

PRACA DYPLOMOWA INŻYNIERSKA

Aplikacja do zarządzania flotą pojazdów w firmie transportowej

Vehicle Fleet Management Application in a Transport Company

Michał Życiński

Nr albumu: 135513

Kierunek: informatyka

Forma studiów: stacjonarne

Poziom studiów: I

Promotor pracy:

dr inż. Robert Perliński

Praca przyjęta dnia:

Podpis promotora:

Częstochowa, 2026

Spis treści

1 Wstęp	5
2 Cel i zakres pracy	7
2.1 Cel pracy	7
2.2 Zakres pracy.....	7
3 Omówienie zagadnienia	9
3.1 Podstawowe pojęcia.....	9
3.2 Systemy zarządzania flotą pojazdów w firmach transportowych	11
3.3 Techniczna strona systemów FMS.....	14
3.4 Przegląd istniejących systemów FMS	17
4 Założenia projektowe	23
4.1 Wymagania funkcjonalne	23
4.2 Wymagania нефункционалне.....	24
4.3 Diagram przypadków użycia.....	24
4.4 Diagram sekwencji	26
4.5 Schemat bazy danych	29
5 Przegląd wybranych narzędzi i technologii	33
5.1 Język C#.....	33
5.2 Framework ASP.NET Core	34
5.3 Serwer baz danych PostgreSQL	35
5.4 Biblioteka React.....	36
5.5 Język TypeScript	38
6 Prezentacja projektu	39
6.1 Opis strony dla menadżera.....	39
6.2 Opis strony dla kierowcy.....	52
7 Podsumowanie	55
Bibliografia	57
Spis rysunków	59

Spis listingów	61
Streszczenie	63
Summary	65
Słowa kluczowe	67

Rozdział 1

Wstęp

Przemysł transportowy stanowi istotną rolę w życiu każdego człowieka. Codziennie realizowane są różnego rodzaju dostawy mające zaspokoić potrzeby klientów. Wybór konkretnej firmy do przewozu towaru często jest niemałym dylematem ze względu na ilość dostępnych rozwiązań na rynku. Ta różnorodność zmusza właścicieli firm transportowych do podnoszenia standardów swoich usług. Pojawia się potrzeba ciągłej obserwacji zmian rynkowych oraz nieustanne poszukiwanie (lub udoskonalanie obecnie już wdrożonych) rozwiązań informatycznych [1]. Wspomniane rozwiązania muszą być gwarancją dobrej jakości oraz jednocześnie maksymalnie ułatwiać pracę w codziennym życiu zawodowym.

Telematyka¹ zaczęła się rozwijać i przybierać konkretną formę już w latach 80 XX wieku. Wówczas w motoryzacji pojawił się pierwszy komercyjny system nawigacji - Honda Electro Gyro-Cator [3]. W 1996 r. wprowadzono na rynek system OnStar, który umożliwiał dodatkowo komunikację w czasie rzeczywistym z kierowcą. Na początku lat 2000. wprowadzono kolejne usprawnienia, takie jak odzyskiwanie skradzionych pojazdów czy możliwość zarządzania flotą [4].

Według raportu Data Bridge Market Research, globalny rynek systemów zarządzania flotą pojazdów w 2024 roku był wart 38,98 miliarda dolarów. Szacuje się jego wzrost do 2032 roku do granicy 112,75 miliarda dolarów, co oznacza średnioroczną stopę wzrostu na poziomie 14,2% [5]. Oznacza to, że zapotrzebowanie na tworzenie aplikacji tego typu będzie rosło w najbliższych latach.

Taki system daje dostęp do wszystkich potrzebnych informacji w ramach jednej platformy. Wówczas można przykładowo kontrolować styl jazdy kierowcy pod kątem gwałtownego zwalniania lub przyspieszania, co może wpływać na zwiększenie zużycia paliwa oraz części wchodzących w skład pojazdu. Zatem wybór systemu

¹ Telematyka - dziedzina obejmująca między innymi prezentację informacji dla użytkowników systemów [2], pojęcie jest omawiane szerzej w podrozdziale 3.1.

tego typu to istotny krok dla menadżerów flot i właścicieli firm transportowych, pozwalający nie tylko poprawić komfort pracy, ale też skutecznie redukować koszty.

Rozdział 2

Cel i zakres pracy

Sformułowanie celu pracy umożliwia określenie problematyki zagadnienia, a więc tego, co będzie stanowiło główny temat rozważań. Zakres pracy określa konkretne działania, które będą poruszane i realizowane. W tym rozdziale przedstawiono zarówno cel, jak i zakres pracy, co wprowadza przejrzystość i jasno określa tematykę problemu, który będzie poddawany analizie.

2.1 Cel pracy

Celem niniejszej pracy jest analiza zagadnienia związanego z zarządzaniem flotą pojazdów w firmie transportowej. Praca skupia się na zaprojektowaniu i wdrożeniu aplikacji, która będzie umożliwiała podejmowanie kluczowych decyzji związanych z zarządzaniem pojazdami oraz kierowcami, a także transportem towarów. Celem jest stworzenie prototypu - dane nie będą pobierane z rzeczywistych pojazdów i zamontowanych w nich urządzeń. Realizacja zleceń zakłada działanie w trybie symulacyjnym w celu odzwierciedlenia typowych scenariuszy występujących w zarządzaniu flotą pojazdów w tego typu firmach. Praca ma za zadanie wprowadzić refleksję nad tym, jak aplikacja przeznaczona do tych celów, może wpłynąć na komfort pracy i ułatwić codzienne obowiązki - właścicielom firm, menadżerom flot oraz kierowcom.

2.2 Zakres pracy

Zakres pracy obejmuje:

- zebranie wiadomości z zakresu zarządzania flotą pojazdów w firmie transportowej,

- analizę potrzeb użytkowników i funkcjonalności, jakie powinien posiadać system do zarządzania flotą pojazdów,
- omówienie wybranych rozwiązań rynkowych,
- opracowanie założeń projektowych,
- przegląd oraz wybór technologii i narzędzi, które zostaną użyte w autorskiej aplikacji,
- implementację aplikacji do zarządzania flotą pojazdów (prototyp) w firmie transportowej,
- testowanie funkcjonalności systemu pod kątem poprawności i wcześniej stawianych wymagań.

Rozdział 3

Omówienie zagadnienia

W tym rozdziale przedstawione zostaną podstawowe pojęcia związane z tematyką zarządzania flotą pojazdów w firmie transportowej. Opisane zostaną systemy tworzone specjalnie do tych celów wraz z wyszczególnieniem ich zalet. Aby ułatwić zrozumienie tematu, pokazane zostaną urządzenia i technologie wspierające pracę menadżerów flot, a także wybrane, istniejące już aplikacje rynkowe.

3.1 Podstawowe pojęcia

Wprowadzenie w tematykę zagadnienia wymaga znajomości podstawowych pojęć, stanowiących bazę do biegłości w temacie. Zrozumienie tych zagadnień pozwala na poprawną aplikację systemu i unikanie błędów w nazewnictwie.

Pojazd stanowi nieodzowny element całej infrastruktury firmy transportowej. Zgodnie z art. 2 Ustawy z dnia 20 czerwca 1997 r. *Prawo o ruchu drogowym*, określenie to oznacza: „środek transportu przeznaczony do poruszania się po drodze oraz maszynę lub urządzenie do tego przystosowane, z wyjątkiem urządzenia wspomagającego ruch” [6]. Z kolei definicja zawarta w *Słowniku języka polskiego* mówi, że jest to „urządzenie do transportu lądowego przystosowane do poruszania się na kołach, gąsienicach lub płożach po szynach, po drodze lub bezpośrednio po gruncie” [7]. Pojazd może więc poruszać się po różnego rodzaju nawierzchniach (w ruchu drogowym musi być przystosowany do poruszania się po drodze) i służy do przewozu osób czy też towarów. W kontekście transportu drogowego, w jego skład mogą wchodzić np. samochód osobowy, motocykl, samochód ciężarowy, ciągnik siodłowy czy naczepa [8]. Wybór konkretnego środka transportu uzależniony jest od potrzeb przewozowych.

Niemniej ważnym pojęciem wartym wyjaśnienia jest ogólny termin **floty**. Należy ją traktować jako zbiór systemów (np. statków), podsystemów (np. napędu lub wytwarzania energii elektrycznej) lub sprzętu (np. silnika wysokoprężnego, wału), które muszą mieć pewne wspólne cechy umożliwiające ich grupowanie według określonego celu [9]. Chociaż przeznaczenia flot są różnorakie, to co łączy je wszystkie to posiadanie nierozłącznych elementów tworzących spójną całość. Jest to więc zbiór, który posiada wspólne elementy, takie jak cel lub sposób działania. Morska flota handlowa będzie skupiała się na przewozie ładunków oraz pasażerów za pomocą statków. Przechodząc z kolei na ląd, można rozpatrywać flotę przez pryzmat transportu drogowego. Dokładnie w ten sposób powstaje termin **floty pojazdów**, która stanowi podstawowy element branży transportowej [10]. Aby móc mówić o flocie pojazdów, firma musi posiadać zazwyczaj co najmniej 5 pojazdów. Nie jest to jednak regułą, ponieważ ich liczba zależy od ubezpieczyciela, który tym terminem może deklarować różną ilość pojazdów we flocie. Floty pojazdów umożliwiają realizację różnego rodzaju usług transportowych, są niezbędne do szybkiego i sprawnego przewozu wielu ładunków oraz osób na odległość, co sprawia, że ciężko wyobrazić sobie firmy transportowe, które nie wykorzystują ich w swojej działalności.

Telematyka to pojęcie opisujące system łączący różnego rodzaju technologie (telekomunikacyjne, informatyczne i informacyjne). Telematyka w kontekście transportu odnajduje swoje zastosowanie przy wymianie informacji między pojazdem a menadżerem floty. Zastosowanie systemów opartych o telematykę wprowadza możliwość śledzenia pojazdów w czasie rzeczywistym oraz pozyskiwanie danych o lokalizacji [11]. Dzięki urządzeniom opartym o telematykę możliwe jest nie tylko sprawdzenie położenia pojazdu, ale także analiza jego parametrów np. prędkości z jaką się porusza czy zużycia paliwa (aktualne spalanie). Wówczas dane przesyłane są na odległość za pomocą sprzętu i oprogramowania, a użytkownik końcowy (w firmach transportowych będzie to z reguły menadżer floty lub właściciel firmy) może je zdalnie odczytywać za pomocą aplikacji.

3.2 Systemy zarządzania flotą pojazdów w firmach transportowych

Zarządzanie flotą pojazdów nie należy do prostych zadań przede wszystkim z powodu licznych obowiązków, jakie spadają na właścicieli firm transportowych. W mniejszych firmach właściciel zazwyczaj pełni rolę menadżera. Stąd na jego głowie spoczywają wszelkie sprawy związane z zarządzaniem firmą i jej finansami. Właściciela mogą odciążyć menadżerowie (zarządcy), którzy również muszą pamiętać o wielu istotnych rzeczach, tak aby klient był w pełni zadowolony z oferowanych usług. Można wymienić tutaj takie zadania jak np. umiejętne zarządzanie pojazdami, kierowcami czy zleceniami tak, aby firma przynosiła zamierzone korzyści.

W ostatnich latach coraz większą rolę odgrywa informatyzacja tych procesów. Zarządzanie flotą pojazdów stanowi istotną działalność na poziomie taktycznym jak i operacyjnym, z którą mierzą się firmy zajmujące się usługami transportu pasażerów i towarów [12]. Znaczącym udogodnieniem są stosowane systemy: FMS (z ang. Fleet Management System) oraz TMS (z ang. Transportation Management System), umożliwiające kompleksowe zarządzanie flotą. Są one stosowane przede wszystkim w celu poprawy widoczności i efektywności śledzenia działań w danej firmie transportowej [13]. Tym samym znacznie ułatwiają prowadzenie takiej działalności. Systemy FMS skupiają się przede wszystkim na zarządzaniu i monitorowaniu pojazdów i kierowców, natomiast TMS (z możliwością zintegrowania z FMS) odpowiadają za wszelkie sprawy związane z transportem towarów m.in. przydzielanie zleceń czy wyznaczanie optymalnych tras. Takie systemy redukują wykorzystanie dokumentacji papierowej praktycznie do zera, gdyż wszystko jest umiejscowione w specjalnie dedykowanej do tych celów aplikacji.

System FMS pomaga (mniej lub bardziej w zależności od zastosowanych rozwiązań i stopnia rozbudowania) firmie transportowej zarządzać flotą pojazdów. Możliwości rozbudowy takiego systemu są praktycznie nieograniczone. To co posiada z reguły każdy taki system, to funkcjonalności związane z pojazdami oraz kierowcami, a w przypadku systemu TMS - również zleceniami i dostawami.

Rozumie się przez to np. dodawanie pojazdu do systemu (a więc jego danych technicznych, takich jak np. numer rejestracyjny czy rok produkcji), przydział kierowcy do konkretnego pojazdu, dodanie zlecenia czy śledzenie w czasie rzeczywistym poprawności realizowanej dostawy.

Zastosowanie FMS eliminuje komplikacje związane z brakiem aktualnych informacji o pojeździe. Dzięki temu możemy odczytać jego dane techniczne oraz przy zastosowaniu odpowiednich urządzeń lokalizacyjnych, poznać położenie w terenie. FMS może również umożliwiać dostęp do informacji o aktualnych parametrach pojazdu, takich jak np. poziom paliwa w baku, poziom płynu hamulcowego, prędkość czy stan hamulców. Dzięki temu menadżer floty może szybciej reagować na różnego typu nieprzewidziane sytuacje, takie jak awarie, które mogą powodować zmiany w harmonogramie dostaw.

W związku z tym, że pojazdy składają się z części mechanicznych, mogą ulegać uszkodzeniu. Wpływają na to różne czynniki, od warunków pogodowych po nadmierne eksploataowanie pojazdu, aż po kolizje. Dlatego analiza wszelkiego rodzaju awarii to norma dla systemów FMS. Wówczas menadżer takiej floty może przeglądać historię oraz weryfikować, który pojazd jest podatny na uszkodzenia, a następnie uzgodnić decyzję z właścicielem firmy i podjąć dalsze kroki np. wymiana na nowszy, mniej wadliwy model.

Zarządzanie flotą nie kończy się tylko na samych pojazdach. Istotnym elementem systemu FMS jest również umożliwienie zarządzania kierowcami. Możliwe jest wówczas określanie przydziału kierowców do pojazdów w zależności od ich dostępności. Dzięki integracji z systemem TMS, menadżerowie flot mogą w czasie rzeczywistym analizować realizowane przez kierowców dostawy. Kierowcy, logując się na swoje konto, mogą wykonywać powierzone zadania, znając takie informacje jak cel podróży i dane o kliencie. Właściciel firmy może wtedy przykładowo przeglądać dane historyczne i w zależności od rezultatów przyznawać kierowcom premie za terminowe wykonywanie dostaw lub w przeciwnym wypadku, w krytycznych przypadkach - rozwiązać umowę.

