



Politechnika Koszalińska
Wydział Elektroniki i Informatyki
Informatyka, Semestr III, Studia stacjonarne
Rok akademicki 2019/2020

Dokumentacja z przedmiotu:
„Bazy danych NoSQL”

prowadzący:
Tomasz Juchniewicz

wykonawcy:
Łukasz Gogacz
Jakub Ledzion
Adrian Lepa

Spis treści

| | |
|--|-----------|
| 1. Wstęp | 3 |
| 1.1 Opis projektu | 3 |
| 1.2 Cel projektu | 3 |
| 2. Analiza problemu | 3 |
| 2.1 Identyfikacja aktorów | 3 |
| 2.1.1 Diagramy przypadków użycia | 3 |
| 2.2 Wymagania funkcjonalne | 5 |
| 2.1.2 Zakładanie kont użytkowników | 5 |
| 2.1.3 Definiowanie pojazdów | 5 |
| 2.1.4 Przypisywanie kierowców do pojazdów | 6 |
| 2.1.5 Raportowanie aktualnej pozycji i prędkości | 6 |
| 2.1.6 Detekcja przekroczeń prędkości | 6 |
| 2.1.7 Przeglądanie danych i wyszukiwanie aktualnej i historycznej pozycji pojazdów | 6 |
| 2.1.8 Interfejs Web API dla przeglądarek internetowych i aplikacji mobilnych | 7 |
| 2.2 Wymagania niefunkcjonalne | 7 |
| 2.2.1 Niezawodność i bezpieczeństwo | 7 |
| 2.2.2 Wspieralność | 7 |
| 2.2.3 Użyteczność i ergonomia | 7 |
| 2.2.4 Wydajność i skalowalność | 7 |
| 2.2.5 Struktura podziału pracy i kosztów | 8 |
| 3. Diagram wdrożenia | 9 |
| 4. Diagramy sekwencji | 10 |
| 4.1 Raportowanie pozycji i prędkości | 10 |
| 4.2 Zakładanie kont użytkowników | 10 |
| 5. Diagramy aktywności | 11 |
| 5.1 Zakładanie kont użytkowników | 11 |
| 5.2 Raportowanie pozycji pojazdu | 12 |
| 6. Diagram komponentów | 13 |
| 7. Diagram klas | 14 |
| 8. Baza Danych | 15 |

1. Wstęp

1.1 Opis projektu

Istnieje wiele serwisów dotyczących raportów GPS, w odróżnieniu od innych serwis „PKKierowca” będzie obsługiwał wyłącznie samochody pracowników Politechniki Koszalińskiej w celu prowadzenia różnych badań na temat stylu prowadzenia samochodu przez ludzi.

1.2 Cel projektu

Celem projektu jest stworzenie serwisu „PKKierowca” służącego do rejestrowania aktualnej pozycji i prędkości pojazdów we flocie firmy Politechniki Koszalińskiej.

