, najczęstsze strony, źródła ruchu itp.
  + Administrator ma większe uprawnienia do analizy danych niż zwykły użytkownik.



Opis:

* Aktorzy :
  + Użytkownik : Osoba, która chce wyświetlić statystyki według określonych kryteriów.
* Przypadek użycia :
  + Filtrowanie danych : Pozwala użytkownikowi stosować filtry, takie jak zakres dat, kraj pochodzenia, typ urządzenia lub przeglądarka, aby uzyskać bardziej precyzyjne wyniki.



Opis:

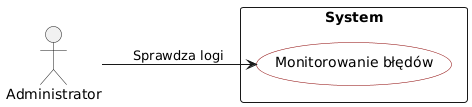
* Aktorzy :
  + Administrator : Osoba odpowiedzialna za zarządzanie kontami użytkowników w systemie.
* Przypadek użycia :
  + Zarządzanie użytkownikami : Funkcjonalność pozwalająca administratorowi dodawać nowych użytkowników, edytować ich role lub usuwać konta.

Obraz zawierający tekst, zrzut ekranu, linia, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Opis:

* Aktorzy :
  + Zewnętrzny system : System zewnętrzny, który współdziała z głównym systemem poprzez wymianę danych.
* Przypadek użycia :
  + Integracja z API zewnętrznym : Funkcjonalność umożliwiająca wymianę danych między głównym systemem a zewnętrznymi API, np. do uzupełniania danych o wizytach.



Opis:

* Aktorzy :
  + Administrator : Osoba odpowiedzialna za monitorowanie działania systemu i wykrywanie błędów.
* Przypadek użycia :
  + Monitorowanie błędów : Funkcjonalność pozwalająca administratorowi przeglądać logi systemowe, aby identyfikować i rozwiązywać problemy techniczne.