

BD2 - Firma transportowa

Kasper Hasior, Konrad Grzegorzczak, Mateusz Nawrot, Krzysztof Najda

Opis projektu

Firma transportowa zapewnia usługi w zakresie odbierania i dostarczania towarów zleconych przez klientów firmy. Klient przekazuje firmie wszelkie detale zlecenia. Firma przetwarza te dane, przypisuje do zlecenia odpowiednie pojazdy oraz kierowców i wykonuje zlecenia.

Zlecenie składa się z ilości towaru (wyrażonego w paczkach), opisu paczek, adresu nadania i dostarczenia oraz terminów złożenia i realizacji zlecenia. Każda z paczek jest scharakteryzowana przez wagę, wymiary i zawartość. Klient może podać informację o specjalnych wymaganiach co do paczki (np. towar może być wrażliwy, szybko psujący się, niebezpieczny). Zlecenie, w zależności od potrzeb, może być realizowane przez jeden lub kilka samochodów. Paczki ze zlecenia przewożone są w ramach kursów. W jednym kursie może być przewożonych od zera do wszystkich paczek z danego zlecenia. W danej chwili, jeden kierowca może obsługiwać jeden pojazd.

Na wyposażeniu firmy jest flota samochodów przewozowych składająca się z pojazdów różnego typu, o określonych wymiarach przestrzeni ładunkowej i maksymalnej ładowności. Pojazdy ponadto są przystosowane do przewozu określonego typu towarów (pojazd chłodniczy - towary szybko psujące się, towary niebezpieczne). Wykonując zlecenie, każdy pojazd ma przypisaną ilość przewożonych paczek w ramach tego zlecenia.

Kierowcy firmy transportowej posiadają uprawnienia do prowadzenia konkretnych typów pojazdów.

Zakładamy, że firma jest scentralizowana i każdy kurs pojazdu zaczyna się w tym samym miejscu, co kończy (siedziba firmy). W ramach kursu możemy mieć następujące akcje:

- odbiór paczki,

- wydanie towaru,
- przejazd do punktu A do punktu B.

W bazie przechowywane są wszystkie wykonane akcje, w tym wszystkie wykonane przejazdy. Każdy przejazd zawiera zbiór odcinków drogi wyznaczający go. Zbiór ten nie musi zawierać wszystkich fizycznie przejechanych odcinków.

Każdy odcinek ma przypisaną klasę oznaczającą typ drogi, którą pojazdy kursują (np. autostrada, droga szybkiego ruchu, droga krajowa). Dla danej trasy/odcinka wyliczany jest średni czas jej pokonania, z uwzględnieniem pory oraz typu dnia (dni robocze/weekendy/święta).

Na bieżąco prowadzone są statystyki odnośnie zleceń i pojazdów:

- Ilość zleceń z podziałem na wykorzystane pojazdy, typy towarów oraz klientów,
- Rzeczywiste czasy przejazdów,
- Częstotliwość przejazdów przez wybrane trasy,
- Częstotliwość przejazdów przez wybrane odcinki dróg,
- Wykorzystanie kierowców.

Założenia projektu

Naszym celem jest stworzenie spójnej bazy danych, która będzie opisywać działalność firmy transportowej. Na bazie modelu pojęciowego stworzony zostanie model logiczny, a następnie fizyczna implementacja bazy danych. Jako system bazodanowy wykorzystanie zostanie PostgreSQL.

Na tej podstawie zrealizowana zostanie aplikacja wspierająca działalność firmy. Wstępne założenia co do funkcjonalności obejmują m.in.:

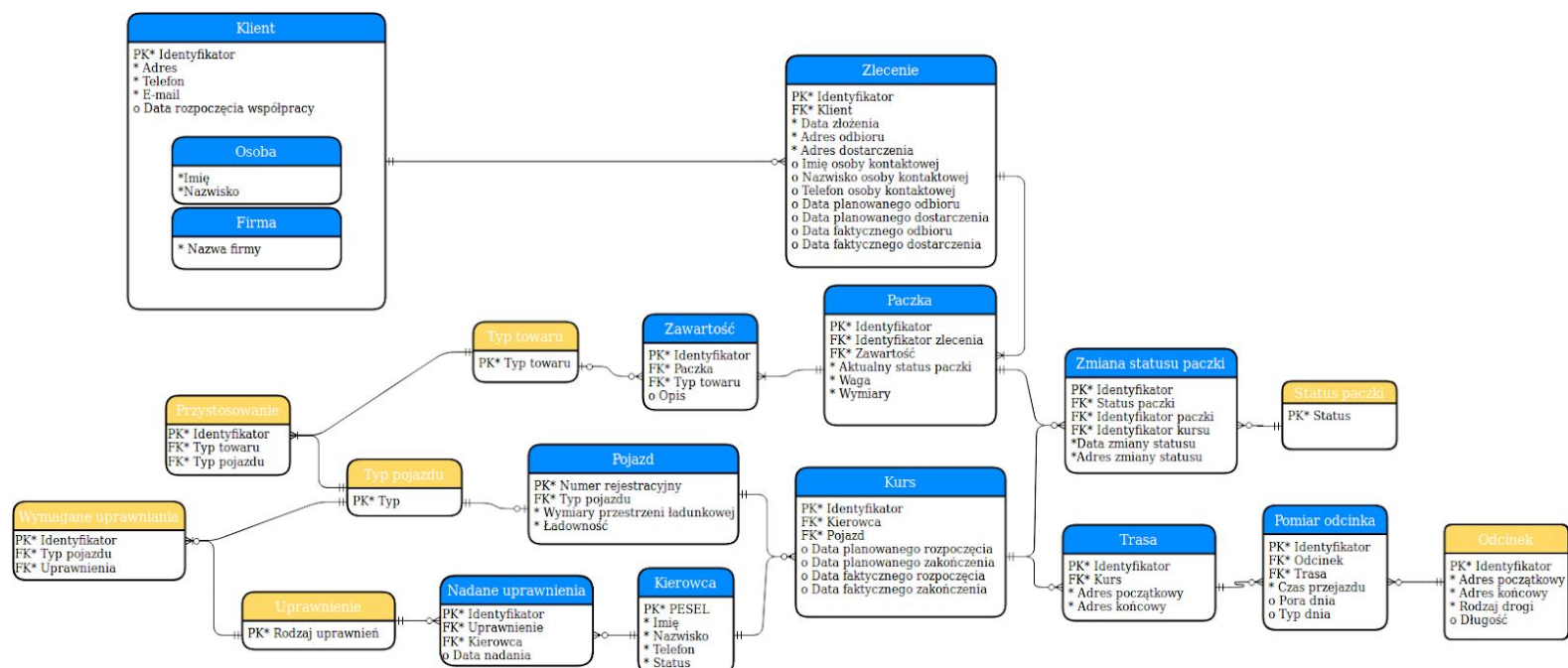
- Dodawanie i modyfikowanie zleceń, pojazdów, kierowców,
- Raportowanie kursów i zrealizowanych zleceń,
- Generowanie raportów o trasach, kierowcach, działalności firmy.

