

**Wydział:** Fizyki i Informatyki Stosowanej

**Kierunek:** Informatyka Stosowana

**Rok:** 2022/23

**Semestr:** zimowy

**Typ:** stacjonarne

**Nr albumu:** 401984

**Data:** 20.11.2022



## Dokumentacja Projektu Flight Planner Przetwarzanie danych w chmurach obliczeniowych

### Spis treści

1	Opis projektu, koncepcji i założeń	2
2	Krótki opis architektury	3
3	Zastosowane narzędzia i technologie	3
4	Reprezentacja grafowa	4
5	Prezentacja działania aplikacji	5
6	Wdrożenie aplikacji	8
7	Ewentualne usprawnienia możliwe do wykonania	8
8	Literatura	9

opracował:  
Tomasz Szkaradek

# 1 Opis projektu, koncepcji i założeń

Tematem projektu jest "Fight Planner" który umożliwia użytkownikowi zarządzanie swoimi podróżami, pozwala też na organizację infrastruktury lotnisk na świecie. Aplikacja zawiera opcje do znajdowania najkrótszej ścieżki między skomunikowanymi lotniskami albo ścieżkę z najmniejszą liczbą przesiadek.

Lista możliwości, jakie zapewnia aplikacja/API:

- Dodanie nowego lotniska do bazy
- Edycja wybranego lotniska
- Usunięcie danego lotniska z bazy
- Dodanie połączenia z wybranym lotniskiem
- Wyszukanie lotniska po wybranym parametrze
- Otrzymanie listy wszystkich dostępnych lotnisk
- Wyszukanie połączenia po nazwie miasta
- Otrzymanie listy wszystkich połączeń
- Wyszukanie najkrótszej ścieżki pomiędzy lotniskami
- Wyszukanie ścieżki z najmniejszą liczbą węzłów pomiędzy lotniskami

The screenshot shows the 'Airports' management interface. At the top, there's a navigation bar with 'FLIGHT-PLANNER', 'Home', 'Airport', 'Search', and 'Sign Out'. Below the navigation bar, the 'Airports' section has a 'Filter' dropdown set to 'Country'. A list of countries (England, Italy, South Korea) is shown. Below this is a form titled 'Add connection' with input fields for 'Country', 'City', and 'Distance', and a 'Submit' button. To the right of the form is a table of existing connections with columns for 'Country', 'City', and a delete icon.