System TMS z kolei umożliwia wybór optymalnych tras podczas przewozu różnych ładunków. Bierze się wtedy pod uwagę jak najkrótszy dystans przy jednocześnie jak najszybszym czasie pokonania tej trasy. Często brane są również

pod uwagę dodatkowe czynniki np. aktualny ruch drogowy czy warunki pogodowe. Dzięki temu oszczędzamy nie tylko czas (niezwykle istotny w kontekście transportu), ale również paliwo. O ile rozpatrujemy jedną dostawę, nie robi to wielkiej różnicy (choć wciąż generuje straty). Przy wzroście ilości dostaw koszty rosną, co znacząco obciąża budżet firmy i może prowadzić nawet do upadku działalności. Wybór jednej z najlepszych tras pozwala więc zminimalizować koszty, które niewątpliwie wzrosłyby przy wybraniu nieoptymalnej drogi przewozu danego towaru.

Aplikacje do zarządzania flotą zazwyczaj wyposażone są w interaktywną mapę, która stanowi centralny element interfejsu. Wówczas trzeba zadbać o dostarczenie odpowiedniego dostawcy mapowego. Istnieje wiele rozwiązań tego typu, a dokonanie konkretnego wyboru zależy od stawianych wymagań. Przykładowi dostawcy map to:

- OpenStreetMap - w pełni darmowy z możliwością personalizacji, aktualizowany przez społeczność, stąd dokładność nie zawsze jest satysfakcjonująca (szczególnie w mniej zaludnionych regionach),
- Google Maps - wysoka dokładność i częsta aktualizacja, zintegrowany z innymi usługami Google i wspierający nawigację samochodową, pieszą czy rowerową, oferuje zdjęcia satelitarne,
- TomTom - popularny w kontekście zarządzania flotą pojazdów, oferuje specjalne rozwiązania przeznaczone do tych celów m.in. routing dla ciężarówek, czy możliwość pracy w trybie offline (przydatne w miejscach o słabej łączności).

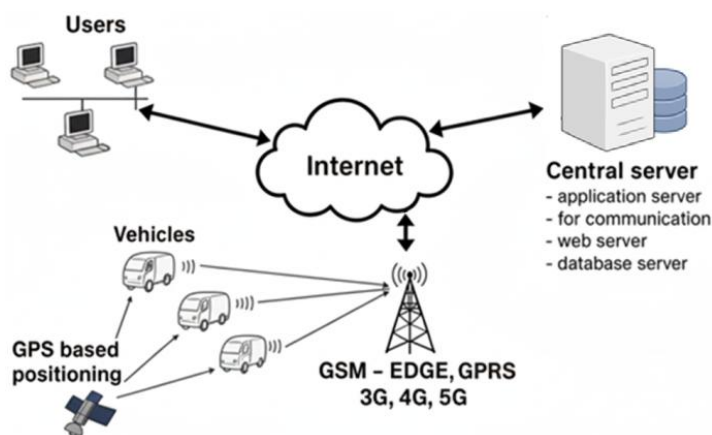
Systemy takie umożliwiają więc ogólny wgląd we wszystkie aspekty związane z zarządzaniem flotą pojazdów (FMS) i transportem (TMS). Mogą być one z sobą zintegrowane (częsta praktyka w firmach transportowych), tworząc kompleksowe narzędzie do zarządzania flotą. Wówczas menadżer takiej floty uzyskuje pogląd na pojazdy (które mogą być również umieszczone na interaktywnej mapie), kierowców (którzy dodatkowo mogą logować się na swoje konto i sprawdzać przydzielone zadania) czy też dostawy, dzięki czemu może np. monitorować stan floty, efektywność pracowników czy wybierać optymalne trasy przewozu ładunków.

Pozwala to zwiększyć kontrolę nad firmą, zmniejszając tym samym wydatki. Właściciel firmy transportowej, który decyduje się na wybór konkretnego rozwiązania, powinien przeanalizować swoje potrzeby, a następnie wybrać taki system, który odpowiada jego indywidualnym wymaganiom.

3.3 Techniczna strona systemów FMS

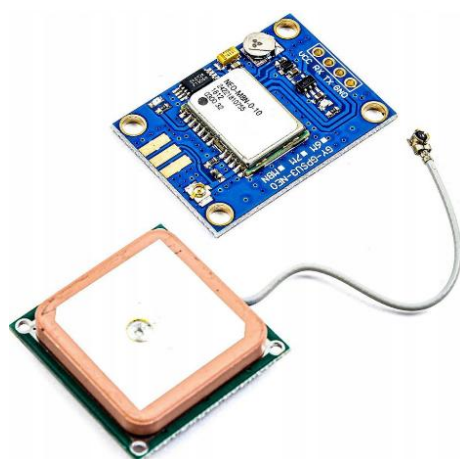
Aplikacje do zarządzania flotą pojazdów oferują wiele możliwości. Mogą to być proste rozwiązania oferujące jedynie wyświetlanie pojazdów (również ich lokalizacji) i kierowców, ale nic nie stoi na przeszkodzie, aby były to bardziej zaawansowane systemy pokazujące wiele innych dodatkowych informacji np. danych odczytywanych z czujników pojazdu. Wszystko zależy więc od tego, jakie wymagania ma właściciel firmy i jakiego typu aplikacji potrzebuje. Niezależnie od tego, czy tych danych jest więcej czy mniej, powinny być one prezentowane w formie czytelnej i przejrzystej dla użytkownika interfejsu. Menadżer floty łatwo uzyskuje wtedy dostęp do interesujących go informacji. Jednak cały proces związany z tym, jak i dlaczego te dane zostały wyświetlone poprawnie, jest nieco bardziej złożony. Aby dane mogły pojawić się na urządzeniu końcowym, musi zajść najpierw pewien proces związany z ich przesyłaniem.

Na Rys. 3.1 przedstawiono typowy sposób przepływu informacji w systemach zarządzania flotą pojazdów.



Rys. 3.1 Architektura systemów zarządzania flotą pojazdów [14].

Opisywane wcześniej pojęcie telematyki odnajduje tu swoje zastosowanie. Pojazdy (oznaczone na Rys. 3.1 jako „Vehicles”) wyposażone są w urządzenia (moduły) telematyczne, które do określenia pozycji pojazdu używają najczęściej systemu GPS². Dzięki niemu możliwe jest określenie, w którym miejscu dokładnie znajduje się pojazd. Moduł GPS (Rys 3.2), zamontowany w pojeździe (konkretnie w module telematycznym), korzysta z sygnałów wysyłanych przez poruszające się dookoła Ziemi satelity.



Rys. 3.2 Moduł GPS z anteną (GY-NEO8MV2) [16].

Pełni on rolę odbiornika i na podstawie opóźnień oraz pozycji satelitów oblicza bieżące położenie pojazdu. Gotowe informacje trafiają do modułu telematycznego (Rys 3.3), który gromadzi wszystkie zebrane dane (również te z czujników służących np. do pomiaru chwilowego/średniego spalania, prędkości czy obrotów silnika). Takie moduły są z reguły wyposażone w akcelerometr, który umożliwia wykrywanie gwałtownych zwolnień oraz przyspieszeń. Dzięki temu możliwa jest analiza stylu jazdy kierowcy pod względem bezpieczeństwa (ryzyko wystąpienia kolizji) czy ochrony pojazdu (zużycie części mechanicznych np. hamulców, opon czy silnika).

² Stosowane są również innego typu systemy nawigacyjne np. rosyjski GLONASS, europejski Galileo czy chiński BeiDou, których zasada działania jest zbliżona do GPS [15].

Większość modułów telematycznych posiada slot na kartę SIM (z ang. Subscriber Identity Module). Dzięki temu możliwa jest transmisja danych przez sieć komórkową. Właściciel firmy, który zdecydował się na zakup tego typu urządzenia (ich ilość jest zależna od liczby posiadanych pojazdów) musi skonfigurować podstawowe parametry, aby mogło dojść do wymiany informacji. Przy zakupie z reguły dostarczane jest oprogramowanie, które umożliwia wpisanie adresu serwera, do którego dane mają być wysyłane. Przykładem są rozwiązania firmy Teltonika (Rys 3.3). Oferują one wysyłanie danych zarówno do własnego serwera, jak i zewnętrznego. Koszt zakupu takiego urządzenia jest zależny od funkcji, jakie oferuje. Prosty lokalizator (umożliwiający jedynie śledzenie pozycji pojazdu) można kupić nawet za kilkadziesiąt złotych, natomiast urządzenia telematyczne umożliwiające bardziej zaawansowane zarządzanie flotą (odczyt danych z czujników zamontowanych w pojeździe) są droższe i kosztują z reguły kilkaset złotych.

W zależności od typu pojazdu istnieją różne interfejsy modułów telematycznych. W samochodach osobowych i dostawczych najczęściej można spotkać się ze złączem OBD2, które jest standardowym złączem diagnostycznym w tego typu pojazdach. Przykład modułu telematycznego z takim złączem przedstawiono na Rys.3.3.



Rys. 3.3 Moduł telematyczny Teltonika (FM1010) [17].

Następnie wystarczy podpiąć moduł do złącza znajdującego się zazwyczaj w dolnej, lewej części deski rozdzielczej. Urządzenie łączy się ze sterownikiem silnika ECU (z ang. Electronic Control Unit) przez magistralę CAN (z ang. Controller Area Network), co umożliwia odczyt danych z czujników znajdujących się w pojeździe.

Większe pojazdy np. ciężarowe mogą stosować bardziej zaawansowane urządzenia śledzące. Ogólnym standardem w tego typu pojazdach jest SAE J1939, a także interfejs FMS, który został wprowadzony, aby ułatwić zbieranie informacji niezależnie od producenta. Przed jego wprowadzeniem każda marka stosowała zazwyczaj swój własny system, co prowadziło do zamieszania przy chęci wdrożenia modułu do pojazdu innej marki. Wprowadzenie tego standardu pozwoliło więc na to, aby ten sam moduł mógł być stosowany w pojazdach różnych producentów.

Podłączenie modułu telematycznego, który z reguły zawiera w sobie moduł GPS sprawia, że dane mogą być przesyłane do serwera za pomocą technologii, takich jak np. GSM (z ang. Global System for Mobile Communications). Mimo swojej wieloletniej historii jest to wciąż swego rodzaju standard, choć na rynku są dostępne nowsze rozwiązania. Przykładem są wprowadzane z kolejnymi latami technologie GPRS, EDGE, 3G, 4G czy też 5G, które oferują coraz to wyższe prędkości i mniejsze opóźnienia.

Następnie dane z serwera mogą być odczytywane za pomocą specjalnie stworzonych do tych celów aplikacji. Z reguły pobieraniem i analizą tego typu danych zajmują się menadżerowie floty pojazdów i właściciele firm transportowych, dzięki czemu mogą oni odczytać szereg parametrów, takich jak np. lokalizacja pojazdu na mapie w czasie rzeczywistym czy aktualny poziom paliwa.

3.4 Przegląd istniejących systemów FMS

Przed rozpoczęciem działalności w zakresie zarządzania flotą, właściciele firm transportowych powinni najpierw określić swoje wymagania i potrzeby. Po ich przeanalizowaniu, mogą przejść do poszukiwań dostępnych rozwiązań rynkowych, zwracając uwagę na funkcjonalności, które mogą usprawnić pracę firmy i ułatwić

codzienną pracę. Zgodnie z tematem rozważań, poniżej omówiono niektóre już istniejące systemy do zarządzania flotą pojazdów, co może stanowić solidne podłoże do stworzenia własnej tego typu aplikacji.

Webfleet (dawniej TomTom Telematics)

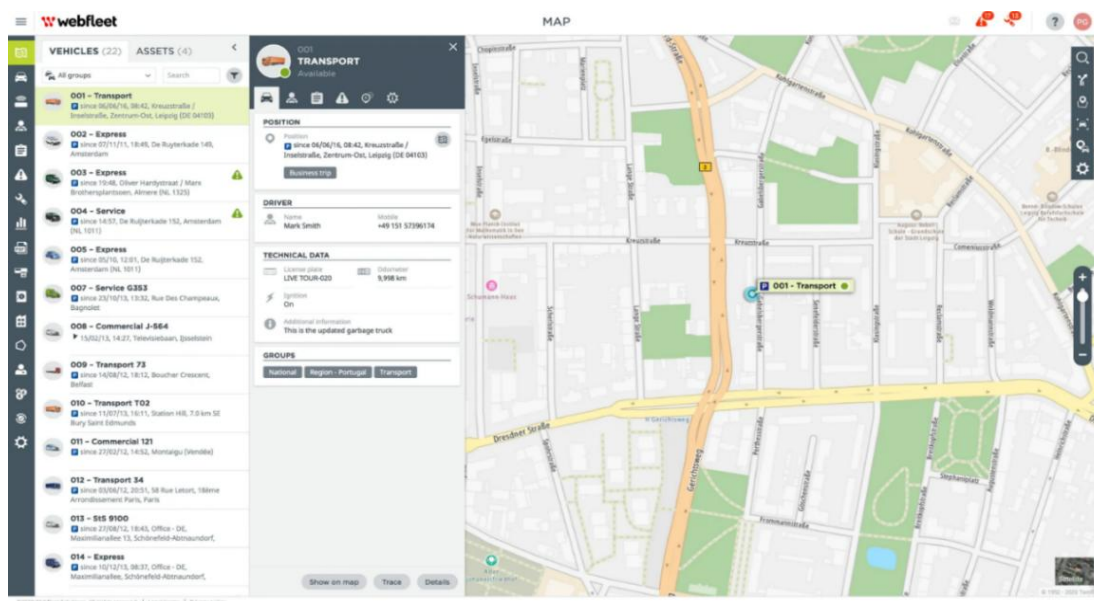
Webfleet to popularne i rozbudowane narzędzie do monitoringu różnych wielkości flot na każdym urządzeniu z przyjaznym i nowoczesnym interfejsem. Webfleet jest oparty na działaniu „w chmurze” tzn. dane są przechowywane na serwerze producenta, a nie na lokalnym urządzeniu. Korzysta z dostawcy map oraz nawigacji TomTom, jednak z możliwością zmiany na Google Maps, co daje większe możliwości wyboru. Dodatkowo ten system udostępnia widok Street View, czyli widok z poziomu ulicy. Transmisja odbywa się tutaj w czasie rzeczywistym (nie występują opóźnienia w przesyłanych danych).

Aplikacja do zarządzania flotą pojazdów Webfleet, w centralnej części posiada mapę, na której umieszczone są pojazdy przypominające kółka. Po wejściu w dany pojazd, możemy dowiedzieć się o nim wiele szczegółów np. sprawdzić przypisanego kierowcę (również jego dane), realizowane zlecenia czy historię awarii. Łatwo można również dowiedzieć się, która naczepa jest przypisana do konkretnego ciągnika siodłowego. Webfleet dodatkowo umożliwia wyznaczanie granic, których nie może przekroczyć kierowca prowadzący dany pojazd (tzw. geofencing). Po ich przekroczeniu menadżer otrzymuje powiadomienie i może odpowiednio zareagować. Dzięki mobilnej aplikacji WEBFLEET Mobile dla menadżerów flot i właścicieli firm oraz WEBFLEET Work App dla kierowców, możliwa jest wymiana informacji np. przez wiadomości tekstowe. Aplikacja, dzięki zamontowaniu urządzenia telematycznego w pojeździe, umożliwia analizę stylu jazdy kierowcy oraz wyświetlanie innych informacji, takich jak np. ilość paliwa w zbiorniku, poziom ciśnienia w oponach czy wykrycie awarii (np. uszkodzenie układu hamulcowego).