Główna część aplikacji powstanie w języku Java w środowisku Spring. Frontend zostanie stworzony przy użyciu różnych technologii HTML, CSS oraz ReactJS.

Etapy projektu

Termin	Opis
11.12.2018	W ramach tego etapu utworzony zostanie model encji, podane zostaną wymagania funkcjonalne i нефункционалне całego systemu. Zdefiniowana zostanie także struktura raportów, które generowane będą w następnych etapach projektu.
18.12.2018	W ramach tego etapu powstanie model logiczny oraz fizyczna implementacja bazy danych w PostgreSQL. Wspecyfikowane zostaną założenia związane z częstotliwością i sposobem korzystania z bazy danych. Na tej podstawie oraz na podstawie struktury wymaganego raportu przeprowadzona zostanie wstępna analiza wydajności różnych rozwiązań i na tej podstawie wybrana zostanie odpowiednia implementacja. Baza wypełniona zostanie przykładowymi danymi wygenerowanymi przez skrypty. Stworzone zostaną również odpowiednie zapytania służące do generowania raportu, którego struktura opisana zostanie w poprzednim etapie.
22.01.2019	W ramach tego etapu powstanie w pełni działająca aplikacja spełniająca wymagania uzgodnione z Prowadzącym. Model bazy danych zostanie już ostatecznie dostosowany na potrzeby aplikacji. Aplikacja uruchomiona zostanie na zewnętrznym VPS-ie.

Model ER:



Kolor żółty - encje metamodelu

Kolor niebieski - encje zwykłe

FK - klucz obcy

PK - klucz główny

o - atrybut opcjonalny

* - atrybut obligatoryjny

Opis modelu:

Encja	Atrybuty
Klient - encja reprezentująca klienta firmy, który może zlecać przewóz towaru. Klient może być osobą lub firmą.	Identyfikator - klucz główny
	Adres - adres klienta
	Telefon - telefon do klienta
	E-mail - e-mail do klienta
	Data rozpoczęcia współpracy - zazwyczaj jest to data pierwszego zlecenia otrzymanego od klienta

Osoba - podtyp encji Klient. Dziedziczy wszystkie atrybuty encji Klient.	Imię
	Nazwisko
Firma - podtyp encji Klient. Dziedziczy wszystkie atrybuty encji Klient.	Nazwa firmy
Zlecenie - encja reprezentująca zlecenie od klienta. Zlecenie składa z co najmniej jednej paczki.	Identyfikator - klucz główny
	Klient - klucz obcy wskazujący na zlecniodawcę
	Data złożenia - data (z dokładnością do minuty) złożenia przez klienta zlecenia
	Adres odbioru - adres mówiący skąd należy odebrać paczkę
	Adres dostarczenia - adres mówiący dokąd należy dostarczyć paczkę
	Imię osoby kontaktowej - jeśli paczkę wysyła firma, to musi wyznaczyć osobę do kontaktu odpowiedzialną za zlecenia
	Nazwisko osoby kontaktowej - jak wyżej
	Telefon osoby kontaktowej - jak wyżej
	Data planowanego odbioru - data mówiąca kiedy firma przewiduje odebrać paczkę. Jest to atrybut opcjonalny
	Data planowanego dostarczenia - data mówiąca kiedy firma przewiduje dostarczyć paczkę. Jest to atrybut opcjonalny
	Data faktycznego odbioru - data mówiąca kiedy firma odebrała paczkę od klienta. Atrybut jest uzupełniany po odebraniu paczki
	Data faktycznego dostarczenia - data mówiąca kiedy firma dostarczyła paczkę od klienta. Atrybut jest uzupełniany po dostarczeniu paczki
Paczka - encja reprezentująca abstrakcyjny składnik zlecenia	Identyfikator - klucz główny, powinien być losowy, ponieważ służyć będzie do sprawdzenia statusu paczki w systemie
	Identyfikator zlecenia - klucz obcy wskazujący na zlecenie, do którego należy paczka
	Zawartość - klucz obcy wskazujący na zawartość paczki

