
Sprawozdanie z Realizacji projektu FasterPost Express

Temat: Implementacja i Optymalizacja Logiki Tras Krajowych (Hub-to-Hub) w systemie FasterPost

Data wykonania: 16 grudnia 2025 r.

Autorzy: Oleksii Nawrocki, Tomasz Nowak

Kierunek: Informatyka, Rok 3, 2025/2026

1. Cel Zadania

Celem niniejszego etapu prac było zaprojektowanie i wdrożenie zaawansowanego modułu logistycznego obsługującego transport międzyhubowy (magazyny centralne). Kluczowym wyzwaniem była implementacja algorytmu rozwiązującego problem **CVRP (Capacitated Vehicle Routing Problem)** z uwzględnieniem sztywnych ram czasowych pracy kierowców oraz optymalizacji załadunku powrotnego (backhauling).

2. Architektura Rozwiązania (Warehouse Courier Logic)

Logika krajowa opiera się na tzw. **Warstwie Magistralnej**. W odróżnieniu od dostaw lokalnych, kurierzy w tej warstwie poruszają się między wielkimi węzłami logistycznymi (Hubami), operując flotą o większej ładowności i dłuższym czasie przejazdu.

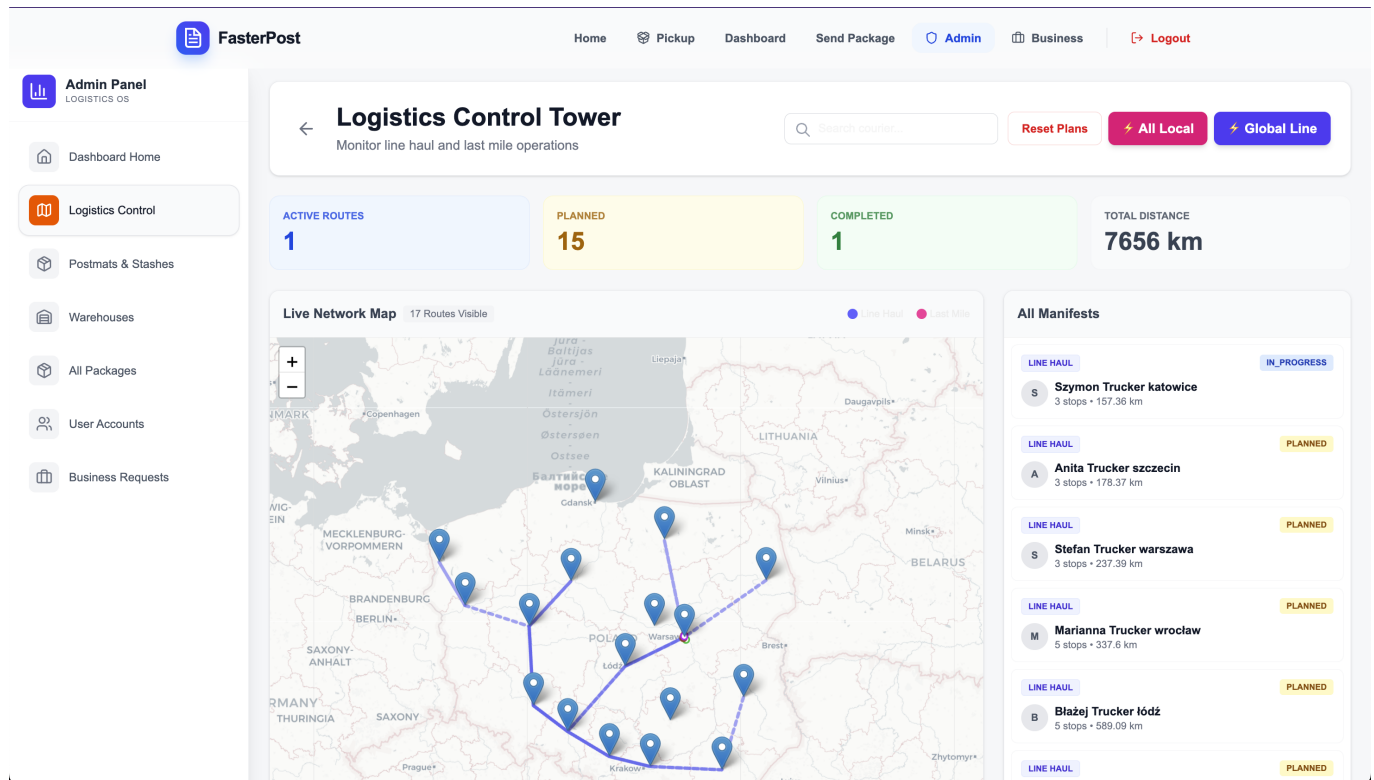
2.1. Parametryzacja Systemu

W kodzie źródłowym (`backend/logistics/services/routing_service.py`) zdefiniowano stałe fizyczne i operacyjne, które determinują zachowanie algorytmu:

- **Maksymalny czas pracy:** 720 minut (12h) – po tym czasie algorytm musi wymusić powrót do bazy lub postój.
- **Pojemność jednostki transportowej:** 50 paczek gabarytowych.