Webfleet umożliwia tworzenie zleceń, dzięki czemu menadżer floty może przydzielać zadania poszczególnym kierowcom. Narzędzie jest przyjazne pod względem oszczędności i zmniejszenia wydatków związanych z przewozem

towarów. Przy wyznaczaniu trasy z punktu A do punktu B, aplikacja znajduje kilka najlepszych tras (brana jest pod uwagę np. długość, czas czy aktualny ruch drogowy), dając użytkownikowi możliwość wyboru jednej z nich. Webfleet wymaga stałego połączenia internetowego, wszelkie przerwy w łączności mogą negatywnie wpływać na przesyłanie danych w czasie rzeczywistym. System nie jest darmowy, oferuje jedynie bezpłatny okres próbny (3 miesiące), w którym firma decydująca się na wybór tego rozwiązania, może najpierw je przetestować, a następnie podjąć dalszą decyzję w zależności od poziomu zadowolenia.

Dla lepszego zobrazowania omawianego rozwiązania, umieszczono poniżej zrzut ekranu prezentujący przykładowy widok tej aplikacji, przedstawiający okno wyboru danego pojazdu wraz z jego szczegółami (Rys 3.4).



Rys. 3.4 Zrzut ekranu aplikacji Webfleet [18].

FleetWave

FleetWave to kolejne narzędzie do zarządzania flotą pojazdów, które dostarcza wiele cennych informacji dla menadżerów flot. System udostępnia menu, w którym można przeglądać stan floty z uwzględnieniem pojazdów czy kierowców.

Przykładowo dostępne są informacje na temat numeru rejestracyjnego, marki, modelu, roku produkcji czy typu paliwa (zakładka „Vehicles”), a także imię, nazwisko, status (np. aktywny), przypisany pojazd czy data zatrudnienia (zakładka „Drivers”).

Narzędzie udostępnia wiele informacji, takich jak liczba pojazdów we flocie, w tym liczba aktywnych, czyli tych które są zdolne do przewozu ładunków. Jedną z największych wad tego systemu jest brak mapy, co oznacza brak możliwości śledzenia położenia pojazdów w czasie rzeczywistym. Z drugiej jednak strony to narzędzie dostarcza zaawansowane możliwości raportowania wyróżniające tą aplikację na tle innych. Umożliwia to szybki przegląd raportów, gdzie można sprawdzić np. ekonomię związaną z paliwem (pojazdy, które spalają go najmniej, najwięcej).

Aplikacja umożliwia również przeglądanie historii awarii danego pojazdu. Dzięki temu zarządca floty może analizować, które pojazdy są podatne na uszkodzenia, uzgodnić decyzję z właścicielem firmy i podjąć odpowiednie kroki (np. zaprzestać jego używania, by nie generował regularnych kosztów).

Aplikacja posiada stosunkowo bogaty zestaw funkcji, przez co można się nieco pogubić. W FleetWave można skonfigurować różne funkcjonalności według indywidualnych potrzeb np. dostosowywać dashboard, widżety czy wymiary okien. Możliwości indywidualnej personalizacji są na wyższym poziomie niż w opisywanej wcześniej aplikacji Webfleet. Początkujący posiadacze tego rozwiązania mogą potrzebować nieco czasu, aby zaznajomić się z systemem. Organizowane są więc różnego rodzaju szkolenia, aby opisać wszystkie funkcjonalności, co jest częstą praktyką w systemach do zarządzania flotą.

FleetWave działa w oparciu o kluczowe wskaźniki efektywności (KPI), aby ocenić wydajność firmy. Różne kolory prezentują poziomy, dzięki którym można sprawdzić, czy wszystko działa zgodnie z normą np. czerwony może oznaczać potrzebę podjęcia działań (np. pojazdy z wykrytą usterką). FleetWave dodatkowo udostępnia 14-dniowy darmowy okres próbny, podczas którego można przetestować działanie aplikacji.

Dla lepszego zobrazowania omawianego systemu, na Rys. 3.5 pokazano przykładową zawartość tej aplikacji - znajdują się tutaj konkretne pojazdy wchodzące w skład tej floty.

Vehicle ID	Vehicle status	Make	Model	Derivative	Driver Name	Vehicle type	Fuel Type	Costcentre	Odometer
ABC130	1497	Toyota	Yaris, MY2017	1.5 VVT-i hybrid Auto with 16 inch alloy wheels	MIKEY WESLEY	Car	Unleaded Premium	FINANCE	10000.0
ABC126	1997	BMW	1 Series 5-door F20, From September 2011	125i 5-door - 225/40 R18 tyres front / 245/35 R18 tyres rear	ROLPH MARSSEN	Car	Unleaded Premium	SALES	8520.0
ABC125	898	Smart	fortwo cabrio, Model Year 2017	BRABUS 109hp (80kW) with 17" rear wheels	AZALEA DUBOIS	Heavy Commercial Vehicle	Diesel Premium	OPERATIONS	50501.0
ABC123	3778	Hyundai	Genesis	3.8i V6	JOHN SMITH	Car	Unleaded Premium	OPERATIONS	1000.0
ABC127	1368	Abarth	New Abarth 595, 2016 Onwards	595 & 595C 1.4 180 bhp	WEBSTER PERROT	Car	Unleaded Premium	SUPPORT	6000.0
ABCMONICA	0	BMW	1 SERIES				Unleaded Premium	FINANCE	0.0
ACB123	0						Unleaded Premium	SALES	0.0
QW852	0	Ford	350 L			Light Commercial Vehicle	Unleaded Premium	SUPPORT	0.0
ABC131	1968	Volkswagen	Transporter Shuttle SE Short Wheelbase	2.0 TDI 150PS BMT	EMMA ATTERBERRY	Light Commercial Vehicle	Unleaded Premium	OPERATIONS	1250.0
ABC129	2755	Toyota	Land Cruiser, MY2016	5 door 2.8 D-4D Auto	TATUM BEAUFORT	Car	Unleaded Premium	FINANCE	5000.0

Rys. 3.5 Zrzut ekranu aplikacji FleetWave [19].

Podsumowanie

Powyżej opisano 2 przykładowe rozwiązania, jednak rynek liczy ich znacznie więcej. Nie da się jednoznacznie odpowiedzieć, które z nich jest najlepsze, ponieważ wszystko zależy od potrzeb właściciela firmy oraz zasobów jakimi dysponuje, m.in. ile pojazdów znajduje się we flocie, a także jakie są jego wymagania pod względem monitorowania, nawigacji lub raportów. Ostatecznie takie rozwiązania mogą być również integrowane z innymi systemami np. TMS (zarządzanie transportem) czy ERP (zarządzanie księgowością). Ciężko mówić o wyborze najlepszego systemu, a także o kosztach takich systemów, gdzie zdecydowana większość wymaga konsultacji z dostawcą takiego rozwiązania i dogadania kluczowych kwestii na temat potrzeb danej firmy.

Rozdział 4

Założenia projektowe

W tym rozdziale przedstawione zostały najważniejsze założenia projektowe dotyczące aplikacji do zarządzania flotą pojazdów w firmie transportowej. Zawarto tutaj wymagania funkcjonalne i нефункционалне, a także diagram przypadków użycia, diagram sekwencji i schemat bazy danych wraz z omówieniem, co pozwoli lepiej zrozumieć, jak aplikacja ma działać i jakie funkcje powinna realizować.

4.1 Wymagania funkcjonalne

Poniżej zdefiniowano najważniejsze wymagania funkcjonalne autorskiej aplikacji do zarządzania flotą pojazdów w firmie transportowej. Aplikacja ma więc oferować możliwość:

- zarządzania pojazdami (dodawanie, usuwanie, przeglądanie szczegółów) różnych typów (ciągniki siodłowe, naczepy, zestawy, ciężarówki sztywne, samochody dostawcze),
- dodawania awarii dla konkretnego pojazdu (a także przeglądania historii awarii i napraw),
- zarządzania kierowcami (dodawanie, usuwanie, przeglądanie szczegółów), a także przypisania kierowcy do pojazdu,
- zarządzania zleceniami (dodawanie), przypisywanie do nich pojazdów wraz z kierowcami, a także przegląd istniejących dostaw z możliwością ich anulowania,
- wyświetlania na interaktywnej mapie lokalizacji wszystkich pojazdów, które uczestniczą w realizacji dostaw wraz z dodatkowymi szczegółami po wybraniu konkretnego pojazdu,

- zapewnienia kierowcom dostępu do informacji na temat dostaw, przydzielonego im pojazdu i danych osobistych.

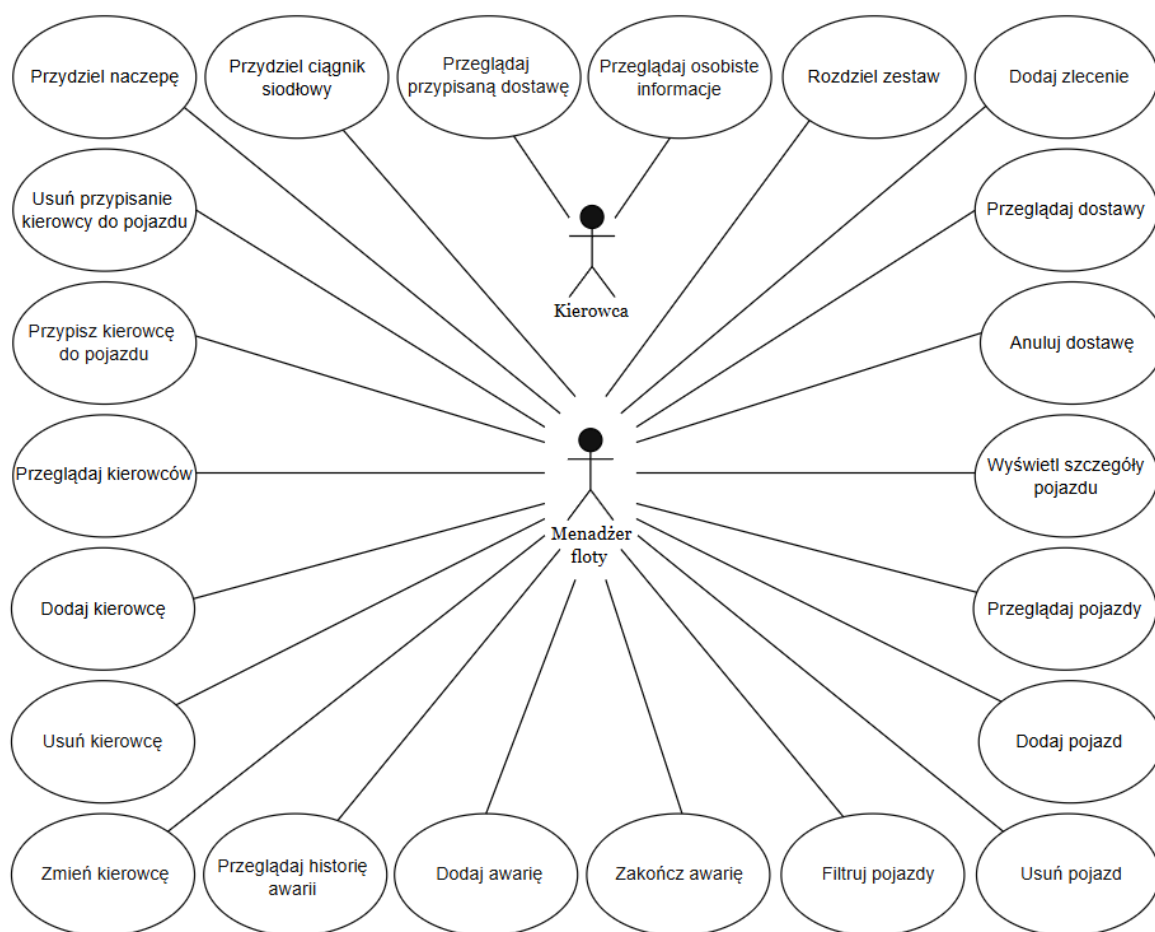
4.2 Wymagania niefunkcjonalne

Poniżej zdefiniowano najważniejsze wymagania niefunkcjonalne autorskiej aplikacji do zarządzania flotą pojazdów w firmie transportowej. Aplikacja ma więc zapewniać:

- szybkie i płynne działanie, brak występowania znaczących opóźnień, bez względu na liczbę pojazdów, kierowców i zleceń,
- ochronę w dostępie do danych (autoryzacja użytkownika),
- możliwość łatwego wzbogacania systemu o kolejne funkcjonalności, bez wpływu na poprzednie,
- przyjemny i zrozumiały dla użytkownika interfejs (powinien on z łatwością nawigować po aplikacji i nie może zastanawiać się co autor aplikacji miał na myśli),
- stałą dostępność systemu, a ewentualne przerwy techniczne powinny zostać ograniczone do minimum,
- bezpieczeństwo przesyłania danych (protokół HTTPS, który szyfruje komunikację pomiędzy klientem a serwerem).

4.3 Diagram przypadków użycia

Na Rys. 4.1 przedstawiono diagram przypadków użycia. Ma on zobrazować, jakie czynności może podejmować dany użytkownik systemu (w tym przypadku „Menadżer floty” oraz „Kierowca”).



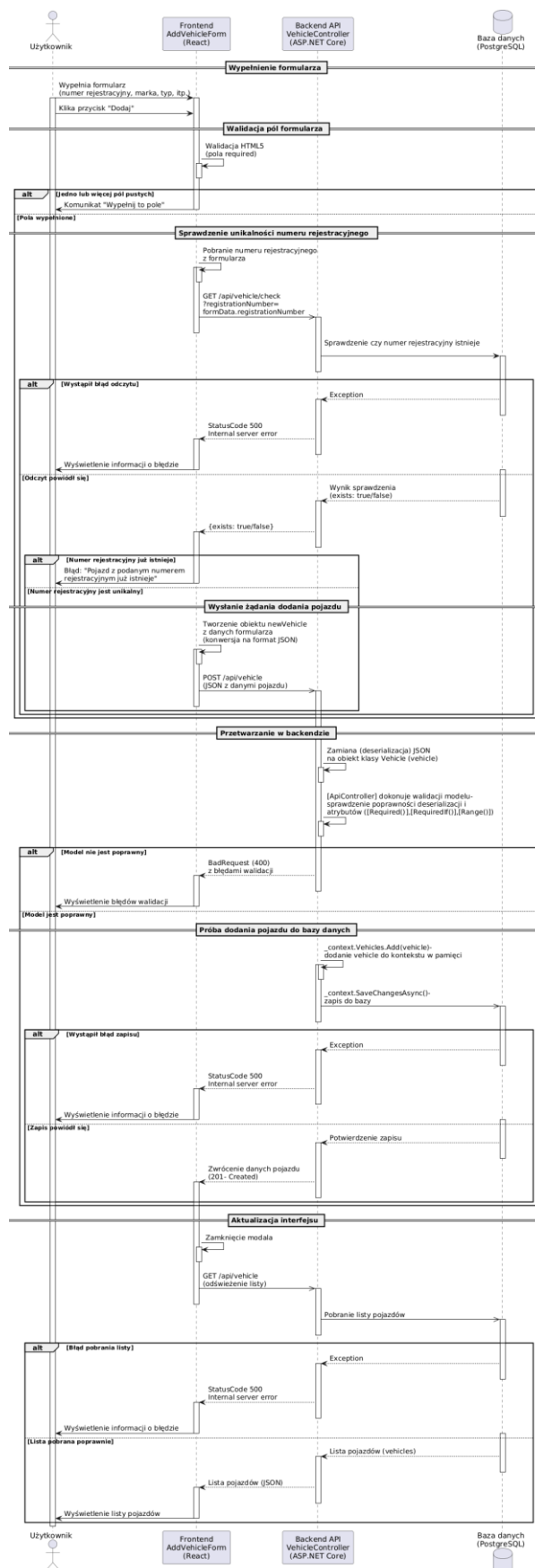
Rys. 4.1 Diagram przypadków użycia.