	Aktualny status paczki - status przypisany paczce wynikający z historii zmian statusu paczki
	Waga
	Wymiary
Zawartość - encja opisująca towar znajdujący się w paczce	Identyfikator - klucz główny
	Paczka - klucz obcy wskazujący na paczkę przechowującą daną zawartość
	Typ towaru - klucz obcy wskazujący na typ towaru przewożonego daną paczką
	Opis - opcjonalny atrybut zawierający dodatkowe informacje o paczce podane przez klienta
Typ towaru - encja opisująca typ towaru zawartego w paczce	Typ towaru - klucz główny będący opisem danego typu
Zmiana statusu paczki - encja reprezentująca zmianę statusu paczki. Aktualny status paczki wynika z jego historii zmian	Identyfikator - klucz główny
	Status paczki - klucz obcy wskazujący na typ statusu paczki
	Identyfikator paczki - klucz obcy wskazujący na paczkę powiązaną z daną zmianą statusu
	Identyfikator kursu - klucz obcy wskazujący na kurs, do którego należy paczka której status został zmieniony
	Data zmiany statusu
	Adres zmiany statusu
Status paczki - encja reprezentująca możliwe statusy, jakie może mieć paczka (paczka odebrana/dostarczona/w drodze)	Status - klucz główny będący opisem danego statusu
Wymagane uprawnienia - encja opisująca jakie uprawnienia są wymagane do prowadzenia danego typu pojazdu.	Identyfikator - klucz główny
	Typ pojazdu - klucz obcy wskazujący na typ pojazdu, którego dotyczą wymagane uprawnienia
	Uprawnienia - klucz obcy wskazujący na uprawnienia, jakie są wymagane do prowadzenia danego typu pojazdu
Uprawnienia - encja reprezentująca rodzaj uprawnień pozwalający prowadzić pojazdy danego typu	Rodzaj uprawnień - klucz główny będący opisem danych uprawnień

Nadane uprawnienia - encja reprezentująca uprawnienia posiadane przez danego kierowcę	Identyfikator - klucz główny
	Uprawnienie - klucz obcy wskazujący na uprawnienie
	Kierowca - klucz obcy wskazujący na kierowcę będącego posiadaczem uprawnień
	Data nadania - data nadania danego typu uprawnień
Kierowca - encja reprezentująca danego kierowcę firmy	PESEL - klucz główny
	Imię
	Nazwisko
	Telefon - numer telefonu kontaktowego do kierowcy
	Status - status danego kierowcy w firmie (aktywny zawodowo/zwolniony/emerytowany)
Pojazd - encja reprezentująca dany pojazd używany do wykonywania kursów	Numer rejestracyjny - klucz główny
	Typ pojazdu - klucz obcy wskazujący na typ pojazdu
	Wymiary przestrzeni ładunkowej - wymiary w cm określające przestrzeń ładunkową danego pojazdu
	Ładowność - największa masa ładunku, jaką może przewozić pojazd
Typ pojazdu - encja określająca typ pojazdu	Typ - klucz główny będący opisem określającym dany typ
Przystosowanie - encja określająca jakie typy towarów mogą być przewożone w danym pojeździe	Identyfikator - klucz główny
	Typ towaru - klucz obcy wskazujący na towary, jakie mogą być przewożone danym typem pojazdu
	Typ pojazdu - klucz obcy wskazujący na pojazd, którego dotyczy przystosowanie do przewożenia danych typów towarów
Kurs - encja reprezentująca ciąg akcji i przejazdów wykonanych w ramach zleceń	Identyfikator - klucz główny
	Kierowca - klucz obcy wskazujący na kierowcę wykonujący dany kurs
	Pojazd - klucz obcy wskazujący na pojazd używany w danym kursie
	Data planowanego rozpoczęcia - data określająca kiedy

	kierowca powinien rozpocząć kurs
	Data planowanego zakończenia - szacunkowa data kiedy kierowca powinien dotrzeć do końca trasy
	Data faktycznego rozpoczęcia - moment wyruszenia z punktu początkowego trasy
	Data faktycznego zakończenia - moment dotarcia do punktu końcowego trasy
Trasa - encja reprezentująca przejazd od punktu A do punktu B pomiędzy kolejnymi akcjami związanymi z paczką (odbiór, dostarczenie) lub kursem (początek, koniec)	Identyfikator - klucz główny
	Kurs - klucz obcy wskazujący, częścią jakiego kursu jest dana trasa
	Adres początkowy - miejsce które określa punkt początkowy trasy
	Adres końcowy - miejsce które określa punkt końcowy trasy
Odcinek - encja reprezentująca pojedynczy fragment drogi, na którym możemy dokonywać pomiarów	Identyfikator - klucz główny
	Adres początkowy - miejsce które określa punkt początkowy odcinka
	Adres końcowy - miejsce które określa punkt końcowy odcinka
	Rodzaj drogi - np. autostrada, droga ekspresowa
	Długość - wyrażona w kilometrach długość odcinka
Pomiar odcinka - encja opisująca przejazd pojedynczego odcinka na danej trasie	Identyfikator - klucz główny
	Odcinek - klucz obcy wskazujący na odcinek, którego dotyczy pomiar
	Trasa - klucz obcy wskazujący trasę, w której skład wchodził odcinek będący przedmiotem pomiaru
	Czas przejazdu - faktycznie zmierzony czas przejazdu danego odcinka
	Pora dnia - pora dnia, w której wykonano pomiar, np. wieczór, noc
	Typ dnia - typ dnia, w której wykonano pomiar, np. dzień zwykły, dzień świąteczny