Country	City	Distance
England	London	
Italy	Rome	
South Korea	Seoul	
Italy	Milan	

(a) Formularz dodawania połączenia

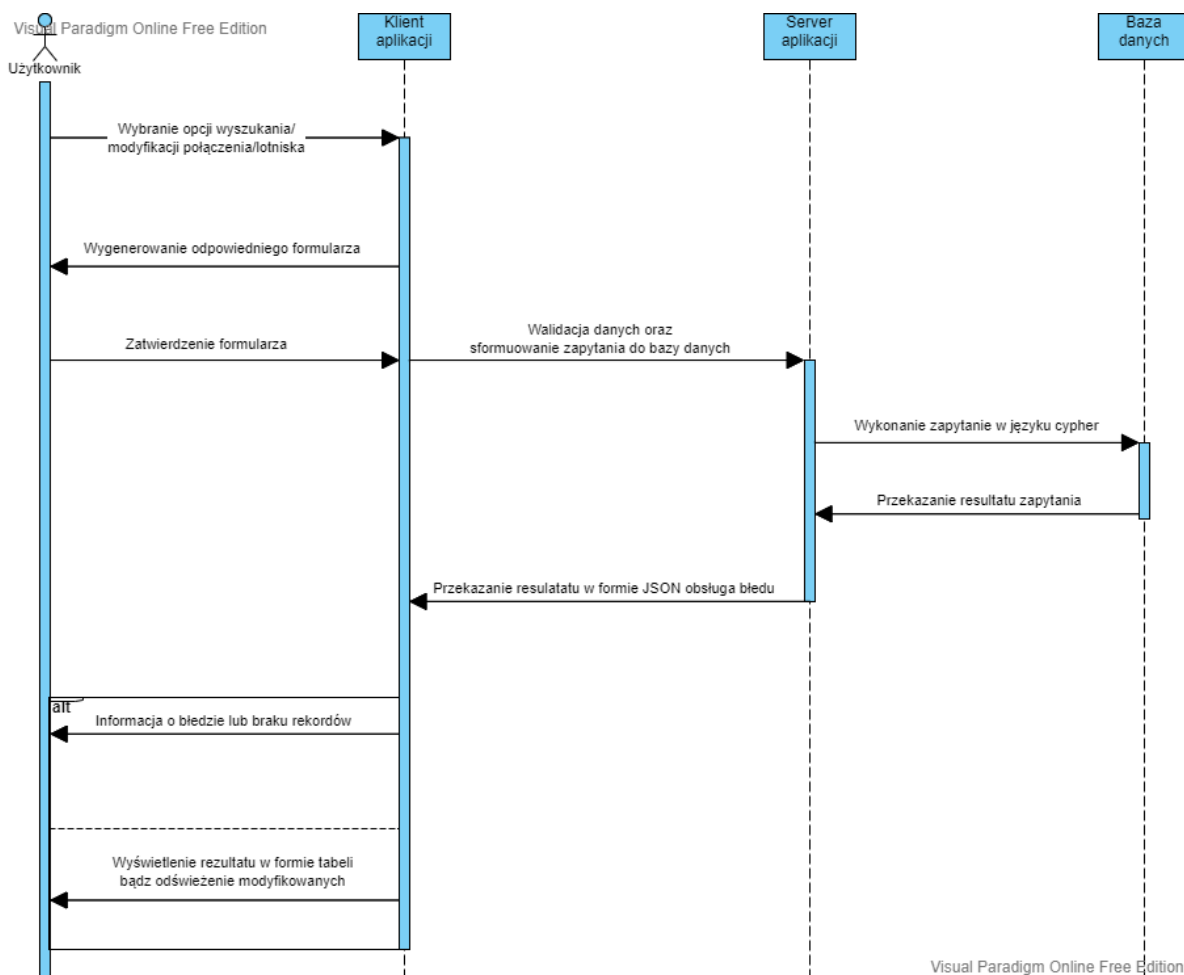
The screenshot shows the 'Flights' section. It has a 'Connections' tab and a 'Find path' button. Below these are input fields for 'City Out' (Warsaw) and 'City In' (Bogota). A table below shows the results of the search, with columns for ID, Country Out, City out, Country In, City In, and Distance.

ID	Country Out	City out	Country In	City In	Distance
36	Poland	Warsaw	USA	Atlanta	600
20	USA	Atlanta	USA	Dallas	100
6	USA	Dallas	Mexico	Mexico City	200
7	Mexico	Mexico City	Columbia	Bogota	200

(b) Najkrótsza ścieżka między Warszawą a Bogotą

## 2 Krótki opis architektury

Aby zaprezentować przepływ danych w aplikacji skonstruowany został diagram sekwencji