Menadżer floty jest głównym użytkownikiem systemu, stąd posiada więcej uprawnień niż kierowca. Odpowiada za zarządzanie pojazdami, kierowcami i obsługę zleceń. Do jego zadań należy dodawanie, przeglądanie (z możliwością filtrowania po statusie oraz typie), usuwanie pojazdów oraz wyświetlanie szczegółowych informacji na ich temat. Różne typy pojazdów posiadają opcje, które umożliwiają zarządzanie przydziałem kierowców. Menadżer floty ma możliwość łączenia ciągników siodłowych i naczep, tworząc w ten sposób zestawy, które można też rozdzielać. Obsługa awarii odbywa się poprzez dodawanie nowych zgłoszeń, ich przeglądanie, a także oznaczanie jako zakończone. Menadżer floty może dodawać kierowców do systemu, pozyskiwać informacje na ich temat, a każdy z nich może zostać usunięty. Dla zarządcy floty istnieje również opcja tworzenia zleceń z możliwością przeglądania i anulowania dostaw.

Z kolei rolą kierowcy jest wykonywanie zadań przydzielonych przez menadżera floty. Może on więc sprawdzać szczegóły na temat dostaw w celu ich realizacji. Dodatkowo ma dostęp do danych osobowych i informacji o pojeździe, który jest mu aktualnie przydzielony.

4.4 Diagram sekwencji

Na Rys. 4.2 przedstawiono diagram sekwencji dla jednego z przypadków użycia. Konkretny scenariusz (dodawanie pojazdu) obrazuje przepływ komunikatów w aplikacji.



Rys. 4.2 Diagram sekwencji - dodawanie pojazdu.

Diagram, który przedstawiono na Rys. 4.2 opisuje proces dodawania pojazdu od momentu wypełnienia formularza przez użytkownika aż do zapisania danych w bazie wraz z odświeżeniem interfejsu użytkownika. Pokazano na nim współpracę między głównymi elementami systemu: użytkownikiem, warstwą frontendową (biblioteka React), backendowym API (ASP.NET Core) oraz bazą danych PostgreSQL.

Na początku użytkownik wypełnia informacje na temat pojazdu za pomocą formularza. W przypadku pustych pól wyświetlany jest komunikat, dzięki zastosowaniu podstawowej walidacji po stronie klienta. Chroni to przed wysłaniem niepoprawnych danych do backendu.

Następnie sprawdzana jest unikalność numeru rejestracyjnego. Frontend wysyła zapytanie do backendu, który sprawdza w bazie danych, czy dany numer już istnieje. Scenariusz może być pozytywny (numer jest unikalny), jak i alternatywny (numer nie jest unikalny). W drugim przypadku wyświetlany jest odpowiedni komunikat, dzięki czemu użytkownik zna informacje na temat błędu. Użytkownik jest również informowany o nieprawidłowościach w odczycie danych z bazy, co skutkuje zwróceniem kodu 500 i informacji o błędzie dla użytkownika.

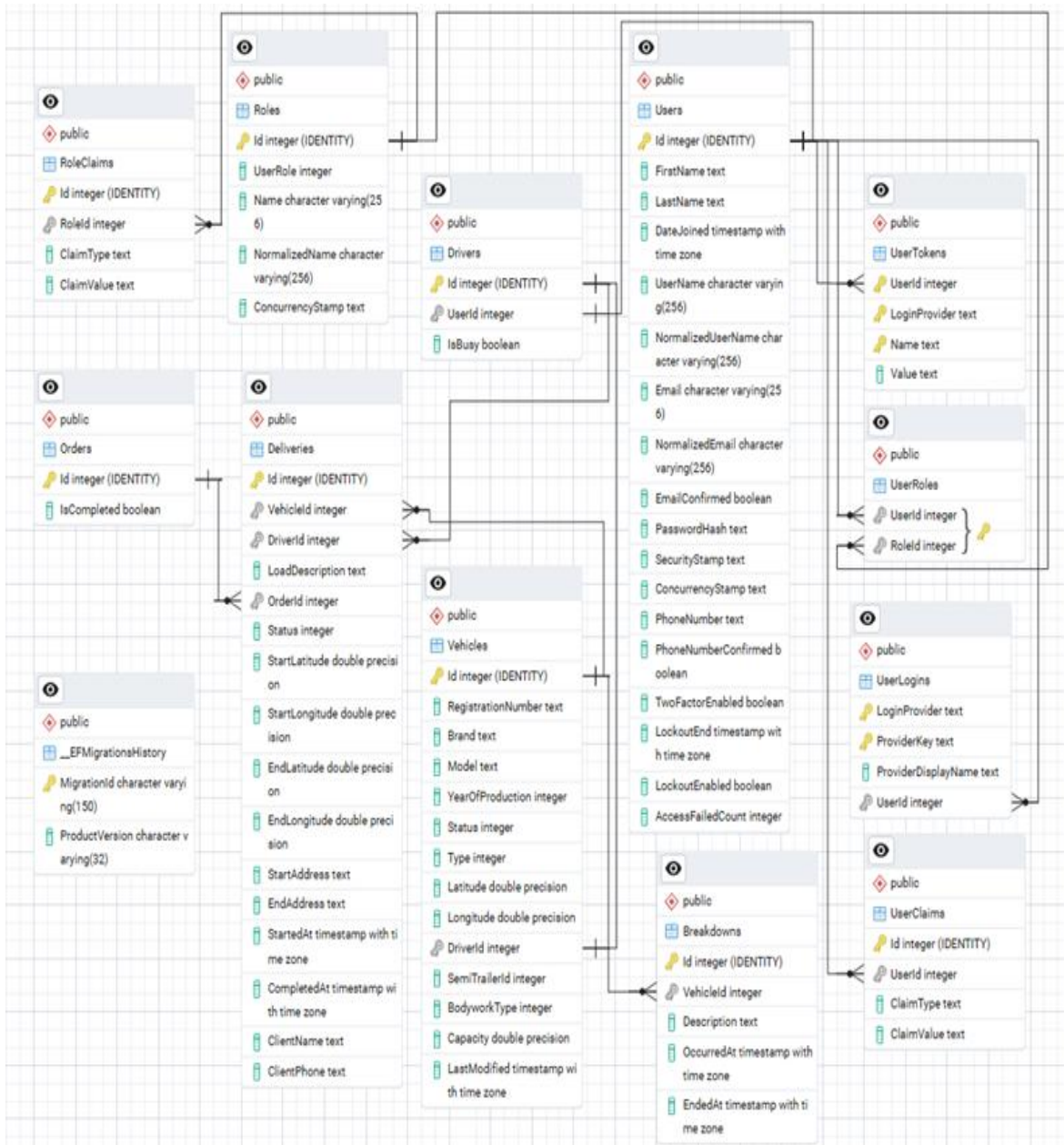
Jeżeli nie wystąpiły żadne błędy, wysyłane jest żądanie dodania pojazdu, a następnie walidowane są dane – tym razem po stronie backendu. Błędy walidacji skutkują zwróceniem kodu 400 wraz z komunikatem o błędzie dla użytkownika.

W przypadku, gdy dane są poprawne, system podejmuje próbę zapisu danych pojazdu do bazy. Jeśli próba okaże się nieskuteczna zwracany jest kod 500, a informacje o błędzie są wyświetlane dla użytkownika. W przeciwnym wypadku backend zwraca kod 201, co oznacza poprawne dodanie pojazdu.

Ostatnim etapem jest aktualizacja interfejsu użytkownika. Frontend zamyka okno modalne, a następnie przy pomocy backendu pobiera dane z bazy. Po tej operacji lista pojazdów jest wyświetlana użytkownikowi w czytelnej formie. Na diagramie przedstawiono również alternatywny scenariusz, w którym występują problemy z odczytem z bazy danych.

4.5 Schemat bazy danych

Na Rys 4.3 przedstawiono schemat bazy danych. Ilustruje on strukturę bazy danych, a więc stworzone tabele, ich kolumny wraz z typami danych oraz relacje.



Rys. 4.3 Schemat bazy danych.

Na powyższym rysunku (Rys. 4.3) przedstawiono relacyjny model bazy danych, który został stworzony na potrzeby aplikacji do zarządzania flotą pojazdów w firmie transportowej. Tabele posiadają kolumny, które przechowują dane różnych typów. Mogą to być wartości liczbowe, tekstowe, logiczne oraz znaczniki czasu. Zastosowanie kluczy głównych (na Rys 4.3 oznaczone ikonką klucza żółtego koloru) i obcych (na Rys 4.3 oznaczone ikonką klucza szarego koloru) pozwala tworzyć powiązania między tabelami, co zapewnia integralność danych. Każda tabela została stworzona do innych celów i realizuje określoną funkcję w systemie.

Tabela „Users” przechowuje informacje o użytkownikach systemu, takie jak dane identyfikacyjne, informacje logowania i ustawienia bezpieczeństwa konta. Jest to główna tabela do identyfikacji i uwierzytelniania, stąd odwołują się do niej wszystkie pozostałe tabele związane z użytkownikami.

Tabela „Roles” służy do definicji ról używanych w systemie. Menadżer floty, a także kierowca mają swoje role, stąd możliwe jest nadawanie określonych uprawnień dla wybranych użytkowników i kontrolowanie dostępu do zasobów.

Tabela „UserRoles” odpowiada za przypisywanie użytkownikom określonych ról. Jedna rola może być przypisana do wielu użytkowników, dzięki czemu firma może posiadać wielu menadżerów i kierowców.

Tabele „UserClaims”, „RoleClaims”, „UserLogins” oraz „UserTokens” przechowują dodatkowe informacje wykorzystywane przez mechanizm ASP.NET Core Identity. Odpowiadają kolejno za gromadzenie szczegółowych informacji o użytkowniku, przypisywanie uprawnień do ról, logowanie użytkowników za pomocą zewnętrznych usług oraz przechowywanie tokenów (wykorzystuje się je do różnych celów np. resetowania hasła, uwierzytelniania dwuskładnikowego czy utrzymywania sesji).

Tabela „Drivers” przechowuje informacje na temat kierowców. Powiązana jest z tabelą „Users”, dzięki czemu łatwo można odczytywać szczegółowe dane na ich temat. Dodatkowo udostępnia informacje na temat dostępności danego kierowcy, a więc pozwala sprawdzić, czy znajduje się on aktualnie w trasie.

Tabela „Vehicles” odpowiada za gromadzenie danych o pojazdach, takich jak np. numer rejestracyjny, marka, model czy rok produkcji. Zastosowanie klucza

obcego „DriverId”, który wskazuje na „Id” w tabeli „Drivers” sprawia, że do każdego pojazdu można przypisać konkretnego kierowcę. Tabela przechowuje również informacje na temat bieżącej lokalizacji pojazdu, dzięki czemu wiadomo, gdzie znajduje się on w danym momencie.

Do tabeli „Breakdowns” będą dodawane wszelkie awarie pojazdów. Przechowuje ona dane na temat awarii dla wybranego pojazdu, a także datę jej wystąpienia i zakończenia. Menadżer floty będzie mógł więc wprowadzać szczegóły na temat awarii, a także analizować czas ich trwania.

Tabela „Orders” zawiera informacje o zleceniach. W skład jednego zlecenia może wchodzić wiele dostaw, które będą realizowane przez kierowców. Tabela przechowuje wartość logiczną „IsCompleted”, która informuje czy dane zlecenie zostało zakończone.

Tabela „Deliveries” przechowuje dane o dostawach. Dzięki niej możliwe jest chociażby przechowywanie informacji na temat miejsca początkowego i docelowego danej dostawy wraz z opisem przewożonego towaru. Rozpoczęcie oraz zakończenie dostawy realizowane jest za pomocą znaczników czasu, dzięki czemu znana jest dokładna data i godzina wystąpienia danego zdarzenia. W tej tabeli przechowywane są również dane klienta, które mogą być pomocne kierowcy w celu skontaktowania się z nim.

Systemowa tabela „_EFMigrationsHistory” przechowuje informacje o wykonanych migracjach bazy danych. Każda migracja jest zapisywana w osobnym wierszu tej tabeli, co pozwala synchronizować kod z bazą danych.

Rozdział 5

Przegląd wybranych narzędzi i technologii

W tym rozdziale opisano niektóre technologie oraz narzędzia, które wykorzystano podczas tworzenia autorskiej aplikacji do zarządzania flotą pojazdów. Pokazano ich cechy, właściwości, a także szereg możliwości i zalet, jakie posiadają, tym samym uzasadniając ich wybór.

5.1 Język C#

Do stworzenia backendu został zastosowany język C#. Jest to język programowania wysokiego poziomu stworzony przez firmę Microsoft. Niewątpliwą zaletą tego języka jest to, że jest on częścią frameworka .NET, który znacznie przyspiesza tworzenie aplikacji. Dzięki temu możliwe jest wykorzystanie gotowych narzędzi i bibliotek, bez potrzeby ich ręcznego pisania.

Język C# posiada statyczne typowanie, dzięki czemu błędy można wychwycić już na samym etapie pisania kodu (podczas kompilacji). Język ten oferuje korzystanie z mechanizmów programowania asynchronicznego przez zastosowanie słów kluczowych `async` oraz `await`, co pozwala na płynną obsługę aplikacji. Wykorzystano je np. podczas implementacji WebSockets, które umożliwiły śledzenie pojazdów w czasie rzeczywistym. C# to również język, który zezwala na stosowanie technologii LINQ (z ang. Language Integrated Query), co jest znacznym ułatwieniem. Dzięki temu możliwe jest pisanie zwięzłych zapytań bazujących na kolekcjach (np. filtrowanie lub sortowanie), przypominających składnię języka SQL (z ang. Structured Query Language). W projekcie skorzystano z tych udogodnień np. przy pobieraniu wszystkich dostępnych pojazdów, co zredukowało ilość kodu i zwiększyło jego czytelność. Język C# użyty został też do tworzenia modeli, które reprezentują rzeczywiste obiekty (np. pojazdy). Dzięki temu możliwe jest łatwe i szybkie rozszerzanie aplikacji o nowe funkcjonalności. Dodatkowo ten język

wykorzystuje system obsługi wyjątków, który znalazł zastosowanie w projekcie. Pozwala to na łatwe wychwycenie błędów i poprawę jakości ogólnego użytkowania. Język C# umożliwia również tworzenie typów wyliczeniowych. Zastosowano je w aplikacji w celu reprezentowania stałych wartości dla typów pojazdów lub ich statusów. Zastosowanie tego rozwiązania zapewnia używanie tylko zdefiniowanych wartości oraz zwiększa czytelność kodu.