Wymagania funkcjonalne

1. Zgłoszenie przez klienta zlecenia za pośrednictwem strony internetowej.
 - Klient powinien być zalogowany na swoje indywidualne konto.
 - Klient powinien mieć dostęp do swoich zleceń.
2. Założenie konta przez klienta.
 - Klient powinien móc założyć konto w systemie oraz wprowadzić dane.
 - Klient powinien mieć możliwość modyfikowania swoich danych.
3. Dodawanie, usuwanie i modyfikowanie informacji o kierowcach.
 - Powinno się to odbywać przez uprawnionego i zalogowanego pracownika firmy.
 - Kierowca powinien otrzymać swój unikalny login i hasło, które posłużą mu do logowania do systemu.
4. Dodawanie, usuwanie i modyfikowanie informacji o pojazdach.
 - Powinno się to odbywać przez uprawnionego i zalogowanego pracownika firmy.
5. Przydzielenie paczek ze zleceń do pojazdów i kierowców firmy.
 - System powinien automatycznie przypisać zlecenia według wcześniej przyjętego algorytmu.
 - System powinien umożliwiać ręczną modyfikację przydziału przez uprawnionego i zalogowanego pracownika.
 - Kierowca powinien mieć możliwość przeglądu swoich aktualnych zadań w formie tabelki.
6. Monitorowanie i modyfikowanie statusów paczek.
 - Modyfikacja statusu powinna się odbywać automatycznie na podstawie ostatniej znanej akcji związanej z paczką.
 - System powinien oferować możliwość wykonania przez klienta zapytania o status paczki.
7. Generowanie raportów dotyczących wykonanych kursów, zleceń, kierowców firmy oraz tras pokonywanych przez pojazdy.

- System powinien oferować możliwość modyfikowania parametrów raportu, np. przedział czasowy, którego dotyczy raport, typ pojazdu.
 - Raporty powinny być wyświetlane w formie tabel lub wykresów.
 - System powinien oferować możliwość eksportu raportu do pliku CSV.
8. Logowanie przejeżdżanych odcinków.
- Pojazdy powinny automatycznie logować przejeżdżane odcinki.
 - System powinien umożliwiać logowanie tylko tych odcinków, które są w bazie.
9. Dodawanie odcinków do monitorowania.
- Powinno się to odbywać przez uprawnionego i zalogowanego pracownika firmy.
10. System powinien automatycznie codziennie tworzyć swoje backupy.
- Tworzenie backupu powinno się odbywać w godzinach nocnych, aby nie zakłócać pracy użytkownikom systemu.

Wymagania niefunkcjonalne

1. Generowanie raportów i zestawień w czasie do 5 minut przy następujących ograniczeniach:
 - a. liczba pracowników ≤ 5000 ,
 - b. liczba wierszy w tabeli zleceń ≤ 1000000 ,
 - c. liczba wierszy w tabeli paczek ≤ 1000000 ,
 - d. liczba wierszy w tabeli pomiarów ≤ 1000000 .
2. System powinien autoryzować użytkownika. Podgląd statusu paczki nie wymaga wcześniejszego zalogowania, wystarczy znać numer paczki.
3. System powinien być odporność na podstawowe ataki takie jak XSS czy SQL injection.
4. System powinien codziennie tworzyć w godzinach nocnych swoje backupy.
5. System powinien działać jednocześnie na dwóch niezależnych serwerach lustrzanych. Awaria jednego nie powinna powodować przerwy w działaniu systemu.
6. Jednorazowo system może być niedostępny nie dłużej niż przez 2 godziny.
7. W ciągu miesiąca system może być niedostępny maksymalnie przez 24 godziny.

Struktura raportu o kierowcach

Raport o kierowcach tworzy zestawienie kierowców o konkretnych uprawnieniach.

Dane wejściowe raportu:

- okres, z jakiego ma on zebrać dane
- typ uprawnień kierowcy
- imię i nazwisko kierowcy
- używany pojazd

Dane wyjściowe raportu:

Raport o kierowcach z okresu 01.01.2019 r. - 01.02.2019 r.

Imię i nazwisko	Jan Kowalski	Adam Nowak
Uprawnienia	Pojazd z przyczepą do 12 ton Przewóz materiałów niebezpiecznych	Pojazd z przyczepą do 12 ton
Używane pojazdy	Pojazd A - 80% czasu	Pojazd A - 10% Czasu
	Pojazd B - 20% czasu	Pojazd C - 90% Czasu
Wykonane kursy	150	10
Paczki dostarczone terminowo	5000	10
Paczki dostarczone nieterminowo	200	3
Sumaryczna waga dostarczonych paczek	7500 kg	15000 kg
Suma przepracowanych godzin	120h	60h
Suma przejechanych kilometrów	7440 km	4380 km
Średnia prędkość jazdy	62 km/h	73 km/h

Pracownik o najwyższej liczbie przewiezionych paczek: Jan Kowalski

Pracownik o najniższej liczbie przewiezionych paczek: Adam Nowak

Pracownik o najwyższej liczbie wykonanych kursów: Jan Kowalski

Pracownik o najniższej liczbie wykonanych kursów: Adam Nowak

Pracownik o najwyższej liczbie przejechanych kilometrów: Jan Kowalski

Pracownik o najniższej liczbie przejechanych kilometrów: Adam Nowak

Struktura raportu o odcinkach

Raport o odcinkach prezentuje zestawienie odcinków pokonywanych

Dane wejściowe raportu:

- okres, z jakiego ma pobrać dane
- typ pojazdu

Dane wyjściowe raportu:

Raport o odcinkach z okresu 01.01.2019 r. - 01.02.2019 r.