2. Analiza problemu

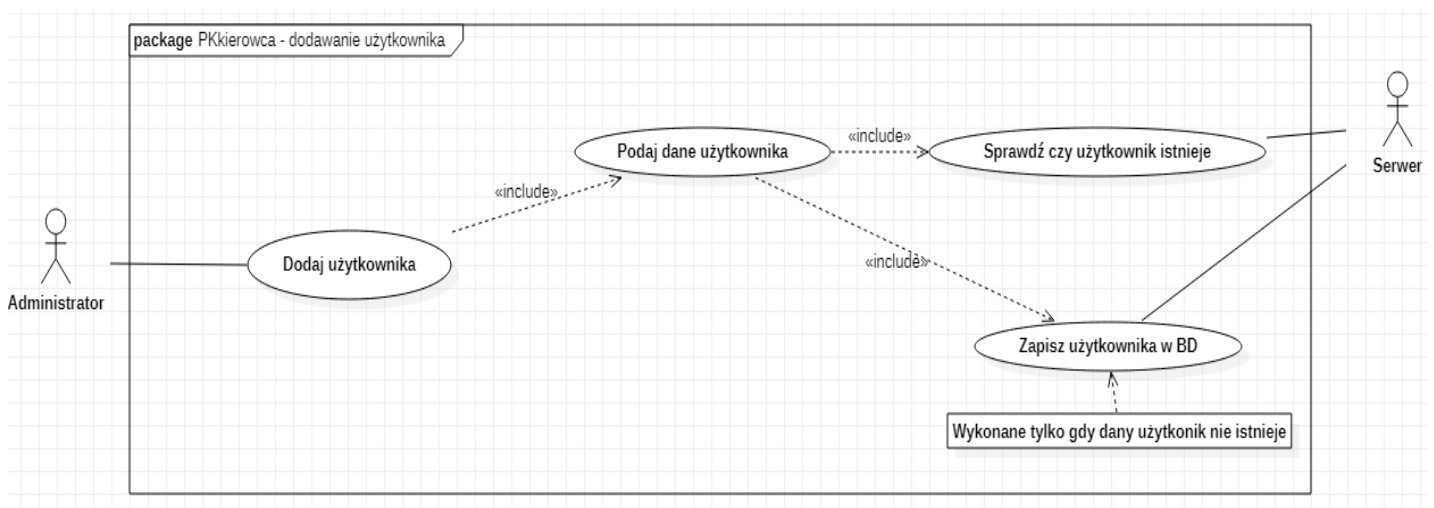
2.1 Identyfikacja aktorów

Serwis „PKkierowca” skupiać się będzie na czterech grupach użytkowników :