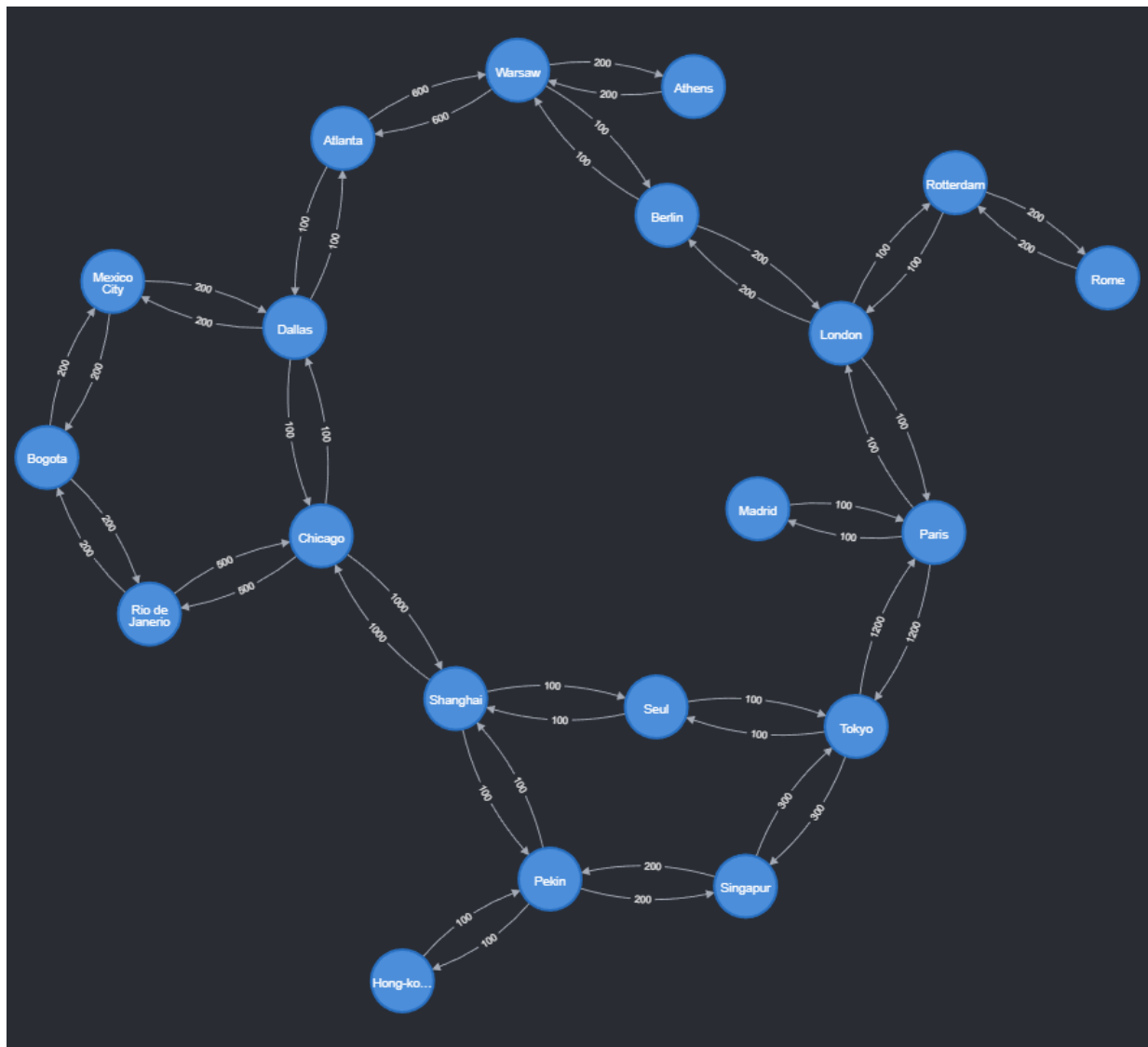
Rysunek 2: Diagram sekwencji funkcjonalności

## 3 Zastosowane narzędzia i technologie

Aplikacja została podzielona na część frontendową napisaną w języku JavaScript wykorzystując technologie React oraz bibliotekę Bootstrap. Natomiast w części backendowej aplikacji również wykorzystano język JavaScript oraz bibliotekę express z ich pomocą stworzono API zdolne zapisywać dane przesłane jako żądanie POST w postaci JSON-a oraz obsługiwać żądania takie jak GET/PUT/DELETE. Baza danych przechowywana jest na chmurowym rozwiązaniu Aura DB dla Neo4j, a zapytania do niej konstruowane są przy pomocy dedykowanego języka Cypher.