Odcinek	Odcinek A	Odcinek B
Długość odcinka	10 km	7 km
Adres początkowy	ul. Prosta 12, Radom	ul. Kwiatowa 1, Warszawa
Adres końcowy	ul. Krzywa 32, Radom	ul. Kolorowa 9, Warszawa
Liczba wystąpień odcinka w wykonywanych kursach	30	20
Średnia prędkość przejazdu (wraz z odchyleniem standardowym) dla danego typu pojazdu	45 km/h +/- 2 km/h	30 km/h +/- 3 km/h
Średni czas przejazdu (wraz z odchyleniem standardowym) dla danego typu pojazdu	13 min +/- 1 min	14 min +/- 1 min

Najczęściej pokonywane odcinki	Ilość	Najrzadziej pokonywane odcinki	Ilość
Odcinek A	30	Odcinek Z	1

Odcinek B	20	Odcinek Y	2
Odcinek C	10	Odcinek X	3

Odcinek z największą średnią prędkością	Średnia prędkość	Odcinek z najmniejszą średnią prędkością	Średnia prędkość
Odcinek D	72 km/h	Odcinek X	12 km/h
Odcinek A	66 km/h	Odcinek Z	22 km/h

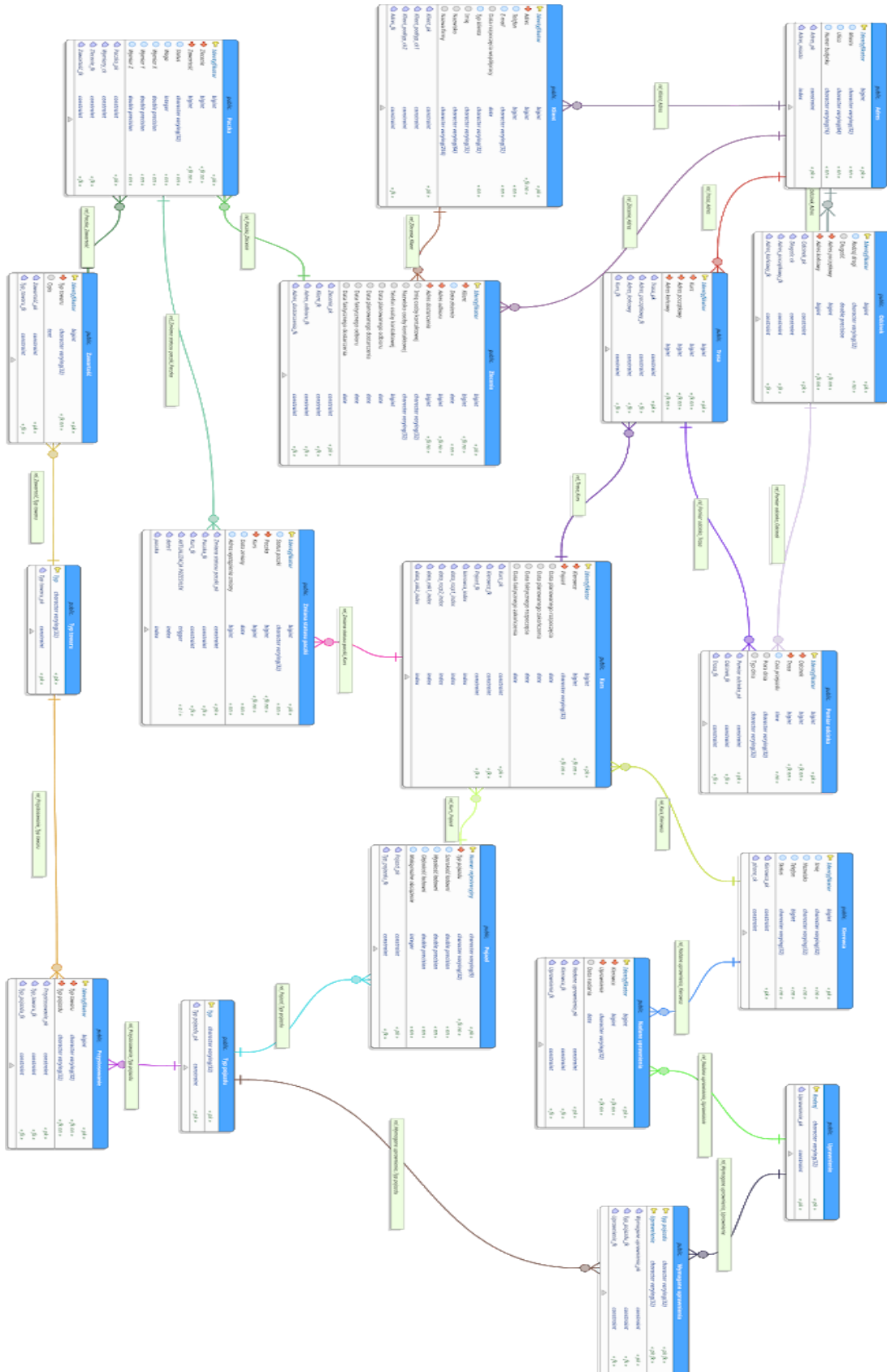
Model logiczny i implementacja:

Na podstawie modelu pojęciowego w programie pgModeler stworzony został model relacyjny. Ze względu na przyjętą wcześniej konwencję i oznaczenia w wielu przypadkach możliwa była transformacja encji w tabele w zasadzie bez większych zmian. Model relacyjny zawierał jeden wzorzec podtypu. Zdecydowaliśmy się na realizację wszystkich podtypów przy użyciu jednej tabeli. Typ rozpoznawany jest za pomocą odpowiedniego pola. Takie podejście ma tę zaletę, że odwołując się do zbioru Klientów nie trzeba wykonywać kosztownej operacji złączenia. Minusem jest zwiększona zajętość pamięci spowodowana sporą liczbą pól opcjonalnych oraz fakt, że należy dokonywać przy operacjach wstawiania i aktualizacji dodatkowych operacji sprawdzających zapewniających spójność danych.

Na tym etapie przyjęliśmy zasadę tworzenia restrykcyjnego modelu. Być może przyjęte przez nas ograniczenia w następnym etapie projektu zostaną złagodzone. Po stronie bazy danych zaimplementowane zostały następujące mechanizmy:

- sprawdzanie wartości NULL,
- zdefiniowanie zachowania podczas usuwania wierszy (CASCADE i RESTRICT),
- trigger przy dodawaniu zmiany statusu paczki odpowiadający za zmianę wartości pola "Status" w tabeli "Paczka",

- Poniższy obrazek prezentuje uzyskany model relacyjny.



Administracja

Jako system baz danych został przez nas wybrany PostgreSQL. Zdecydowaliśmy się nie na lokalne instancje, a na jedną wspólną na zewnętrznym serwerze. Na chwilę obecną nasza baza uruchomiona jest na osobnym VPS-ie (Aruba Cloud) z systemem operacyjnym Ubuntu 18.04 LTS Server. Zainstalowany PostgreSQL jest w wersji 10. Ze względów bezpieczeństwa dostęp do bazy wymaga tunelowania ssh. Administracja bazą odbywa się przy użyciu aplikacji pgAdmin4, która to zainstalowana została na naszym serwerze i oferuje webowy interfejs oraz dostęp zdalny (również wymagane jest tunelowanie). Takie podejście ułatwia wspólną pracę nad jedną bazą, jednak wymaga dodatkowych nakładów związanych z zarządzaniem środowiskiem.

Dane

Baza została wypełniona przykładowymi danymi za pomocą skryptu napisanego w języku Python. Skrypt generuje dane do każdej z tabel zgodnie z typami danych w nich zawartych, a następnie zapisuje je do pliku w formie instrukcji INSERT. Wprowadzenie danych do bazy jest wykonywane przez wprowadzenie wygenerowanych instrukcji INSERT do skryptu w aplikacji pgAdmin4.

Analiza założeń korzystania z bazy

W ramach korzystania z bazy danych i aplikacji, często wykonywane będą operacje wstawianie wierszy do tabel Zlecenie, Adres, Kurs, Paczka, Zawartość, Zmiana statusu paczki. Zbiór kierowców lub flota pojazdów nie zmieniają się tak często, zatem tabele Kierowca czy Pojazd są tabelami, do których stosunkowo rzadko dodawane będą nowe rekordy. Najrzadziej zmieniane będą tabele odpowiadające encjom metamodelu.

Dla klientów firmy jedną z podstawowych operacji będzie sprawdzenie statusu paczki. Sprawdzenie to wymagać będzie wykonania zapytania typu SELECT na tabeli Paczka. Porównywany będzie identyfikator paczki, który jest kluczem głównym, zatem dodatkowe indeksy nie powinny być potrzebne.

W ramach realizacji zleceń firma często będzie tworzyć nowe wiersze w tabeli Zmiana statusu paczki i na tej podstawie zmieniać status paczki w tabeli Paczka. Firma również będzie aktualizować daty w tabeli Kurs i Zlecenie. Zlecenie można uznać za zrealizowane dopiero wtedy, kiedy wszystkie paczki z danego zlecenia zostaną dostarczone, zatem w momencie, kiedy paczka jest dostarczona, należy sprawdzić zapytaniem typu EXISTS, czy jest jeszcze paczka z danego zlecenia, która nie została dostarczona. Sugeruje to potrzebę wprowadzenia indeksu na polu Zlecenie w tabeli Paczka.

Aplikacja będzie generować również dla firmy raporty o kierowcach oraz raporty o odcinkach. Raporty te będą dotyczyły dużej części tabel w bazie danych. Warto zauważyć, że podstawowym kryterium obu raportów będzie przedział czasowy, który zazwyczaj będzie bardzo niewielki w porównaniu z okresem działalności firmy. Wprowadzając indeksy na polu dat w tabelach Kurs i Zlecenie możemy bez przeglądania całości tych tabel znacząco zawęzić zakres przeszukiwanych danych.

