

Architektura oprogramowania - projekt

Patryk Matuszewski

Wiktor Kuchta

Patryk Prokopski

10 stycznia 2025

System Wyszukiwania Lotów

Projekt jest systemem umożliwiającym wyszukiwanie dostępnych połączeń lotniczych na podstawie wybranych lotnisk wylotu i przylotu oraz daty. Wyszukiwanie odbywa się w oparciu o bazę danych zawierającą szczegóły lotów, takie jak linie lotnicze, numery lotów, lotniska wylotu i przylotu, godziny wylotu i przylotu, oraz dzień tygodnia. System oblicza także przewidywany czas lotu na podstawie współrzędnych geograficznych lotnisk.

1 Zakres projektu

1.1 Wyszukiwarka lotów

Użytkownicy mogą wyszukiwać dostępne połączenia lotnicze, podając:

- Lotnisko wylotu,
- Lotnisko przylotu,
- Datę wylotu.

1.2 Panel administratora

Panel administracyjny umożliwia:

- Dodawanie nowych lotów do bazy danych,
- Usuwanie istniejących lotów na podstawie numeru lotu,
- Zabezpieczenie hasłem dla dostępu do panelu administracyjnego.

1.3 Baza danych

Baza danych zawiera dane o lotach, takie jak:

- Linia lotnicza,
- Numer lotu,
- Lotnisko wylotu i przylotu,
- Dzień tygodnia (numer dnia),
- Godzina wylotu,
- Czas trwania lotu.

1.4 Algorytm obliczania odległości i czasu lotu

System korzysta z geolokalizacji, aby obliczyć odległość między lotniskami na podstawie ich współrzędnych geograficznych. Na tej podstawie system wylicza szacunkowy czas trwania lotu.

2 Technologie użyte w projekcie

2.1 Flask

Flask jest lekkim frameworkiem webowym w języku Python, który obsługuje backend aplikacji. Używamy go do obsługi zapytań HTTP, renderowania stron HTML i zarządzania logiką aplikacji.

2.2 SQLite

Jako system zarządzania bazą danych używamy SQLite, który przechowuje dane dotyczące lotów. Jest to lekka, relacyjna baza danych, dobrze pasująca do aplikacji o mniejszej skali, jak nasz projekt.

2.3 HTML/CSS (Bootstrap)

Frontend aplikacji został zbudowany z użyciem HTML i CSS, gdzie w celu zwiększenia estetyki strony, zastosowano bibliotekę Bootstrap do tworzenia responsywnych i atrakcyjnych interfejsów użytkownika.

2.4 OpenCage Geocoding API

W celu obliczenia odległości pomiędzy lotniskami, projekt korzysta z API OpenCage Geocoding, które zapewnia możliwość przekształcania nazw lotnisk na współrzędne geograficzne (szerokość i długość geograficzną). Na podstawie tych współrzędnych obliczamy odległość między dwoma punktami na Ziemi za pomocą wzoru Haversine.

2.5 Wzór Haversine

Do obliczenia odległości pomiędzy dwoma punktami na Ziemi, zastosowano wzór Haversine, który uwzględnia krzywiznę Ziemi i zapewnia dokładność obliczeń dla małych i średnich odległości.

2.6 Flash Messages (Flask)

Do wyświetlania komunikatów o statusie operacji (np. dodanie lotu, usunięcie lotu), wykorzystano mechanizm flash messages w Flasku, który pozwala na przekazywanie informacji między stronami.

3 Architektura

Architektura tej aplikacji opiera się na modelu klient-serwer z wykorzystaniem Flask jako frameworku serwerowego, bazy danych SQLite do przechowywania informacji o lotach oraz zewnętrznych API OpenCage API do pobierania współrzędnych lotnisk.

3.1 Łatwość w implementacji i utrzymaniu

- Flask jest lekkim frameworkiem, który pozwala szybko budować aplikacje internetowe. Jest wystarczająco elastyczny, by dostosować się do potrzeb projektu, i nie wprowadza nadmiarowych komplikacji.
- SQLite to lekkostrawna baza danych, idealna do aplikacji, które nie potrzebują pełnej skalowalności w przypadku dużych ilości danych, a jej konfiguracja i integracja są szybkie i proste.

3.2 Modularność

- Aplikacja jest podzielona na moduły, takie jak obsługa zapytań o loty, obliczenia odległości czy integracja z API. Można łatwo modyfikować lub dodawać nowe funkcje, zachowując przejrzystość kodu.
- Użycie API OpenCage do rozszerzenia funkcjonalności aplikacji (np. pobieranie współrzędnych geograficznych, stref czasowych) sprawia, że aplikacja jest bardziej elastyczna i ma dostęp do bogatych zasobów zewnętrznych.

4 Atrybuty jakościowe

4.1 Dostępność

Aplikacja jest dostępna przez interfejs WWW, gdzie użytkownicy mogą wprowadzać zapytania dotyczące dostępnych lotów. Zastosowanie Flask sprawia, że aplikacja jest lekka i szybka.

4.2 Dostosowalność

Zadaliśmy o możliwość modyfikacji oprogramowania po jego wdrożeniu - możliwość dodawania nowych funkcji, dodania nowych API lub zmiany struktury bazy danych sprawia, że aplikacja jest dostosowalna do zmieniających się potrzeb.

4.3 Bezpieczeństwo

Wprowadzenie prostego systemu autoryzacji admina z hasłem zapewnia bezpieczeństwo dla części aplikacji związanej z dodawaniem/usuwaniem lotów.

4.4 Skalowalność

Aplikacja bazuje na SQLite (bazie lokalnej), ale możliwe jest łatwe przeniesienie jej na bardziej zaawansowaną bazę danych (np. PostgreSQL, MySQL) w razie potrzeby skalowania.

5 Cykl biznesowy

5.1 Dodawanie nowych lotów

- Administrator loguje się do panelu administracyjnego (weryfikacja hasła).
- Po zalogowaniu może dodać nowe połączenie lotnicze, wpisując szczegóły (np. linia lotnicza, numer lotu, lotnisko odlotu, lotnisko przylotu, dzień tygodnia, godzina wylotu).
- Dane są zapisywane w bazie danych, co umożliwia ich późniejsze wyszukiwanie.

5.2 Wyszukiwanie lotów

- Użytkownik wprowadza lotnisko wylotu, lotnisko przylotu oraz datę.
- Aplikacja wyszukuje dostępne loty w bazie danych na podstawie wprowadzonego dnia tygodnia.
- Dla każdego lotu obliczany jest czas lotu i szacowany czas przylotu, a następnie wyświetlany na stronie wyników.

5.3 Usuwanie lotów

- Administrator ma możliwość usunięcia lotu z bazy danych za pomocą panelu administracyjnego, co polega na wyszukaniu numeru lotu i jego usunięciu z systemu.