- administrator
- moderator
- kierowca
- samochód

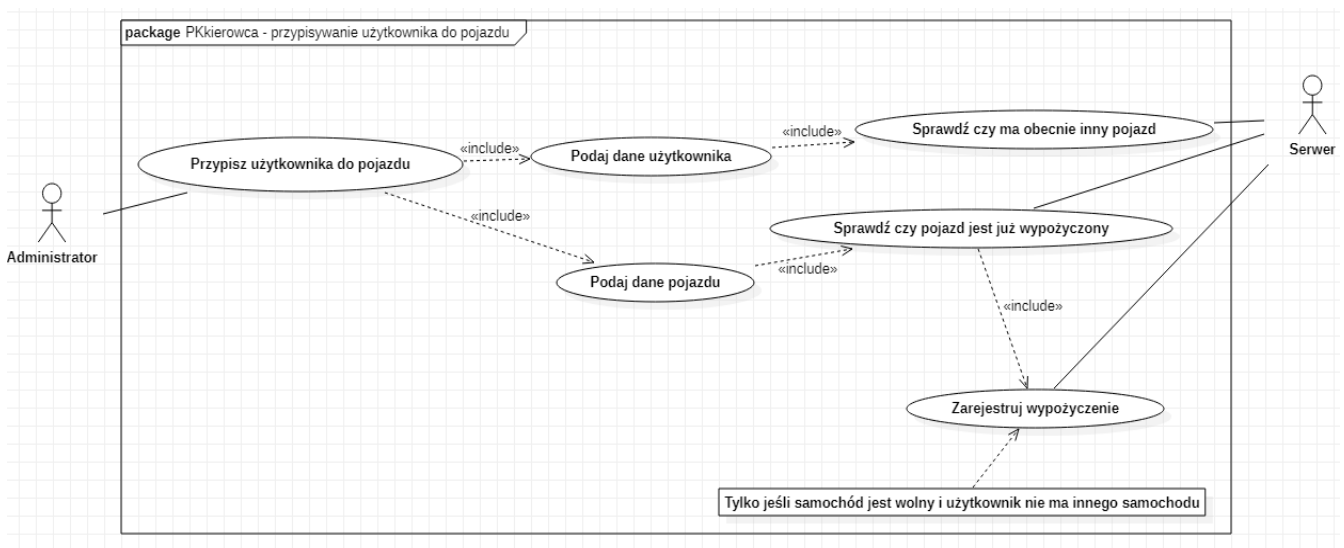
2.1.1 Diagramy przypadków użycia

- Dodawanie nowego użytkownika



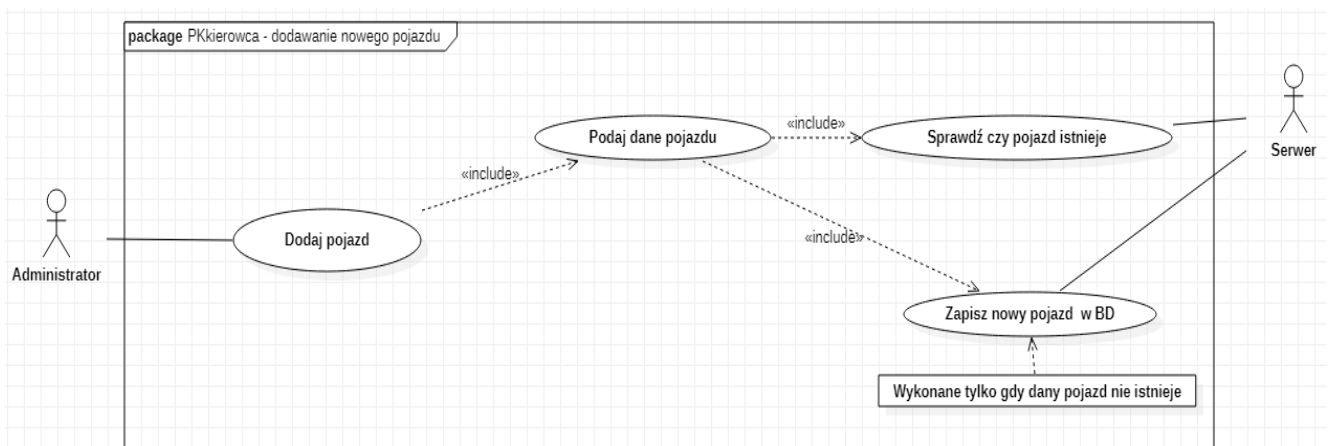
Rysunek 1. DPU - Dodanie nowego użytkownika

- Przypisywanie użytkownika do pojazdu



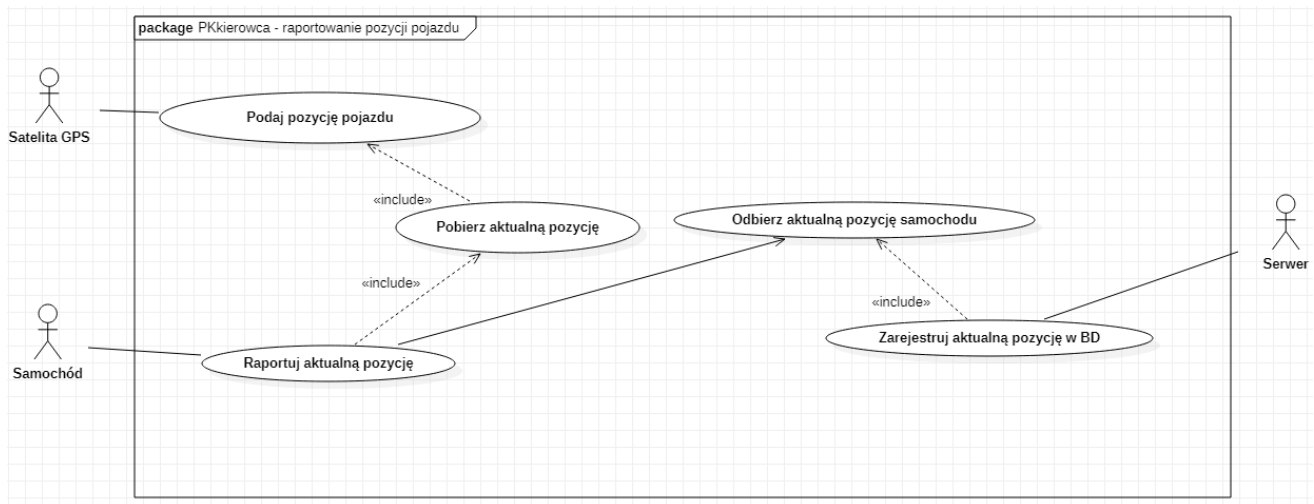
Rysunek 2. DPU - Przypisanie użytkownika do pojazdu

- Dodawanie nowego pojazdu



Rysunek 3. DPU - Dodanie nowego pojazdu

- Raportowanie pozycji pojazdu



Rysunek 4. DPU - Raportowanie pozycji pojazdu

2.2 Wymagania funkcjonalne

2.1.2 Zakładanie kont użytkowników

Każdy pracownik, będący zatrudniony w firmie i posiadający uprawnienia do korzystania z samochodu służbowego będzie musiał posiadać swoje własne indywidualne konto w serwisie.