Raporty, zapytania i indeksy

Skupiliśmy się na raportach dotyczących kierowców. Poniższy skrypt generuje liczbę przewożonych paczek dla każdego z kierowców w zadanym okresie:

```
SELECT "Kierowca"."Identyfikator", COUNT(*) AS "Liczba paczek"
```

```
FROM
```

```
(
```

```
SELECT *
```

```
FROM "Paczka", "Zmiana statusu paczki"
```

```
WHERE "Paczka"."Identyfikator" = "Zmiana statusu paczki"."Paczka"
```

```
AND "Zmiana statusu paczki"."Status paczki" = 'Dostarczona'
```

```
AND "Zmiana statusu paczki"."Data zmiany" > '2001-09-28'
```

```
AND "Zmiana statusu paczki"."Data zmiany" < '2001-10-28'
```


) AS tmp, "Kurs", "Kierowca"

*WHERE tmp."Kurs" = "Kurs"."Identyfikator" AND "Kierowca"."Identyfikator" =
"Kurs"."Kierowca"*

GROUP BY "Kierowca"."Identyfikator";

Podzapytanie dokonuje złączenia tabel “Paczka” i “Zmiana statusu paczki”, co sugeruje zasadność utworzenia indeksu na atrybucie "Zmiana statusu paczki"."Paczka". Ponadto warunki na "Zmiana statusu paczki"."Data zmiany" są mocno selektywne, również tutaj warto rozważyć dodanie indeksu.

Poniższe zapytanie sumuje liczbę przepracowanych (w ramach kursów) godzin przez poszczególnych kierowców.

*SELECT "Kierowca"."Identyfikator", date_part('hour', "Kurs"."Data faktycznego
zakończenia")::timestamp - "Kurs"."Data faktycznego rozpoczęcia")::timestamp)*

AS "Suma przepracowanych godzin"

FROM "Kurs", "Kierowca"

WHERE "Kurs"."Kierowca" = "Kierowca"."Identyfikator"

*GROUP BY "Kierowca"."Identyfikator", "Kurs"."Data faktycznego zakończenia",
"Kurs"."Data faktycznego rozpoczęcia";*

W ramach tego zapytania wykonywane jest złączenie na tabelach “Kurs” oraz “Kierowca”. Sprawdziliśmy różnicę w wydajności po dodaniu indeksu na kluczu obcym:

*EXPLAIN ANALYZE SELECT **

FROM "Kurs", "Kierowca"

WHERE "Kurs"."Kierowca" = "Kierowca"."Identyfikator";

Wynik:

"Hash Join (cost=16.30..33.52 rows=570 width=376) (actual time=0.048..0.067 rows=50 loops=1)"

" Hash Cond: ("Kurs"."Kierowca" = "Kierowca"."Identyfikator")"

" -> Seq Scan on "Kurs" (cost=0.00..15.70 rows=570 width=114) (actual time=0.009..0.013 rows=50 loops=1)"

" -> Hash (cost=12.80..12.80 rows=280 width=262) (actual time=0.025..0.025 rows=50 loops=1)"

" Buckets: 1024 Batches: 1 Memory Usage: 12kB"

" -> Seq Scan on "Kierowca" (cost=0.00..12.80 rows=280 width=262) (actual time=0.005..0.011 rows=50 loops=1)"

"Planning time: 0.125 ms"

"Execution time: 0.099 ms"

Po dodaniu indeksu na kluczu obcym uzyskaliśmy:

"Hash Join (cost=2.12..16.48 rows=50 width=376) (actual time=0.034..0.055 rows=50 loops=1)"

" Hash Cond: ("Kierowca"."Identyfikator" = "Kurs"."Kierowca")"

" -> Seq Scan on "Kierowca" (cost=0.00..12.80 rows=280 width=262) (actual time=0.009..0.012 rows=50 loops=1)"

" -> Hash (cost=1.50..1.50 rows=50 width=114) (actual time=0.017..0.017 rows=50 loops=1)"

" Buckets: 1024 Batches: 1 Memory Usage: 12kB"

" -> Seq Scan on "Kurs" (cost=0.00..1.50 rows=50 width=114) (actual time=0.004..0.009 rows=50 loops=1)"

"Planning time: 0.124 ms"

"Execution time: 0.080 ms"

Wzrost wydajności jest pomijalny.

Zauważmy, że w przypadku kursu interesuje nas konkretny zakres dat. Przykładowe zapytanie:

*EXPLAIN ANALYZE SELECT **

FROM "Kurs"

WHERE "Kurs"."Data planowanego rozpoczęcia" > '2018-08-28'

AND "Kurs"."Data planowanego rozpoczęcia" < '2018-08-29';

Wynik:

"Seq Scan on "Kurs" (cost=0.00..1.75 rows=1 width=114) (actual time=0.017..0.017 rows=0 loops=1)"

" Filter: (("Data planowanego rozpoczęcia" > '2018-08-28'::date) AND ("Data planowanego rozpoczęcia" < '2018-08-28'::date))"

" Rows Removed by Filter: 50"

"Planning time: 0.091 ms"

"Execution time: 0.044 ms"

Dodajmy indeks na atrybut "Data planowanego rozpoczęcia". Teraz baza może korzystać z indeksu. Danych na chwilę obecną jest niewiele, dlatego skanowanie sekwencyjne wierszy jest wystarczająco szybkie i ma mniejszy narzut, jednak dla dużych danych często chcemy uniknąć skanowania sekwencyjnego.