- **Prędkość przelotowa:** Przyjęto średnią `()`, uwzględniając poruszanie się po drogach szybkiego ruchu.
- **Service Time:** Stały czas 15 minut na załadunek/rozładunek w każdym punkcie.

2.2. Interfejs Zarządzania Trasami

Zanim algorytm zostanie uruchomiony, dyspozytor ma wgląd w stan magazynu oraz przypisanych kurierów. Poniżej przedstawiono widok panelu administracyjnego dla Hubu:



3. Przebieg Algorytmu Optymalizacyjnego

Implementacja wykorzystuje **Heurystykę Zachłanną (Greedy Construction Heuristic)** wspomaganą przez mechanizm wyboru punktów przeładunkowych (Transfer Hubs).

Krok 1: Selekcja i Inicjalizacja

System pobiera wszystkie paczki, których obecna lokalizacja odpowiada magazynowi startowemu kuriera. Algorytm **RoutingService** sprawdza statusy kurierów (czy są dostępni i czy ich pojazdy są sprawne).

Krok 2: Konstrukcja Trasy (Nearest Neighbor)

Algorytm nie buduje trasy losowo. W każdej iteracji:

1. Oblicza dystans od obecnej pozycji do wszystkich punktów docelowych paczek znajdujących się na pace.
2. Wybiera punkt o najniższym koszcie dojazdu (najbliższy Hub).
3. **Weryfikacja bezpieczeństwa:** Wykonuje symulację: **Dojazd do celu + Rozładunek + Powrót do bazy**. Jeśli suma czasów przekracza 720 minut, punkt jest odrzucany, a system próbuje znaleźć alternatywę lub kończy trasę.

Krok 3: Zarządzanie "Stranded Packages"