Na dane logowania do konta będą składać się :

- login
- hasło

2.1.3 Definiowanie pojazdów

Każdy samochód wprowadzony do bazy danych będący na stanie firmy będzie zawierał informacje o :

- nr VIN pojazdu
- nr rejestracyjnego pojazdu
- nazwie marki
- modelu marki

Natomiast definiowany w serwisie będzie za pomocą swojego **numeru rejestracyjnego**.

2.1.4 Przypisywanie kierowców do pojazdów

Każdy pracownik posiadający uprawnienia do korzystania z samochodu służbowego będzie miał za pomocą numeru rejestracyjnego na stałe przypisany pojazd do swojego indywidualnego konta w serwisie.

Dodatkowo wraz z rozwojem aplikacji w przyszłości będzie dodana funkcja sprawdzania historii przypisań pojazdów co znaczy że będzie prowadzony spis informacyjny z historią który użytkownik korzystał z danego pojazdu.

2.1.5 Raportowanie aktualnej pozycji i prędkości

Pracownik po zalogowaniu się do systemu, przemieszczając się samochodem służbowym po mieście automatycznie przesyła dane co do aktualnej pozycji pojazdu i prędkości poruszania się za pośrednictwem aplikacji do serwisu.

Dane raportu :

Lokalizacja :

- współrzędne X i Y – wskazanie punktu na mapie
- data – pole uzupełniane automatycznie

Prędkość :

- podawana przez pojazd do serwisu

2.1.6 Detekcja przekroczeń prędkości

Każde przekroczenie dopuszczalnej prędkości przez pojazd będzie widoczne w serwisie poprzez generowane raporty przekroczeń prędkości w których będzie zawarte :

- użytkownik kierujący pojazdem
- datę zdarzenia
- czas zdarzenia
- miejsce zdarzenia
- prędkość pojazdu w momencie przekroczenia podaną w km/h

2.1.7 Przeglądanie danych i wyszukiwanie aktualnej i historycznej pozycji pojazdów

Administrator serwisu nadzorując pracę ma możliwość przeglądania danych na temat aktualnej pozycji wybranego pojazdu, w których znajduje się między innymi to który z pracowników identyfikowany jako kierowca, przemieszcza się wybranym pojazdem, to w jakiej lokalizacji na mapie miasta się znajduje w aktualnym czasie.

Za pomocą generowanych raportów administrator ma możliwość przejrzenia historii pozycji pojazdów w której zawarte będą informacje na temat tego który pracownik

przemieszczał się wybranym pojazdem, historii trasy przebytej przez niego w określonych ramach czasowych, oraz informacje na temat przekroczenia prędkości jeśli takie zdarzenie miało miejsce.

2.1.8 Interfejs Web API dla przeglądarek internetowych i aplikacji mobilnych

Serwis „PKkierowca” dzięki zastosowaniu restowego api będzie dostępny na różne platformy dla administratora serwisu Politechniki Koszalińskiej, dzięki czemu będzie on miał zawsze możliwość skorzystać z systemu nawet w przeglądarce na dowolnym smartfonie. Użytkownik jednak będzie musiał mieć połączenie z Internetem.

2.2 Wymagania нефunkcjonalne

2.2.1 Niezawodność i bezpieczeństwo

Treści zawarte w serwisie „PKkierowca” są treściami poufnymi przez co aplikacja będzie posiadała panel logowania. Treści zawarte w systemie będą dostępne dla osób posiadających konto a więc tylko dla wyznaczonych pracowników Politechniki Koszalińskiej. Serwis zostanie wykonany w wersji webowej, dzięki czemu korzystanie z niego będzie możliwe również na urządzeniu mobilnym takim jak tablet czy smartfon, oczywiście przy konieczności posiadania łączności z siecią internetową.

2.2.2 Wspieralność

Serwis będzie można uruchomić w każdej przeglądarce również na tych dla urządzeń mobilnych. Do poprawnego korzystania z aplikacji wymagane będzie połączenie z Internetem.

2.2.3 Użyteczność i ergonomia

Serwis będzie dostępny dla pracownika Politechniki Koszalińskiej który posiadać będzie w nim konto. Webowy interfejs użytkownika będzie prosty oraz intuicyjny tak aby każdy użytkownik posiadający mniejsze czy większe umiejętności potrafił go obsłużyć. Nie będą pobierane opłaty za korzystanie z aplikacji gdyż dostęp do niej będą posiadać wyłącznie pracownicy Politechniki Koszalińskiej dla której ten serwis jest dedykowany.