"Index Scan using data_rozp1_index on "Kurs" (cost=0.14..8.16 rows=1 width=114) (actual time=0.024..0.024 rows=0 loops=1)"

" Index Cond: (("Data planowanego rozpoczęcia" >= '2017-08-28'::date) AND ("Data planowanego rozpoczęcia" <= '2017-10-28'::date))"

"Planning time: 0.103 ms"

"Execution time: 0.043 ms"

Poniższe zapytanie wykorzystywane w raporcie o odcinkach:

SELECT o."Identyfikator", COUNT() as "Liczba wystąpień", o."Długość", AVG(p."Czas przejazdu") as "Średni czas przejazdu", a."Miasto", a."Ulica"*

FROM "Odcinek" o, "Pomiar odcinka" p, "Adres" a

WHERE o."Identyfikator" = p."Odcinek" AND a."Identyfikator" = o."Adres początkowy"

GROUP BY o."Identyfikator", a."Miasto", a."Ulica"

Funkcjonalności, które zamierzamy zaimplementować w aplikacji:

W ramach aplikacji zaimplementowane zostaną następujące funkcjonalności:

1. Zgłoszenie przez klienta zlecenia.
 - Klient powinien być zalogowany na swoje indywidualne konto.
 - Klient może podejrzeć swoje zlecenia.
 - Firma może przyjąć zlecenie lub je odrzucić.
2. Przydzielenie paczek ze zleceń do pojazdów i kierowców firmy.
 - Uprawniony użytkownik powinien mieć możliwość tworzenia kursów.
 - System powinien umożliwiać przypisanie paczki do kursu przez uprawnionego użytkownika.
 - System powinien umożliwiać zmianę statusu paczki przez kierowców i innych uprawnionych pracowników.
 - System powinien umożliwiać podgląd zleceń i paczek oraz który kierowca i pojazd został do nich przyporządkowany.

- System powinien generować ostrzeżenie w przypadku przypisania paczki do zakończonego kursu lub przypisania paczki do kursu realizowanego przez zbyt mały pojazd.
- 3. Monitorowanie i modyfikowanie statusów paczek.
 - Modyfikacja statusu powinna się odbywać automatycznie na podstawie ostatniej znanej akcji związanej z paczką.
 - System powinien oferować możliwość wykonania zapytania o status paczki na podstawie jej identyfikatora.
 - Podgląd statusu paczki nie powinien wymagać logowania - wystarczy znajomość numeru paczki.
- 4. Generowanie raportów dziennych i analitycznych.
 - Raporty powinny być generowane przez uprawnionego użytkownika.
 - Raport dzienny dotyczył będzie [aktywności kierowców](#) z okresu poprzedniego dnia.
 - Raport analityczny dotyczył będzie [aktywności kierowców](#) z zadanego okresu oraz [przejechanych odcinków](#).
- 5. Weryfikacja uprawnień użytkowników.
 - System powinien uniemożliwić próbę wykonania akcji przez nieuprawnionego użytkownika.
 - System powinien rozróżniać różne poziomy uprawnień: użytkownik niezalogowany, klient, kierowca, zarządca paczek (do aktualizacji statusu paczek), analityk (do generowania raportów), administrator.
- 6. System powinien być odporny na podstawowe ataki takie jak XSS czy SQL injection.
- 7. Możliwość usuwania i wstawiania do bazy pozostałych danych na potrzeby symulacji działania systemu, w szczególności logowanie przejechanych odcinków (na potrzeby raportu o odcinkach).

Dodatkowo zostaną przedstawione efekty automatycznych działań takich jak przypisanie zlecenia do pojazdów i kierowców lub modyfikacja statusu paczki w celu sprawdzenia poprawności działania systemu.

Aplikacja:

Aplikacja powstała w języku Java (wersja 10) przy użyciu frameworka Spring. Skorzystaliśmy z komponentów Spring Boot-a. W celu komunikacji z bazą danych wykorzystaliśmy środowisko ORM bazujące na JPA - Hibernate. Aplikacja stworzona została jako webowa, zaś komunikacja odbywa się za pomocą REST API. Za graficzny interfejs odpowiadają technologie HTML, CSS (wykorzystaliśmy ogólnodostępny szablon), oraz język javascript (jQuery). Zrezygnowaliśmy z zastosowania ReactJS-a, gdyż byłaby to kolejna złożona technologia, a z racji tego, że dopiero uczymy się wcześniej wymienionych, za bardzo utrudniłoby projekt.

W ramach aplikacji zaimplementowane zostały następujące funkcjonalności:

1. Rejestracja zlecenia w systemie.
 - Możliwość podejrzenia zarejestrowanych zleceń z dowolnego zadanego okresu.
 - Tworzenie nowych zleceń dla paczek.
2. Monitorowanie i modyfikacja statusu paczki.
 - Podgląd danych i statusu paczki poprzez przypisany jej numer.
3. Tworzenie kursu wraz z przypisaniem do niego odpowiedniego kierowcy oraz pojazdu.
4. Generowanie raportu o aktywności kierowców z dowolnego zadanego okresu.

Aplikacja oferuje graficzny interfejs użytkownika. Działa na serwerze Tomcat na zewnętrznym serwerze na porcie 8080.

Niestety nie wszystkie wcześniejsze założenia udało się zrealizować. Przede wszystkim projekt okazał się być bardziej złożony niż przewidywaliśmy - zrezygnowaliśmy z implementacji autoryzacji użytkownika - nasza aplikacja pierwotnie miała wyróżniać wiele różnych ról i praw dostępu, jednak uznaliśmy, że z punktu widzenia przedmiotu nie jest to najistotniejsza kwestia.