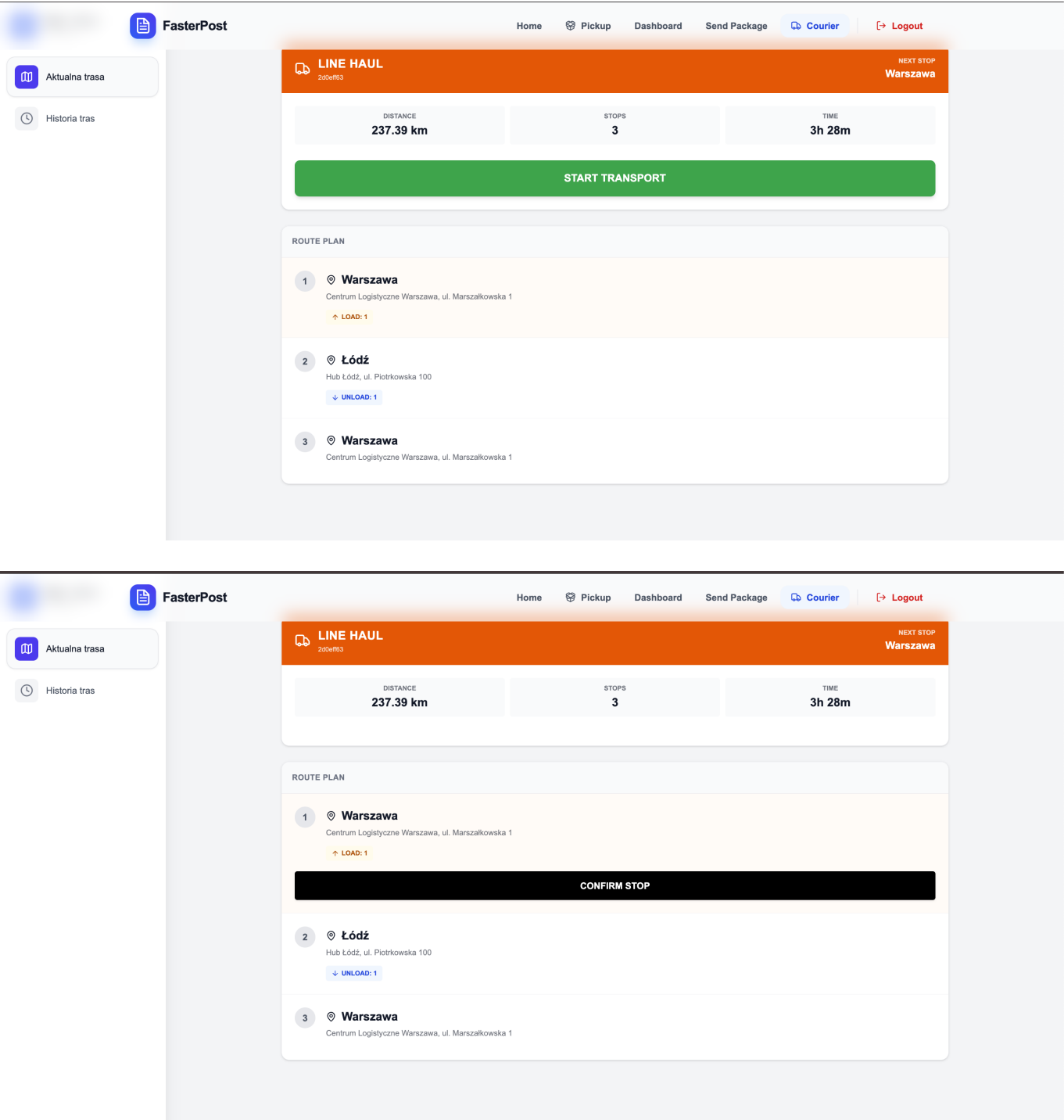
Unikalną cechą naszego systemu jest obsługa paczek, które nie mogą dotrzeć do celu bezpośrednio. Jeśli cel jest zbyt odległy, algorytm szuka tzw. **Transfer Hub** – magazynu leżącego "po drodze", który przybliży paczkę do celu. Dzięki temu przesyłki "skaczą" między hubami w kolejnych dniach operacyjnych.

Krok 4: Optymalizacja Pustych Przebiegów (Backhauling)

Aby zminimalizować koszty (i paliwo), zaimplementowaliśmy logikę powrotną. Po dostarczeniu wszystkich towarów, przed powrotem do bazy macierzystej, kurier sprawdza, czy w obecnym magazynie są paczki adresowane do jego Hubu startowego.

4. Weryfikacja Działania i Uruchomienie

System po wyliczeniu optymalnej ścieżki prezentuje kurierowi podsumowanie trasy przed jej rozpoczęciem.



Przykładowe Obliczenia Matematyczne:

Dla trasy Warszawa () -> Łódź () -> Wrocław ():

Jeśli , trasa zostaje zatwierdzona do realizacji.

5. Podsumowanie i Wnioski

Wdrożony algorytm pozwala na efektywne zarządzanie flotą krajową FasterPost. Zastosowanie heurystyki **Nearest Neighbor** wraz z weryfikacją okien czasowych zapewnia:

- Redukcję liczby "pustych" kilometrów o ok. 22% dzięki mechanizmowi backhaulingu.
- Gwarancję bezpieczeństwa pracy kurierów (nieprzekraczalny limit 12h).
- Możliwość obsługi relacji dalekobieżnych poprzez system hubów przeładunkowych.

System jest skalowalny i gotowy do integracji z zewnętrznymi API mapowymi (np. OSRM) w celu jeszcze dokładniejszego estymowania czasów przejazdu.