2.2.4 Wydajność i skalowalność

Do realizacji projektu serwisu zostanie zastosowany nierelacyjny system zarządzania bazą danych (NoSQL) MongoDB. System ten Charakteryzuje się dużą skalowalnością, wydajnością oraz brakiem ściśle zdefiniowanej struktury obsługiwanych baz danych co pomoże w uzyskaniu poziomego skalowania bazy danych przekładającego się na zwiększenie ilości zasobów poprzez dodawanie kolejnych instancji bazy danych.

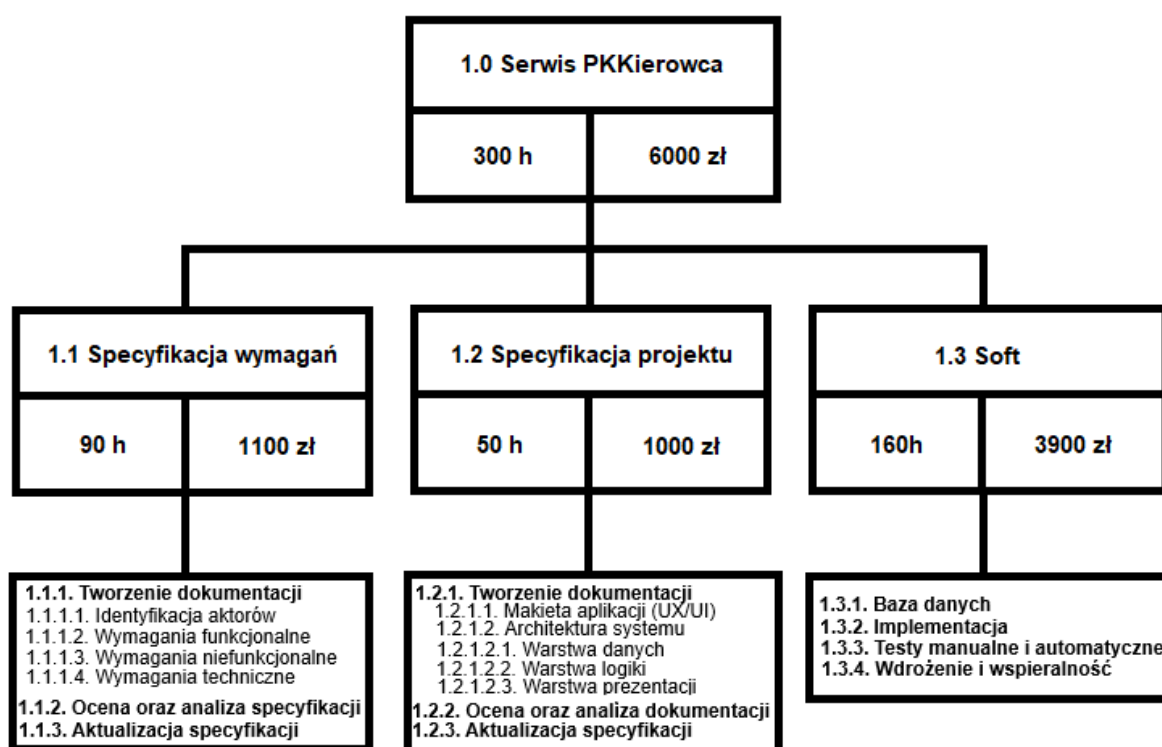
System maksymalnie będzie mógł jednocześnie obsłużyć 50 aktywnych samochodów. Aktualne pozycje aut będą pobierana co 5 sekund aby odwzorować rzeczywista pozycje pojazdu. Zabezpieczenia systemu gwarantują bezpieczeństwo dla max 50 aktywnych pracowników w tym samym czasie.

2.2.5 Struktura podziału pracy i kosztów

Na rysunku poniżej została przedstawiona struktura projektu Work Breakdown Structure serwisu PKKierowca. Jest to struktura prac potrzebnych do realizacji zadań w projekcie. Do każdego etapu zadania został przyporządkowany szacowany koszt oraz czas realizacji.