Język C# oferuje więc wiele nowoczesnych rozwiązań, takich jak np. programowanie asynchroniczne (brak blokowania wątku głównego podczas wykonywania operacji), a zastosowanie technologii LINQ umożliwia tworzenie czytelnych zapytań na zbiorach danych. Po analizie zagadnienia podjęto decyzję, że skutecznie sprawdzi się podczas tworzenia aplikacji do zarządzania flotą pojazdów.

5.2 Framework ASP.NET Core

ASP.NET Core to framework stworzony przez firmę Microsoft służący do tworzenia nowoczesnych aplikacji internetowych. Umożliwia on tworzenie aplikacji z wykorzystaniem Web API (z ang. Web Application Programming Interfaces). Interfejs ten stosuje metody HTTP (z ang. Hypertext Transfer Protocol) do wymiany danych. Skorzystano z niego przy tworzeniu aplikacji - frontend komunikuje się z backendem wysyłając żądania HTTP (np. GET, POST, PUT), które można wykorzystać do różnych celów np. wyświetlenia listy pojazdów, dodawania nowego kierowcy czy rozdzielenia zestawu.

Framework dostarczany jest jako część platformy .NET. W aplikacji skorzystano z .NET 8.0, najnowszej wersji LTS (z ang. Long Term Support) z długoletnim wsparciem. Framework ASP.NET Core posiada natywne wsparcie dla języka C#, dzięki czemu wszystkie funkcjonalności są tworzone w tym języku. Niewątpliwą jego zaletą jest wstrzykiwanie zależności, co umożliwia łatwe przenoszenie obiektów między klasami. Dzięki temu potrzebne serwisy i konteksty są dostarczane tam, gdzie są wymagane, bez potrzeby tworzenia ich samodzielnie w klasach. Ułatwia to znacznie pracę i przyspiesza pisanie kodu. Istotnym elementem w aplikacji do zarządzania flotą pojazdów jest utrzymywanie stałego

połączenia między serwerem a klientem. Protokół WebSocket umożliwia obustronną komunikację i wymianę danych, dzięki czemu wszyscy podłączeni klienci mogą otrzymywać na bieżąco informacje np. o położeniu pojazdów na mapie. Wówczas ograniczamy liczbę zapytań oraz obciążenie serwera, ponieważ WebSocket nie wymaga wysyłania żądania, aby uzyskać odpowiedź. Framework ASP.NET Core oferuje wbudowane wsparcie dla WebSockets. Gdyby nie to udogodnienie, należałoby dokonywać dodatkowej konfiguracji lub instalować odpowiednie biblioteki. Użycie tego frameworka umożliwia również dodawanie atrybutów np. do klas lub metod. W aplikacji wykorzystano je do routingu API, dzięki czemu wiadome było, która metoda w kontrolerze odpowiada za dane żądanie HTTP. Przy tworzeniu aplikacji do zarządzania flotą pojazdów zastosowano również mechanizm CORS, który wpływa na bezpieczeństwo aplikacji i jest udostępniany w ramach frameworku ASP.NET Core. Z powodu innych portów dla frontendu oraz backendu, skonfigurowano politykę CORS zezwalającą na żądania pochodzące z różnych źródeł. Framework oferuje również wiele innych wbudowanych mechanizmów, które wykorzystano przy tworzeniu projektu np. obsługę uwierzytelniania użytkowników i przyznawania odpowiednich ról (ASP.NET Core Identity).

ASP.NET Core posiada wiele udogodnień, które mogą znacząco przyspieszyć cały proces pisania kodu, a protokół WebSocket jest kluczowy do utrzymania stałego połączenia między serwerem a klientem, co jest niezwykle ważne podczas tworzenia systemów wymagających komunikacji w czasie rzeczywistym. Po analizie stwierdzono, że ten framework okaże się dobrym wyborem do budowy autorskiej aplikacji do zarządzania flotą pojazdów.

5.3 Serwer baz danych PostgreSQL

PostgreSQL to popularny i wydajny system do zarządzania relacyjną bazą danych. Znany jest również pod nazwą Postgres. Działa na zasadzie otwartej licencji, więc nie ma potrzeby ponoszenia żadnych opłat licencyjnych.

PostgreSQL posiada wszelkie niezbędne typy danych, które były potrzebne do stworzenia aplikacji do zarządzania flotą pojazdów m.in. numeryczne,

łańcuchowe czy czasowe. PostgreSQL zapewnia transakcje ACID (skrót od angielskich słów Consistency - niepodzielność, Atomicity - atomowość, Isolation - izolacja, Durability - trwałość). Dzięki temu możliwe jest zachowanie poprawności danych. Przykładowo, atomowość zapewnia, że zostaną wykonane bezbłędnie wszystkie operacje, a w przeciwnym razie żadna. Ten system stawia więc na bezpieczeństwo i zapewnienie spójności danych. Każdy użytkownik z niego korzystający ma dostęp do informacji udostępnianych przez innych członków społeczności. PostgreSQL udostępnia publiczną, rozbudowaną dokumentację, która może pomóc w rozwiązywaniu problemów powstałych podczas instalacji lub konfiguracji. Ten system stawia również na wydajność. Zapewnia wysokie prędkości odczytu i zapisu danych, które przy dużej ilości danych są niezwykle ważne. Jego zaletą jest również możliwość indeksowania, co przekłada się na szybsze wyszukiwanie w dużych zbiorach danych. PostgreSQL można łatwo wdrożyć do frameworka ASP.NET Core, dzięki instalacji tylko jednego pakietu (Npgsql.EntityFrameworkCore.PostgreSQL). Dodatkowo, dzięki narzędziu pgAdmin 4 (które wykorzystano podczas tworzenia aplikacji), można w łatwy sposób wyświetlać potrzebne informacje w formie graficznej i zarządzać bazą danych.

Praca z bazą danych ma spełniać wymagania użytkownika, a także być przyjemna i nie sprawiać problemów. PostgreSQL zapewnia wszystko, co niezbędne do stworzenia aplikacji do zarządzania flotą pojazdów np. odpowiednie typy danych, wysoką wydajność czy specjalne narzędzie do graficznej reprezentacji danych. Po przeanalizowaniu zagadnienia stwierdzono, że PostgreSQL będzie odpowiednim wyborem do przechowywania danych w aplikacji, a narzędzie pgAdmin 4 umożliwi czytelne przeglądanie danych i ułatwi pracę z bazą danych.

5.4 Biblioteka React

React (również React.js lub ReactJS) to biblioteka JavaScript służąca do tworzenia interfejsów użytkownika. Do ich budowy używa komponentów (np. lista awarii lub formularz dodawania pojazdu), które są od siebie niezależne oraz mogą być używane wielokrotnie. Dzięki temu raz napisany komponent, może być użyty ponownie w różnych miejscach aplikacji. Daje to dużą oszczędność czasu,

ponieważ nie trzeba pisać ponownie tego samego kodu (pozytywnie wpływa to również na jego czytelność).

Biblioteka React została użyta do stworzenia frontendu autorskiej aplikacji do zarządzania flotą pojazdów. Jej wielką zaletą jest możliwość tworzenia dynamicznego interfejsu. Oznacza to, że nie jest odświeżana cała strona, a jedynie te fragmenty, które rzeczywiście uległy zmianie. Wpływa to pozytywnie na szybkość działania oraz umożliwia tworzenie aplikacji jednostronicowych w prosty i przyjemny sposób. Są one nazywane inaczej SPA (z ang. Single Page Application) i sprawiają, że cała aplikacja internetowa działa na jednej stronie, a przechodzenie pomiędzy widokami nie wymaga przeładowania. Jest to szczególnie ważne w przypadku systemów do zarządzania flotą pojazdów, gdzie po przejrzeniu dostępnych rozwiązań rynkowych zauważono, że zdecydowana ich większość to aplikacje typu SPA. Biblioteka React świetnie współpracuje z różnymi narzędziami np. Vite (przyspiesza uruchamianie i budowanie aplikacji) lub React-Leaflet (umożliwia łatwe wyświetlanie mapy i markerów), z których skorzystano przy tworzeniu aplikacji. Biblioteka React umożliwia łatwą komunikację z backendem. Dane mogą być pobierane i wysyłane dzięki użyciu zapytań HTTP, a potem wyświetlane dynamicznie w interfejsie użytkownika. Skorzystano z tych rozwiązań w aplikacji do zarządzania flotą pojazdów, dzięki czemu frontend mógł np. pobierać dane o pojazdach oraz kierowcach i wyświetlać je użytkownikowi. Zaletą biblioteki React jest jej pełna kompatybilność z językiem TypeScript (omawianym w podrozdziale 5.5). Dzięki temu można używać ich razem i nie trzeba stosować żadnych dodatkowych narzędzi. Bibliotekę React cechuje również aktywna społeczność, dzięki czemu łatwo jest znaleźć odpowiedzi na wszelkiego rodzaju problemy pojawiające się podczas tworzenia aplikacji.

Podsumowując, biblioteka React opiera się na tworzeniu komponentów, dzięki którym można szybko i wygodnie dodawać, modyfikować lub usuwać poszczególne elementy interfejsu, bez wpływu na pozostałą część aplikacji. Ta oraz inne wcześniej przedstawione zalety (np. możliwość tworzenia dynamicznych interfejsów i jednostronicowych aplikacji internetowych) pozwoliły na stwierdzenie, że jest to idealne rozwiązanie do stworzenia aplikacji będącej tematem ogólnych rozważań.

5.5 Język TypeScript

Język TypeScript to nadzbiór języka JavaScript. Zawiera więc wszystkie cechy JavaScriptu, ale wprowadza również pewne własne rozszerzenia. Został stworzony przez firmę Microsoft jako język programowania wysokiego poziomu.

Głównym powodem stosowania języka TypeScript jest poprawa bezpieczeństwa, a także jakości kodu. Ze względu na statyczne typowanie, możliwe jest sprawdzanie typu zmiennej na etapie samego pisania kodu. Dodatkowo język TypeScript wspiera tworzenie interfejsów, które określają, jakie typy danych oraz właściwości ma mieć dany obiekt. Pozwala to na bieżąco analizować błędy, jeszcze przed samym uruchomieniem aplikacji. Użycie tego języka pozwala wyeliminować sytuacje, w których do komponentu trafiają dane, które nie są poprawne. Jeśli tak się stanie, kompilator powiadomi o powstałym błędzie, w przeciwieństwie do języka JavaScript, gdzie taki błąd na etapie kodu mógłby nie zostać zauważony. Język TypeScript nie naraża użytkownika na szukanie ciężkich do znalezienia błędów. Jest to ogromna zaleta, ponieważ ewentualnie powstałe pomyłki mogą być szybciej rozwiązane, co oszczędza czas i przyspiesza programowanie. Po powrocie do kodu (po dłuższej nieaktywności), można zapomnieć wiele istotnych funkcjonalności aplikacji. Dzięki jasno zdefiniowanym zasadom (typom i interfejsom) nie jest to problem, ponieważ z łatwością można przypomnieć sobie całą strukturę danych. Język ten umożliwia współpracę z nowoczesnymi narzędziami i bibliotekami np. React, co sprawia, że pisanie kodu staje się niezwykle wygodne. Jest on również dobrze zintegrowany do pracy w popularnych środowiskach programistycznych np. Visual Studio lub Visual Studio Code, dzięki czemu możliwe jest korzystanie z dodatkowych udogodnień np. szybkie przejście do definicji czy podpowiedzi składni.

Język TypeScript ułatwia więc tworzenie aplikacji, zwiększając jednocześnie bezpieczeństwo przez zastosowanie statycznego typowania. Wiele błędów może być wykrytych już na samym etapie pisania kodu, co oszczędza czas i eliminuje problemy, które mogłyby powstać przez brak ich wykrycia (co może zdarzyć się np. w JavaScript). Z powodu szybkiego odnajdywania pomyłek, jak i innych zalet np. jasnego określenia struktur danych za pomocą interfejsów, zdecydowano, że jest to dobry i praktyczny wybór do tworzenia aplikacji do zarządzania flotą pojazdów.

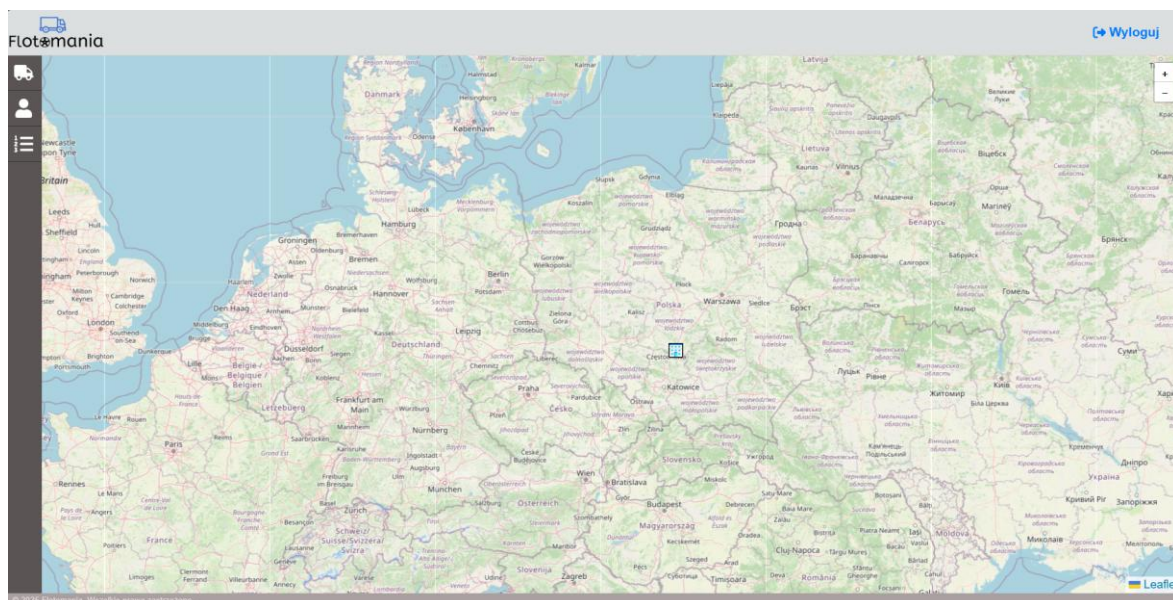
Rozdział 6

Prezentacja projektu

W tym rozdziale przedstawiono autorski sposób realizacji aplikacji do zarządzania flotą pojazdów. Opisane zostaną poszczególne funkcjonalności aplikacji i przedstawione w formie zrzutów ekranu dla lepszego zobrazowania. Ze względu na koszty urządzeń telematycznych, jak i ogólną specyfikę tematu (konieczność posiadania wielu pojazdów oraz realnego pokonywania tras w celu zbierania danych), zdecydowano się nie pobierać danych z modułu telematycznego, zamontowanego w pojeździe. Zaprojektowany system jest prototypem, stąd realizacja zleceń jest symulowana (kierowcy pojazdów nie pokonują realnych tras w celu realizacji dostaw). Pozwala to na prezentację funkcjonalności systemu, które nie mogłyby zostać omówione, bez zastosowania takiego rozwiązania.