## 4 Reprezentacja grafowa

Na poniższym zdjęciu możemy zobaczyć strukturę grafową z postaci grafu skierowanego naszej bazy danych. Wierzchołkami w tym grafie są poszczególne lotniska natomiast występujące między nimi krawędzie/relacje to połączenie skierowane (skierowanie wskazuje lotnisko startowe i lotnisko docelowe)



Rysunek 3: Reprezentacja grafowa lotnisk i połączeń

## 5 Prezentacja działania aplikacji

Projekt dostarcza dwie podstawowe funkcjonalności z poziomu interfejsu użytkownika:

- Zarządzanie lotniskami
- Zarządzanie/wyszukiwanie optymalnych połączeń

Na początku musimy zarejestrować się do naszej aplikacji. Wypełniwszy formularz wysyłany jest kod aktywacyjny na podany mail, a następnie tworzone jest nasze konto, do którego możemy się zalogować.

The screenshot shows the 'Sign In' form of the 'FLIGHT-PLANNER' application. The header includes the logo, navigation links for Home, Sign In, and Sign Up, and a user profile icon. The form itself is titled 'Sign In' and contains two input fields: 'Email:' and 'Password:'. Below these fields is a red 'Sign In' button. At the bottom of the form, there is a link that says 'Need an Account? Sign Up'.

Rysunek 4: Formularz logowania

The screenshot shows the 'Register' form of the 'FLIGHT-PLANNER' application. The header is identical to the previous screenshot. The form is titled 'Register' and includes the instruction 'Please fill this form to create an account'. It contains several input fields with validation checkmarks: 'First Name:', 'Last Name:', 'Email:', 'Password:', and 'Confirm Password:'. The 'Email' field is pre-filled with 'email@student.ogh.edu.pl'. Below the 'Confirm Password' field is a red 'Next' button. At the bottom, there is a link that says 'Already registered? Sign In'.

Rysunek 5: Formularz rejestracyjny

Na górze ekranu znajduje się pasek nawigacji. Z jego poziomu, za pomocą rozwijanej opcji 'Add' można dodać nowe lotnisko. Opcja z kategorii 'Search' zwraca listę połączeń między lotniskami oraz dostarcza możliwości znalezienia najkrótszej ścieżki pomiędzy określonymi miastami lub ścieżki z najmniejszą liczbą przesiadek. W zakładce 'Add' – > 'Manage Airports' znajduje się lista wszystkich lotnisk z możliwością filtrowania po określonym kraju. Po kliknięciu odpowiedniej opcji ukazuje się nam 3 typy formularzy służące one do dodawania nowego lotniska jego edycji oraz dodania połączenia z innym lotniskiem. Posiadamy też możliwość usunięcia danego lotniska razem ze wszystkimi jego połączeniami.

(a) Formularz dodawania lotniska

(b) Formularz edycji lotniska

(c) Formularz dodawania połączenia

Tabela w zakładce "Connections" zwraca nam wszystkie połączenia pomiędzy miastami aplikacja zapewnia nam możliwość filtrowania i sortowania rekordów.

Tabela widoku "Find Path" ma jednakową formę dla każdego zapytania zwraca listę lot-

Connections					
ID	Country out	City out	Country in	City in	Distance
search (88) ...	search (88) ...	search (88) ...	search (88) ...	search (88) ...	search (88) ...
26	Germany	Berlin	England	London	200
24	Netherlands	Rotterdam	England	London	100
23	France	Paris	England	London	100
14	Germany	Berlin	England	London	200
12	Netherlands	Rotterdam	England	London	100
11	France	Paris	England	London	100
25	Italy	Rome	Netherlands	Rotterdam	200
24	England	London	Netherlands	Rotterdam	100
13	Italy	Rome	Netherlands	Rotterdam	200
12	England	London	Netherlands	Rotterdam	100

Rysunek 7: Tabela ze wszystkimi połączeniami

nisk, spełniające zadany warunek, w formie tabeli. Dla każdego lotniska zwracana jest id połączenia, nazwa kraju oraz miasta lotniska startowego oraz kraju i miasta celu podróży. Podana jest też estymowany dystans pomiędzy lotniskami.