6.1 Opis strony dla menadżera

Na Rys. 6.1 przedstawiono wygląd strony głównej aplikacji dla menadżera floty pojazdów.



Rys. 6.1 Strona główna aplikacji (menadżer).

Widoczna jest na tutaj mapa, która jest centralnym punktem oraz znajdujące się po lewej stronie menu boczne („Pojazdy”, „Kierowcy” oraz „Dostawy”).

Na rysunku (Rys. 6.1) w centralnej części można zauważyć również obiekt przypominający budynek. Jest to baza, czyli główna siedziba firmy transportowej. Stąd właśnie ruszać będą kierowcy w celu realizacji dostaw.

Po wybraniu „Pojazdy” z bocznego menu, otrzymujemy listę obecnych pojazdów we flocie, z możliwością filtracji po statusie („Niedostępny”, „Dostępny”, „W trasie”, „Powrót do bazy”, „Uszkodzony”) oraz typie („Ciężarówka siodłowa”, „Naczepa”, „Zestaw”, „Ciężarówka sztywna”, „Samochód dostawczy”). Jednak, aby mieć tutaj widoczny jakikolwiek pojazd, najpierw trzeba go dodać. Umożliwia to niewielki przycisk (+) znajdujący się po prawej stronie tego menu. Dodanie pojazdu wymaga wpisania podstawowych informacji w zależności od wybranego typu. Na przykład dla ciężarówki siodłowej będą to: numer rejestracyjny, marka, model, rok produkcji, natomiast dla samochodu dostawczego będzie to jeszcze dodatkowo ładowność, a także możliwość wyboru typu zabudowy (plandeka/firana, furgon, kontener, wywrotka, doka, chłodnia, izoterma, autolaweta). Dla lepszego zobrazowania, formularz dodawania pojazdu został przedstawiony na Rys. 6.2.

Rys. 6.2 Formularz dodawania pojazdu.

Po wypełnieniu wszystkich danych i kliknięciu przycisku „Dodaj”, w kodzie wywoływana jest metoda `AddVehicle`. Jak wcześniej wspomniano, cały backend (w związku z tym również wspomniana metoda) oparty jest na technologii ASP.NET Core Web API. Frontend wysyła więc odpowiednie żądania HTTP, a backend je przetwarza i zwraca odpowiedź. W tym przypadku wysłano żądanie HTTP POST (wraz z wypełnionymi danymi pojazdu) do endpointu `/api/vehicle` w kontrolerze `VehicleController`. Dzięki temu backend mógł sprawdzić poprawność danych, zwracając odpowiedź w czytelnym formacie JSON (z ang. JavaScript Object Notation). W przypadku braku błędów można było wstawić rekord do bazy, dokonać zapisu i zwrócić odpowiedź `201 Created` (zasób został utworzony). Kod przedstawiający te operacje (metoda `AddVehicle`) został zamieszczony poniżej (Listing 6.1).

```
[HttpPost]
public async Task<ActionResult<Vehicle>> AddVehicle([FromBody] Vehicle vehicle)
{
    if (vehicle == null)
    {
        return BadRequest("Vehicle data is required.");
    }
}
```

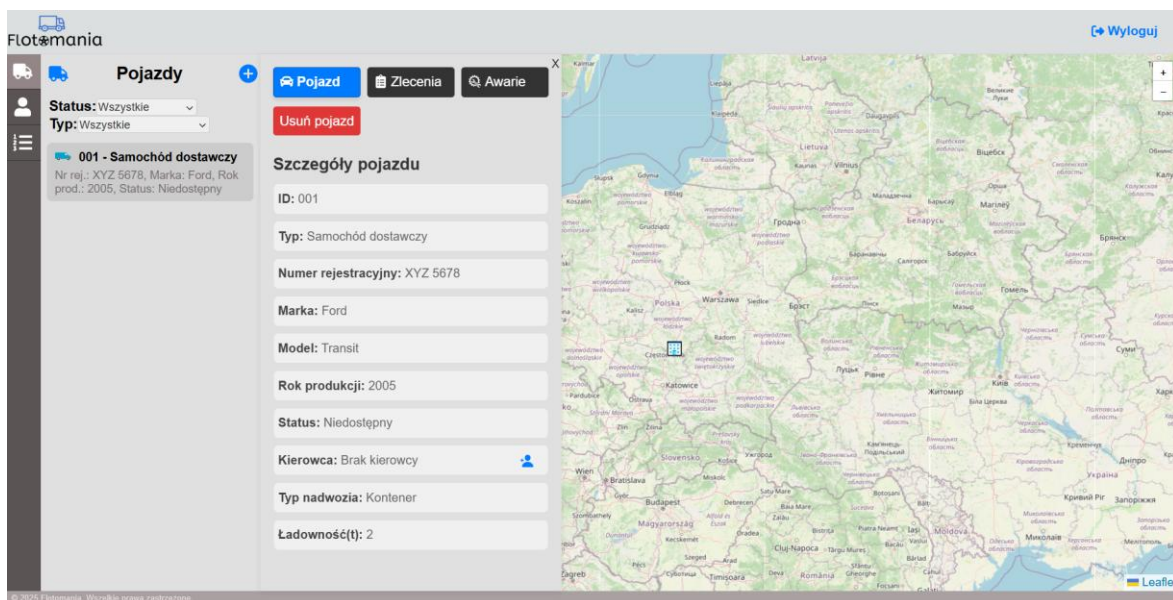
```

    }
    try
    {
        _context.Vehicles.Add(vehicle);
        await _context.SaveChangesAsync();
        return CreatedAtAction(nameof(GetVehicleById), new { id = vehicle.Id }, vehicle);
    }
    catch (Exception ex)
    {
        return StatusCode(500, $"Internal server error: {ex.Message}");
    }
}

```

Listing 6.1 Metoda obsługująca dodawanie pojazdu.

Po dodaniu pojazdu możemy przejść do wyświetlenia jego szczegółów. Znajdują się tutaj podstawowe informacje, które wpisaliśmy podczas dodawania pojazdu. Istnieje tutaj możliwość przydzielania kierowcy, a także usunięcia pojazdu. W przypadku ciągnika siodłowego oraz naczepy jest również opcja tworzenia zestawu (kolejno „Przydziel naczepę” oraz „Przydziel ciągnik siodłowy”). Okno szczegółów dla samochodu dostawczego z przykładowymi danymi przedstawia Rys. 6.3.



Rys. 6.3 Zawartość sekcji ze szczegółami pojazdu (samochód dostawczy).

Aby móc przydzielać zlecenia pojazd musi mieć kierowcę. Bez spełnienia tego warunku przycisk dodawania zlecenia jest niedostępny. Dodawanie kierowców realizujemy poprzez wybór „Kierowcy” w menu bocznym. Umożliwia to niewielka ikona „Dodaj kierowcę”, znajdująca się po prawej stronie. Po jej kliknięciu wyświetla się formularz umożliwiający wypełnienie danych. Został on przedstawiony na Rys. 6.4.

The screenshot shows a web application interface for 'Flot*mania'. On the left, there is a sidebar with a menu icon and a user icon. The main header area displays 'Kierowcy' and a message 'Nie znaleziono żadnych kierowców.' (No drivers found). In the top right corner, there is a 'Wyloguj' (Logout) button. A modal window titled 'Dodaj kierowcę' (Add driver) is open in the center. This modal contains five input fields: 'Imię:' (Name), 'Nazwisko:' (Surname), 'Email (Username):', 'Hasło:' (Password), and 'Numer telefonu:' (Phone number). Below these fields is a blue button labeled 'Dodaj' (Add). The modal also features a close button (X) in the top right corner.

Rys. 6.4 Formularz dodawania kierowcy.

Uzupełniamy tutaj podstawowe informacje na temat kierowcy - jego imię, nazwisko, email oraz numer telefonu. Dodatkowo jest tutaj pole do wpisania hasła, które będzie pomocne do poprawnego zalogowania się kierowcy. Po dodaniu kierowcy można zobaczyć jego dane, a dodatkowo status (domyślnie „Dostępny”, ponieważ kierowca nie wykonuje żadnej dostawy) oraz przydzielony pojazd (domyślnie brak pojazdu). Dostępna jest również akcja do usuwania kierowcy. Wszystko to obrazuje Rys. 6.5.

FlotaMania [Wyloguj]

Kierowcy

ID	Imię	Nazwisko	Email	Numer telefonu	Status	Pojazd	Akcje
1	Jan	Nowak	jannowak@jan.pl	123456789	Dostępny	---	

© 2020 FlotaMania. Wszelkie prawa zastrzeżone.

Rys. 6.5 Efekt dodania przykładowego kierowcy.

Mając dodanego kierowcę można przypisać mu pojazd. Przechodząc do szczegółów pojazdu możemy wybrać ikonę postaci z plusem po lewej stronie. Po jej kliknięciu wybieramy z listy dodanego przed chwilą kierowcę, tym samym przydzielając mu pojazd. Zostało to zaprezentowane na Rys. 6.6.

FlotaMania [Wyloguj]

Pojazdy

Status: Wszystkie
Typ: Wszystkie

001 - Samochód dostawczy
Nr rej.: XYZ 5678, Marka: Ford, Rok prod.: 2005, Status: Niedostępny

Szczegóły pojazdu

ID: 001
Typ: Samochód dostawczy

Rok produkcji: 2005
Status: Niedostępny
Kierowca: Brak kierowcy
Typ nadwozia: Kontener
Ładowność(t): 2

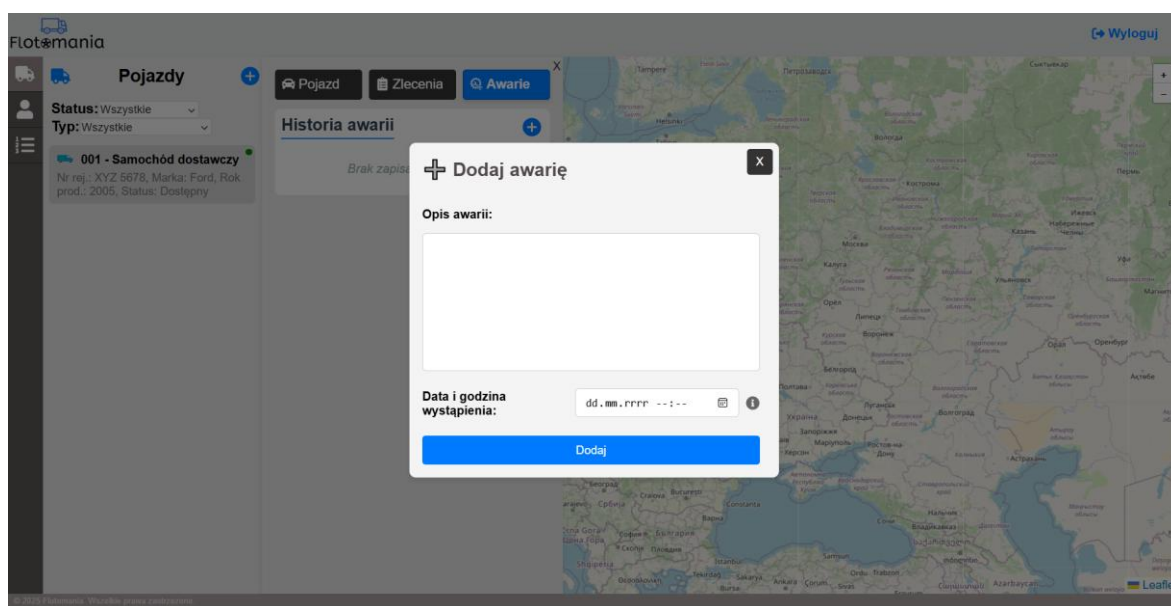
Wybierz kierowcę (bez przypisanego pojazdu):

ID	Imię	Nazwisko	Email	Numer telefonu	Akcja
1	Jan	Nowak	jannowak@jan.pl	123456789	Wybierz

© 2020 FlotaMania. Wszelkie prawa zastrzeżone.

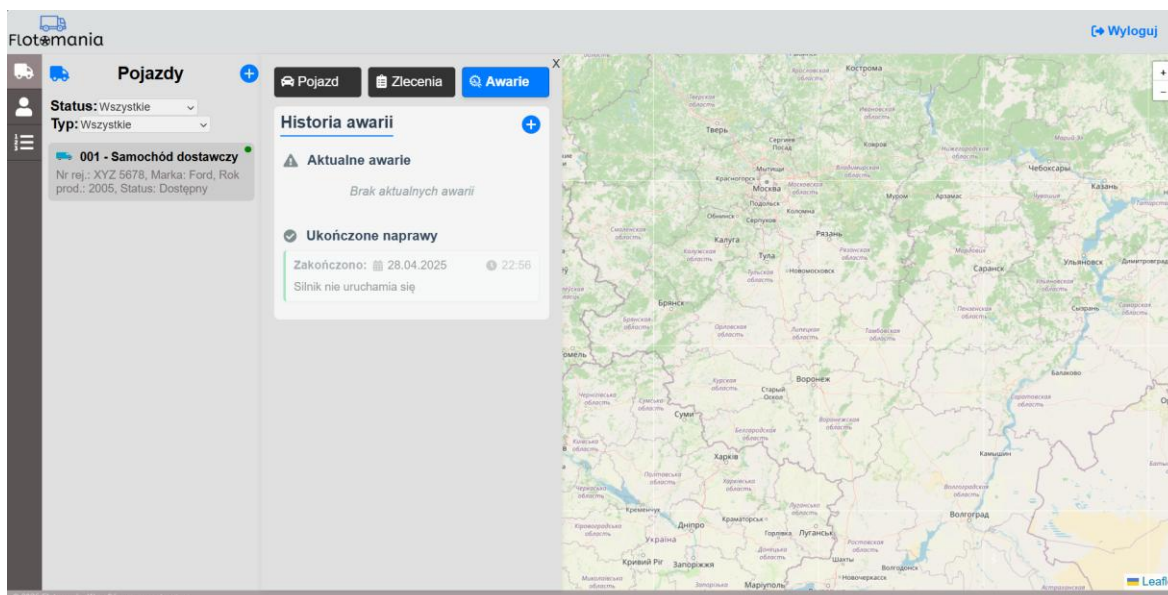
Rys. 6.6 Widok wyboru kierowcy.

Po wybraniu kierowcy, pojazd staje się dostępny (pojawia się obok niego zielona kropka). Można również w tym momencie usunąć przypisanie kierowcy do pojazdu (czerwona ikonka obok kierowcy), a także zmienić kierowcę (niebieskie strzałki symbolizujące wymianę). Po tym wszystkim można przystąpić do dodawania zlecenia (pod warunkiem, że pojazd nie ma awarii). Samo dodawanie awarii odbywa się w zakładce „Awarie”, gdzie możemy wpisać opis awarii, a także datę i godzinę jej wystąpienia. Zobrazowane jest to na Rys. 6.7.



Rys. 6.7 Formularz dodawania awarii.

Dodaną awarię możemy również zakończyć, wybierając przycisk „Zakończ”. Wtedy trafia ona do sekcji „Ukończone naprawy”. Efekt końcowy tego działania został zaprezentowany na Rys. 6.8.



Rys. 6.8 Widok zakładki awarii.

Dodawanie zlecenia nie jest możliwe dla samego ciągnika siodłowego oraz samej naczepy. Jest to możliwe jedynie dla całości (zestawu), ciężarówki sztywnej oraz samochodu dostawczego. Wynika to z logicznych zasad działania transportu. Jeśli więc spełnione jest to wymaganie, czyli jest wybrany pojazd umożliwiający podróż (oraz jak wspomniano wcześniej pojazd nie ma awarii), może wyruszyć on w drogę. Odbывается to w zakładce „Zlecenia”, gdzie znak plusika (+) symbolizuje dodawanie zlecenia. Po jego naciśnięciu menadżer floty może uzupełnić podstawowe informacje o zleceniu. Wpisuje tu opis towaru (co kierowca ma przewozić), a także miejsce początkowe i docelowe podróży. Polami opcjonalnymi są imię i nazwisko oraz telefon klienta, które mają być pomocne kierowcy w kontakcie z klientem. Po wpisaniu danych miejsc (początkowego oraz docelowego) istnieje opcja wyboru alternatywnych tras. Jest to możliwe dzięki wysłaniu zapytania do OSRM (z ang. Open Source Routing Machine). Jest to biblioteka, która wyznacza trasę pomiędzy punktami i zwraca odpowiedź na temat możliwych tras, dystansu czy też czasu przejazdu. Skorzystano z niej w metodzie `GetRouteCoordinatesAsync` w klasie `RouteService`, co pozwoliło wyznaczyć współrzędne trasy między dwoma punktami, które będą przydatne podczas symulacji ruchu pojazdu (umożliwią przemieszczanie się pojazdu po mapie). Metoda została przedstawiona na Listingu 6.2.


```

public async Task<List<(double lat, double lng)>> GetRouteCoordinatesAsync(double startLng, double startLat, double endLng, double endLat, int? selectedRouteIndex = null)
{
    using (HttpClient client = new HttpClient())
    {
        var url = string.Format(System.Globalization.CultureInfo.InvariantCulture,
            "https://router.project-osrm.org/route/v1/driving/{0},{1};{2},{3}?alternatives=2&geometries=geojson&overview=full",
            startLng, startLat, endLng, endLat);

        var response = await client.GetAsync(url);
        if (!response.IsSuccessStatusCode)
        {
            throw new InvalidOperationException($"Cannot retrieve route. OSRM returned: {response.StatusCode}");
        }

        var json = await response.Content.ReadAsStringAsync();
        var data = JsonSerializer.Deserialize<OsrmsResponse>(json);

        if (data == null || data.routes == null || data.routes.Length == 0)
        {
            throw new InvalidOperationException("No route could be retrieved from OSRM.");
        }

        int index = selectedRouteIndex.HasValue ? selectedRouteIndex.Value : 0;
        var coordinates = data.routes[index].geometry.coordinates;
        var routeCoords = new List<(double lat, double lng)>();

        foreach (var coord in coordinates)
        {
            routeCoords.Add((Math.Round(coord[1], 6), Math.Round(coord[0], 6)));
        }

        return routeCoords;
    }
}

```

Listing 6.2 Metoda zwracająca listę współrzędnych między punktami.

Przykładowo wypełnione dane wraz z wyborem odpowiedniej trasy zostały przedstawione na Rys. 6.9.

Rys. 6.9 Formularz dodawania zlecenia.

Wybór przycisku „Dodaj” rozpoczyna symulację trasy. Po dodaniu zlecenia kierowca wyrusza w drogę. Startuje z bazy i zmierza w kierunku miejsca początkowego (adres wpisany parę chwil wcześniej), aby dokonać załadunku towaru. W kodzie wywoływana jest tutaj metoda `SimulateRouteAsync` (`vehicleId`, `routeCoordinates`, `cancellationToken`), która iteruje po kolejnych współrzędnych trasy. Dzięki temu możliwe będzie przesuwanie pojazdu na mapie, ponieważ jego pozycja będzie aktualizowana w bazie danych co 3 sekundy. Metodę przedstawiono na Listingu 6.3.

```
private async Task SimulateRouteAsync(int vehicleId, List<(double lat, double lng)> routeCoordinates, Cancellation_token cancellationToken)
{
    if (routeCoordinates == null || routeCoordinates.Count == 0)
    {
        throw new InvalidOperationException($"No route coordinates for vehicle ID {vehicleId}.");
    }
}
```

```

using var scope = _scopeFactory.CreateScope();
var dbContext = scope.ServiceProvider.GetRequiredService<ApplicationDbContext>();

foreach (var point in routeCoordinates)
{
    try
    {
        cancellationToken.ThrowIfCancellationRequested();

        var vehicle = await dbContext.Vehicles.FindAsync(new object[] { vehicleId },
            cancellationToken);
        if (vehicle != null)
        {
            await dbContext.Entry(vehicle).ReloadAsync(cancellationToken);

            if (vehicle.Status == VehicleStatus.Damaged)
            {
                return;
            }

            vehicle.Latitude = Math.Round(point.lat, 6);
            vehicle.Longitude = Math.Round(point.lng, 6);
            vehicle.LastModified = DateTime.UtcNow;

            await dbContext.SaveChangesAsync(cancellationToken);
        }

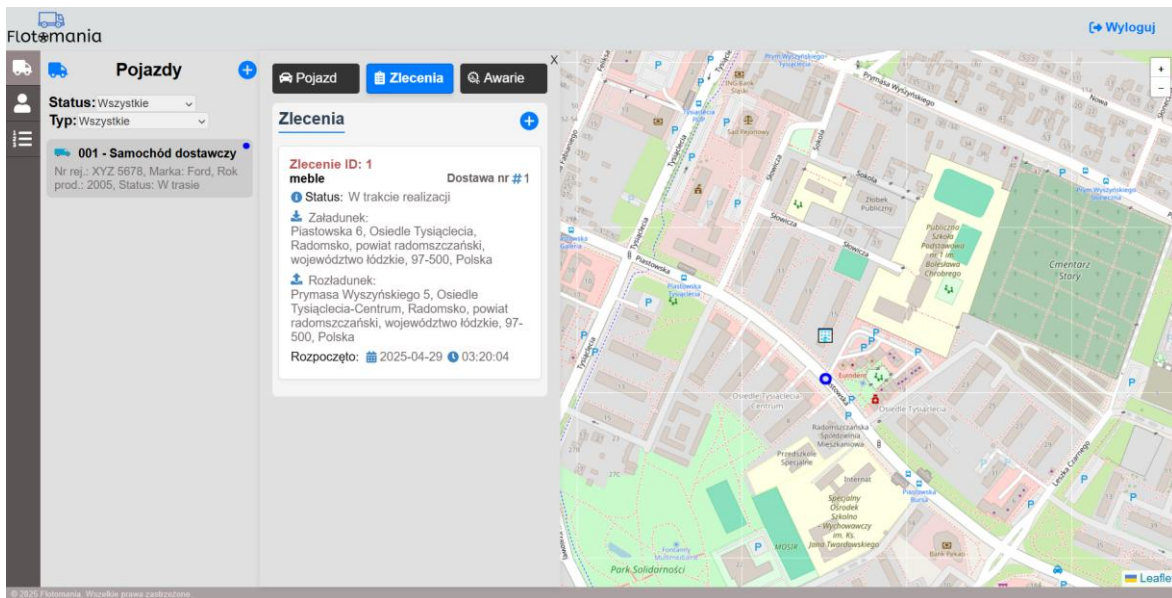
        await Task.Delay(3000, cancellationToken);
    }
    catch (TaskCanceledException)
    {
        return;
    }
}
}

```

Listing 6.3 Metoda realizująca symulację trasy - iteracja po punktach.

Pojazd po osiągnięciu miejsca początkowego, następnie udaje się w kierunku miejsca docelowego, korzystając z tej samej funkcji (zmieniają się tylko

współrzędne, ponieważ trasa jest już inna). Pojazd (oznaczony niebieskim okręgiem) zmierzający w kierunku celu pokazuje Rys. 6.10.



Rys. 6.10 Pojazd rozpoczynający dostawę.

W tym momencie można dodawać kolejne dostawy, a pojazd po ukończeniu poprzedniej, natychmiast będzie kierował się w stronę kolejnej. Zwiększa to w realnych warunkach efektywność i ogranicza „puste przebiegi”, które wiążą się z pokonywaniem tras bez ładunku. W tym przypadku więc, pojazd nie będzie wracał do bazy i zaczynał kolejnej dostawy, a zrobi to od razu po zakończeniu poprzedniej dostawy (zwiększa to efektywność firmy, oszczędzając czas czy też paliwo). Dodatkowo każdą dostawę (oprócz tej obecnie realizowanej) można anulować - wybierając przycisk „Anuluj”. Wówczas nie będzie ona wykonywana przez kierowcę pojazdu.

W zakładce „Dostawy” (menu boczne po lewej stronie) możemy przeglądać aktualne, anulowane i ukończone dostawy. Dla każdej z nich wyświetlane są potrzebne menadżerowi informacje, takie jak np. kierowca wykonujący dostawę czy też miejsce startowe i końcowe dostawy. Zobrazowano to na Rys. 6.11.

ID	Pojazd	Kierowca	Ładunek	Start	Cel	Zlecenie	Status	Rozpoczęto	Zakończono
1	001 – Samochód dostawczy	Jan Nowak	meble	Piastowska 6, Osiedle Tysiąclecia, Radomsko, powiat radomszczański, województwo łódzkie, 97-500, Polska	Prymasa Wyszynskiego 5, Osiedle Tysiąclecia-Centrum, Radomsko, powiat radomszczański, województwo łódzkie, 97-500, Polska	1	Ukończona	29.04.2025, 03:20:04	29.04.2025, 03:22:17

Rys. 6.11 Widok zakładki „Dostawy”.

6.2 Opis strony dla kierowcy

Strona dla kierowcy jest mniej rozbudowana. Znajdują się tutaj podstawowe informacje, które umożliwiają współpracę kierowcy z menadżerem floty. Główny widok strony dla kierowcy przedstawia Rys. 6.12.

Flotamania

Dostawy

Informacje

Wyloguj

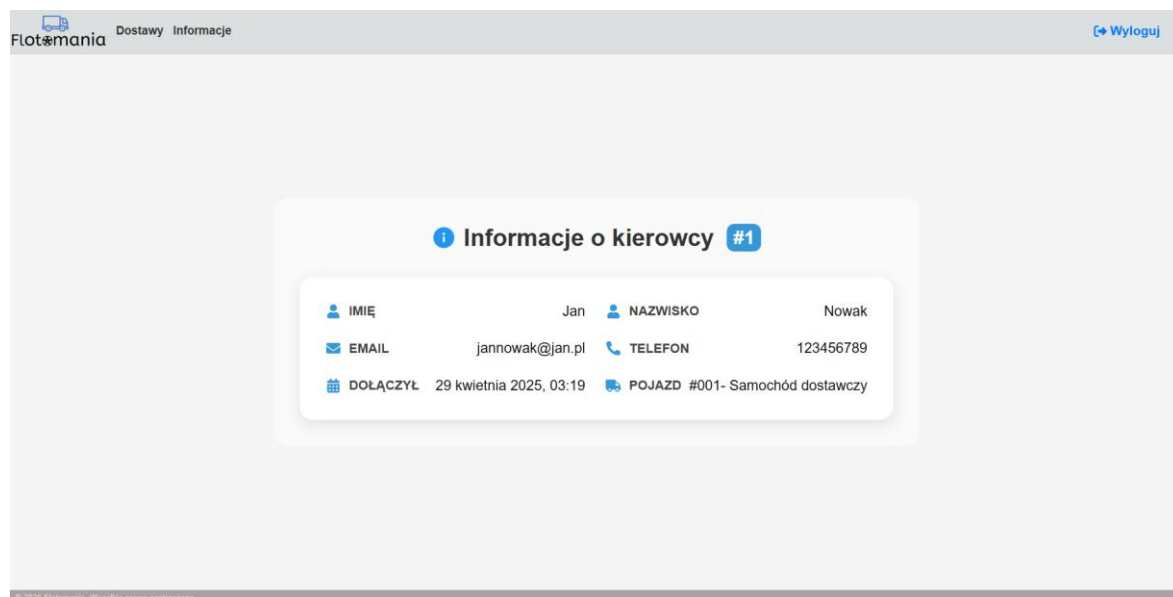
Twoje dostawy

ID	Zlecenie	Start	Cel	Rozpoczęto	Zakończono	Ładunek	Status	Klient	Telefon Klienta
1	1	Piastowska 6, Osiedle Tysiąclecia, Radomsko, powiat radomszczański, województwo łódzkie, 97-500, Polska	Prymasa Wyszyńskiego 5, Osiedle Tysiąclecia-Centrum, Radomsko, powiat radomszczański, województwo łódzkie, 97-500, Polska	29.04.2025, 03:20:04	29.04.2025, 03:22:17	meble	Ukończona	-	-

Rys. 6.12 Główny widok strony dla kierowcy.

Na powyższym rysunku (Rys. 6.12) można zauważyć dodane wcześniej przez menadżera zlecenie, w skład którego wchodzi ta dostawa. Kierowca ma na bieżąco wgląd w swoje dostawy (te w trakcie realizacji, anulowane oraz zrealizowane). Widać tutaj informacje o np. punkcie startowym („Start”), punkcie docelowym („Cel”) oraz danych klienta, które na tym rysunku są puste ze względu na ich opcjonalność podczas dodawania zlecenia.

Z górnego menu nawigacyjnego dostępna jest również opcja wyboru sekcji „Informacje”. Po przejściu do niej kierowca ma możliwość przeglądania osobistych informacji, takich jak imię, nazwisko, email, telefon, data i godzina dołączenia, a także przypisany pojazd. Przedstawia to Rys. 6.13.



Rys. 6.13 Efekt wyboru sekcji „Informacje”.

Rozdział 7

Podsumowanie

W pracy omówiono zagadnienie zarządzania flotą pojazdów w firmach transportowych. Wyjaśniono podstawowe pojęcia w tym zakresie, opisano działanie systemów tworzonych do tych celów oraz ich wpływ na codzienną pracę właścicieli firm transportowych, menadżerów flot czy kierowców. Zauważono, że obecność takich systemów może istotnie ułatwić prowadzenie działalności i poprawić sytuację finansową firmy np. umożliwiając one zarządzanie pojazdami oraz kierowcami (kierowcy mogą również logować się na swoje konto i wykonywać powierzone zadania), dając wgląd w historię awarii pojazdów (co pozwala ocenić ich stopień niezawodności) czy pozwalając na wybór optymalnej trasy przejazdu (oszczędność paliwa, co wpływa na redukcję kosztów). Przedstawiono również architekturę systemów zarządzania flotą pojazdów, opisując przy tym sposób przesyłu informacji z pojazdu na ekran menadżera floty. Dodatkowo opisano niektóre z aplikacji dostępnych na rynku, zwracając uwagę na cechy takich systemów i ich standardy, co pozwoliło sprawniej zaprojektować autorską aplikację.