Pierwszym wpisem jest lotnisko początkowe, natomiast kolejnymi są następujące po sobie przystanki, prowadzące do celu podróży. Ostatnim wierszem w tabeli jest wybrany cel podróży

Find path					
ID	Country Out	City out	Country In	City in	Distance
36	Poland	Warsaw	USA	Atlanta	600
20	USA	Atlanta	USA	Dallas	100
6	USA	Dallas	Mexico	Mexico City	200
7	Mexico	Mexico City	Columbia	Bogota	200

Rysunek 8: Tabela z wyszukaną najkrótszą ścieżką z Warszawy do Bogoty

Pliki w folderze `src/routes/` obsługują następujące adresy(endpointy):

- **/airport (GET)** Otrzymanie wszystkich lotnisk
- **/airport/collect (POST)** Otrzymanie lotniska po wybranym parametrze
- **/airport (POST)** Dodanie nowego lotniska
- **/airport (PUT)** Edycja wybranego lotniska
- **/airport (DELETE)** Usunięcie wybranego lotniska
- **/connection (GET)** Otrzymanie wszystkie połączenia
- **/connection/collect (POST)** Otrzymanie połączenia po wybranym parametrze
- **/connection (POST)** Dodanie połączenia z wybranym lotniskiem
- **/connection/shortest\_path (POST)** Wyszukanie najkrótszej ścieżki pomiędzy lotniskami
- **/connection/fastest\_path (POST)** Wyszukanie ścieżki z najmniejszą liczbą węzłów pomiędzy lotniskami

## 6 Wdrożenie aplikacji

Aplikacje można uruchomić z pomocą kontenera utworzonego za pomocą pliku `Dockerfile`.

Wykonując komendę `"docker-compose up -d"` budujemy uprzednio obraz `react-image` (jeśli taki jeszcze nie istniał), następnie kopiuje odpowiednie foldery z projektu, a następnie instalowane są potrzebne moduły komendą `npm install`. Potem budowana jest aplikacja reactowa komendą `npm run build`, a następnie włączany jest server, który obsługuje komunikacje z bazą danych oraz wyświetla aplikacje posługując się uprzedni zbudowanym klientem.

Aplikacja działa za pomocą kontenera Docker, ponieważ platformy takie jak Heroku czy IBM nie pozwalają już na bezpłatną uruchomienie aplikacji na serwerze.

Repozytorium kodu źródłowego jest dostępne w serwisie github:

<https://github.com/MrLipa/FlightPlanner>

## 7 Ewentualne usprawnienia możliwe do wykonania

Aby usprawnić działanie aplikacji należy dodać o wiele więcej połączeń z dodatkowym parametrem daty lotu, a następnie wyszukiwać optymalnych połączeń ze względu na dystans, liczbę przesiadek oraz godziny odlotu i przylotu.

Do pełni funkcjonalności można też dodać dodatkowy rodzaj węzła, jakim jest samolot, a następnie przypinać/odpinać go od węzłów lotniska, jeśli wykonywany jest przelot z jednego lotniska do drugiego.



## 8 Literatura

- [1] Antoni Dydejczyk, Przetwarzanie danych w chmurach obliczeniowych  
*[https://newton.fis.agh.edu.pl/ antek/index.php?sub=dc\\_wykl](https://newton.fis.agh.edu.pl/antek/index.php?sub=dc_wykl)*
- [2] Neo4j, Shortest path planning  
*<https://neo4j.com/docs/cypher-manual/current/execution-plans/shortestpath-planning/>*