Wszelkiego rodzaju projekty, bez wcześniejszego zaprojektowania planu są chaotyczne i nieefektywne. Gdy projekt nie ma określonych reguł, łatwo jest popełnić błędy i zapomnieć o wielu istotnych funkcjach, przez co traci on na jakości. Aby wyeliminować te problemy opracowano założenia projektowe, w których zawarto wytyczne odnośnie tworzonej aplikacji i przedstawiono plan projektu. W pracy dodatkowo dokonano przeglądu wybranych narzędzi i technologii użytych do tworzenia aplikacji. Przedstawiono zalety i uzasadniono wybór, co pokazało, że postawienie właśnie na te narzędzia i technologie było właściwe.

Ostatecznie dokonano prezentacji projektu, przedstawiając stronę dla menadżera floty i kierowcy. Postanowiono nie decydować się na zakup urządzeń telematycznych ze względu na ich ceny, a także ograniczenia logistyczne np. obowiązek posiadania wielu pojazdów czy pokonywania kilometrów w realnych

warunkach w celu zbierania danych. Zastosowano więc symulację dostaw, która pozwoliła realizować wirtualne zlecenia. Dzięki temu możliwe było przeprowadzenie testów funkcjonalności aplikacji, eliminując tym samym konieczność posiadania rzeczywistych urządzeń i pojazdów. Dzięki temu opracowano prototyp, który może być w przyszłości rozbudowany o dodatkowe funkcje.

Analizując cel pracy, pomyślnie omówiono zagadnienie zarządzania flotą pojazdów w firmie transportowej. Dodatkowo udało się zaprojektować i wdrożyć aplikację, która umożliwia zarządzanie pojazdami, kierowcami i zleceniami. Zdecydowano się na rozwiązanie symulacyjne (prototyp) ze względu na brak użycia urządzeń telematycznych (koszty oraz ograniczenia logistyczne wynikające z obowiązku posiadania wielu pojazdów czy pokonywania realnych tras). Umożliwiło to realizację dostaw z możliwością sprawdzenia bieżącej lokalizacji pojazdu na mapie. Praca miała również wprowadzić refleksję nad tym, jak aplikacja do zarządzania flotą pojazdów w firmie transportowej, może wpłynąć na komfort pracy i ułatwić codzienne obowiązki właścicielom firm transportowych, menadżerom flot i kierowcom. Zauważono, że wpływ jest znaczący, bowiem „papierologia” jest znacząco zredukowana, a wszystko czego potrzebuje właściciel firmy i menadżer floty (wymagania mogą być różne i należy dostosować system indywidualnie do potrzeb) jest umieszczone w ramach jednej aplikacji. Wówczas zarządca wie np. jakie ma pojazdy w swojej flocie, w jakim są stanie oraz może wybierać optymalną trasę przejazdu dla kierowców. Właściciel firmy może analizować te dane i wyciągać wnioski, które mogą przyczynić się do zmniejszenia wydatków. Kierowcy z kolei mogą realizować zadania, logując się na swoje konto. Przyspiesza to cały proces wymiany informacji, praca przebiega sprawniej i jest bardziej zorganizowana. Powyższa analiza pozwala więc stwierdzić, że cel pracy został w pełni zrealizowany.

Aplikacja w przyszłości może być rozbudowana o dodatkowe funkcje, które zwiększą jej użyteczność i efektywność. Przykładowo można zastosować integrację z urządzeniami telematycznymi, co pozwoli odczytywać informacje o np. prędkości, spalaniu czy temperaturze silnika, a także umożliwi wzbogacenie systemu o analizę danych np. na temat stylu jazdy kierowców czy średniego spalania paliwa. Stworzoną aplikację można rozbudować o różnego rodzaju raporty, które pozwolą ocenić stan i efektywność firmy np. ilość pojazdów we flocie, liczba wykonanych zleceń przez danego kierowcę czy pojazdy z największą liczbą usterek.

Bibliografia

- [1] Topolski M., *Planowanie optymalnej trasy przejazdu transportu samochodowego z wykorzystaniem miękkich metod obliczeniowych*, Autobusy. Technika, Eksploatacja, Systemy Transportowe, 2016, Tom 17, Nr 6, str. 1174–1179.
- [2] Świątecki P., *Pasażer dobrze poinformowany – aplikacje telematyczne dla przewozów pasażerskich*, Infrastruktura Transportu, 2012, Nr 6, str. 40–42.
- [3] Rabe. J., *Lane-Precise Localization with Production Vehicle Sensors and Application to Augmented Reality Navigation*, KIT Scientific Publishing, 2019, p.1.
- [4] Radius, *History of Telematics*, <https://www.radius.com/en-gb/telematics/explained/history/>, stan na dzień: 08.12.2025.
- [5] Data Bridge Market Research, *Global Fleet Management Market Size, Share, and Trends Analysis Report – Industry Overview and Forecast to 2032*, <https://www.databridgemarketresearch.com/reports/global-fleet-management-market>, stan na dzień: 08.12.2025.
- [6] Ustawa z dnia 20 czerwca 1997 r. *Prawo o ruchu drogowym* (t.j. Dz.U. z 2024 r. poz. 1251).
- [7] Szymczak M. (red.), *Słownik języka polskiego*, Warszawa: PWN, 1988, Tom 2, str. 767.
- [8] Wojskowe Centrum Rekrutacji w Bydgoszczy, *Klasyfikacja pojazdów wg rodzajów, podrodzajów i przeznaczenia*, https://wcrbydgoszcz.wp.mil.pl/u/pages/atts/2022/12/bKlasyfikacja_pojazdow.pdf, stan na dzień: 09.12.2025.
- [9] Kinnunen S-K., Happonen A., Marttonen-Arola S., Kärri T., *Traditional and extended fleets in literature and practice: Definition and untapped potential*, International Journal of Strategic Engineering Asset Management, 2019, Vol. 3, No. 3, pp. 239–261.
- [10] Redmer A., *Strategic vehicle fleet management – the replacement problem*, LogForum, 2016, Vol. 12, No. 1, pp. 17–24.
- [11] Kiba-Janiak M., Tronina P., *Wpływ systemów telematycznych na usprawnienie międzynarodowych łańcuchów dostaw*, Zeszyty Naukowe. Organizacja i Zarządzanie / Politechnika Śląska, 2017, Z. 103, str. 79–93.

- [12] Bielli M., Bielli A., Rossi R., *Trends in Models and Algorithms for Fleet Management*, Procedia - Social and Behavioral Sciences, 2011, Vol. 20, pp. 4-18.
- [13] Hidayat M., Kinoro I., *Comparative Efficiency of Fleet Management System Versus Transportation Management System on Transportation Vehicle Tracking System Efficiency*, The Management Journal of Binaniaga, 2023, Vol. 8, No. 2, p. 171.
- [14] Gáspár P., Szalay Z., Aradi S., *Chapter 14. Fleet Management Systems*, https://www.mogi.bme.hu/TAMOP/jarmurendszer_kiranyitasa_angol/math-ch14.html, stan na dzień: 14.12.2025.
- [15] Kaplan E., Hegarty C., *Understanding GPS/GNSS: Principles and Applications*, Artech House, 2017, pp. 1–10.
- [16] Rosfix, *Moduł GPS GY-NEO8MV2 z anteną*, <https://rosfix.pl/produkt/modul-gps-gy-neo8mv2-z-antena/>, stan na dzień: 14.12.2025.
- [17] TrackSystem, *Teltonika FM1010*, <https://tracksystem.pl/docs/lokalizatory/teltonika/teltonika-fm1010/>, stan na dzień: 14.12.2025.
- [18] Webfleet Solutions, *Fleet Management Software Webfleet Features*, https://www.webfleet.com/en_us/webfleet/products/webfleet/features/, stan na dzień: 14.12.2025.
- [19] Watts R., *Chevin FleetWave Review*, <https://www.pcmag.com/reviews/chevin-fleetwave>, stan na dzień 14.12.2025.

Spis rysunków

Rys. 3.1	Architektura systemów zarządzania flotą pojazdów [14].	14
Rys. 3.2	Moduł GPS z anteną (GY-NEO8MV2) [16].	15
Rys. 3.3	Moduł telematyczny Teltonika (FM1010) [17].	16
Rys. 3.4	Zrzut ekranu aplikacji Webfleet [18].	19
Rys. 3.5	Zrzut ekranu aplikacji FleetWave [19].	21
Rys. 4.1	Diagram przypadków użycia.	25
Rys. 4.2	Diagram sekwencji - dodawanie pojazdu.	27
Rys. 4.3	Schemat bazy danych.	29
Rys. 6.1	Strona główna aplikacji (menadżer).	40
Rys. 6.2	Formularz dodawania pojazdu.	41
Rys. 6.3	Zawartość sekcji ze szczegółami pojazdu (samochód dostawczy).	43
Rys. 6.4	Formularz dodawania kierowcy.	44
Rys. 6.5	Efekt dodania przykładowego kierowcy.	45
Rys. 6.6	Widok wyboru kierowcy.	45
Rys. 6.7	Formularz dodawania awarii.	46
Rys. 6.8	Widok zakładki awarii.	47
Rys. 6.9	Formularz dodawania zlecenia.	49
Rys. 6.10	Pojazd rozpoczynający dostawę.	51
Rys. 6.11	Widok zakładki „Dostawy”.	52
Rys. 6.12	Główny widok strony dla kierowcy.	53
Rys. 6.13	Efekt wyboru sekcji „Informacje”.	54

Spis listingów

Listing 6.1	Metoda obsługująca dodawanie pojazdu.....	41
Listing 6.2	Metoda zwracająca listę współrzędnych między punktami.....	48
Listing 6.3	Metoda realizująca symulację trasy - iteracja po punktach.....	49

Streszczenie

Praca skupia się na zaprojektowaniu i wdrożeniu aplikacji (prototypu) przeznaczonej dla firmy transportowej, która będzie umożliwiała podejmowanie kluczowych decyzji związanych z zarządzaniem pojazdami oraz kierowcami, a także transportem towarów.

Rozdział pierwszy pokrótce opisuje historię telematyki i systemów zarządzania flotą pojazdów. Pokazano tutaj istotną rolę, jaką pełnią takie systemy, a także udowodniono rosnący poziom zapotrzebowania na tego typu rozwiązania w najbliższych latach.

W rozdziale drugim przedstawiono cel oraz zakres pracy, co stanowi fundament całego projektu. Wypisano działania stanowiące zakres pracy, a ich pomyślna realizacja będzie oznaczała osiągnięcie wyznaczonego celu.

Rozdział trzeci wprowadza w tematykę zarządzania flotą pojazdów w firmie transportowej. Przedstawiono tutaj podstawowe pojęcia związane z zagadnieniem. Opisano również systemy do zarządzania flotą pojazdów - ich zasadę działania i wsparcie, jakie stanowią dla właścicieli firm transportowych, menadżerów flot i kierowców, a także przedstawiono istniejące rozwiązania rynkowe. Taka analiza pozwoliła określić, czego można oczekiwać od takich systemów, co było pomocne przy tworzeniu własnej tego typu aplikacji.

Rozdział czwarty przedstawia wszelkie założenia tworzonej aplikacji (projektowanie systemu). Pokazano tutaj, co system ma robić i jak powinien działać. W tym rozdziale wypisano wymagania funkcjonalne i нефункционалне, a także przedstawiono w formie graficznej oraz omówiono diagram przypadków użycia, diagram sekwencji i schemat bazy danych.

W rozdziale piątym dokonano przeglądu wybranych narzędzi i technologii użytych podczas tworzenia aplikacji do zarządzania flotą pojazdów. Zostały one pokrótce streszczone z wyszczególnieniem wielu zalet. Pozwoliło to upewnić się, że wybór właśnie tych konkretnych rozwiązań był właściwy podczas tworzenia aplikacji do zarządzania flotą pojazdów w firmie transportowej.

Rozdział szósty poświęcony jest części praktycznej - opisuje stworzoną aplikację. Przedstawiono tutaj jej wygląd, a także funkcjonalności np. zarządzanie pojazdami, kierowcami, awariami czy przydzielanie zleceń dla kierowców. Opisano stronę zarówno dla właściciela, jak i kierowcy, a wszystko zobrazowano zrzutami ekranu dla lepszego zrozumienia działania aplikacji.

Ostatni rozdział przedstawia rezultaty, a więc podsumowuje to, co zostało wykonane w pracy. Uzasadniono tutaj, że cel pracy został w pełni zrealizowany, a także przedstawiono kilka potencjalnych możliwości rozbudowy stworzonej aplikacji.

Summary

The thesis focuses on the design and implementation of an application (prototype) for a transport company, which will enable key decisions to be made regarding the management of vehicles and drivers, as well as the transport of goods.

The first chapter briefly describes the history of telematics and vehicle fleet management systems. It shows the important role that such systems play, and also proves the growing demand for this type of solution in the coming years.

The second chapter presents the objective and scope of the thesis, which form the foundation of the entire project. The activities constituting the scope of the thesis are listed, and their successful completion will mean the achievement of the stated objective.

The third chapter introduces the subject of vehicle fleet management in a transport company. It presents the basic concepts related to the issue. It also describes vehicle fleet management systems - how they work and the support they provide to transport company owners, fleet managers, and drivers, and presents existing market solutions. This analysis made it possible to determine what can be expected from such systems, which was helpful in creating our own application of this type.

The fourth chapter presents all the assumptions of the application being developed (system design). It shows what the system is supposed to do and how it should work. This chapter lists functional and non-functional requirements, as well as presents in graphical form and discusses the use case diagram, sequence diagram, and database schema.

The fifth chapter reviews selected tools and technologies used in the development of the vehicle fleet management application. They are briefly summarized, highlighting their many advantages. This ensured that the choice of these specific solutions was the right one for the development of a vehicle fleet management application for a transport company.

The sixth chapter is devoted to the practical part - it describes the created application. It presents its appearance and functionalities, such as vehicle, driver and breakdown management, or assigning orders to drivers. The website is described for both the owner and the driver, and everything is illustrated with screenshots for a better understanding of how the application works.

The last chapter presents the results, summarizing what has been accomplished in the thesis. It justifies that the objective of the thesis has been fully achieved and presents several potential possibilities for expanding the created application.

Słowa kluczowe

flota pojazdów, zarządzanie flotą pojazdów, pojazdy, kierowcy, zlecenia transportowe, planowanie tras, telematyka, system FMS, urządzenia telematyczne, ASP.NET Core

fleet of vehicles, vehicle fleet management, vehicles, drivers, transport orders, route planning, telematics, FMS system, telematics devices, ASP.NET Core