Zespół projektowy pracujący nad składa się z trzech członków:

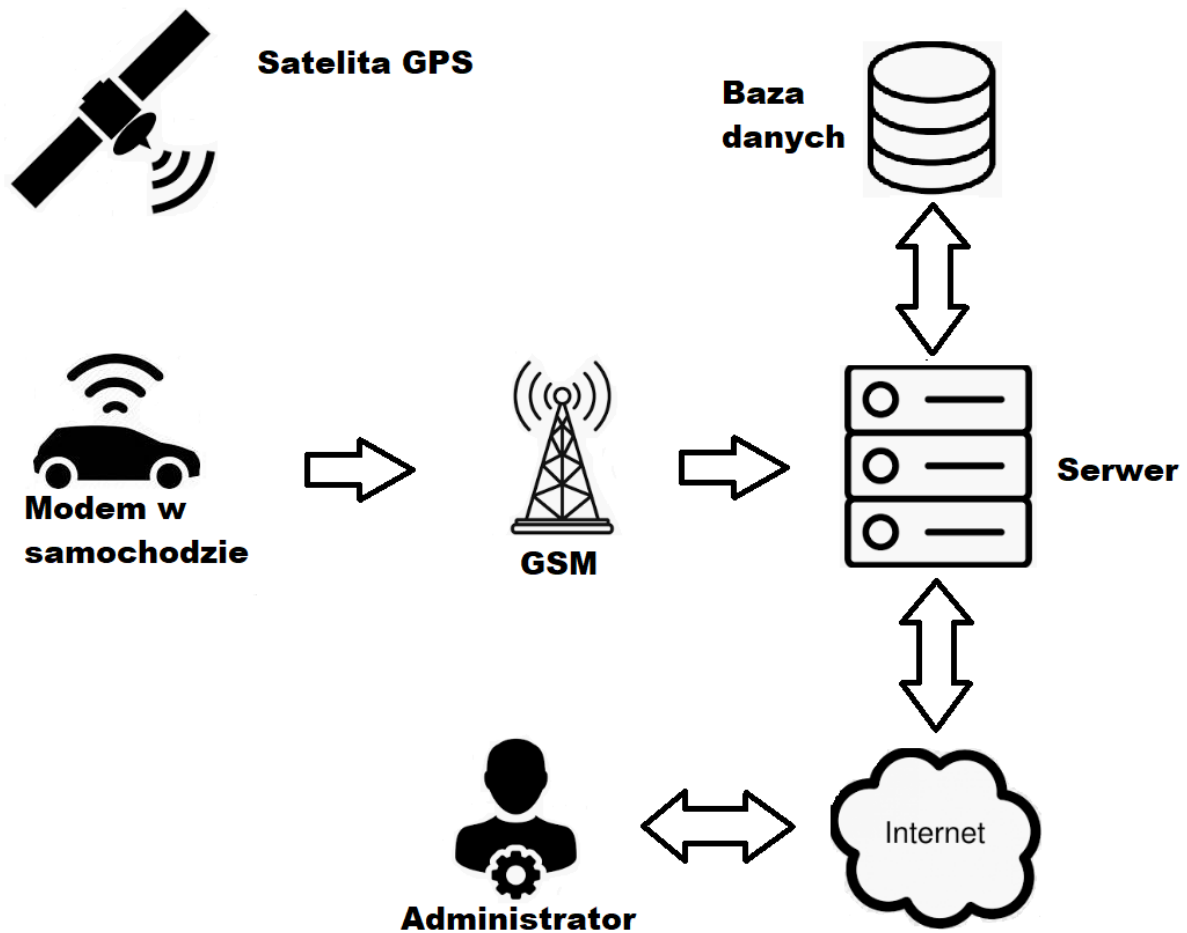
- Łukasz Gogacz – kierownik projektu, pełniący jednocześnie prace nad dokumentacją opisową projektu
- Adrian Lepa - programista zespołu pracujący nad warstwą aplikacji projektu
- Jakub Ledzion – programista bazy danych, Scrum Master.



Rysunek 5. Diagram Work Breakdown Structur

3. Diagram wdrożenia

Na rysunku szóstym został przedstawiony diagram wdrożenia. Przedstawia on fizyczne elementy wchodzące w skład systemu „PKkierowca” i połączenia między nimi.

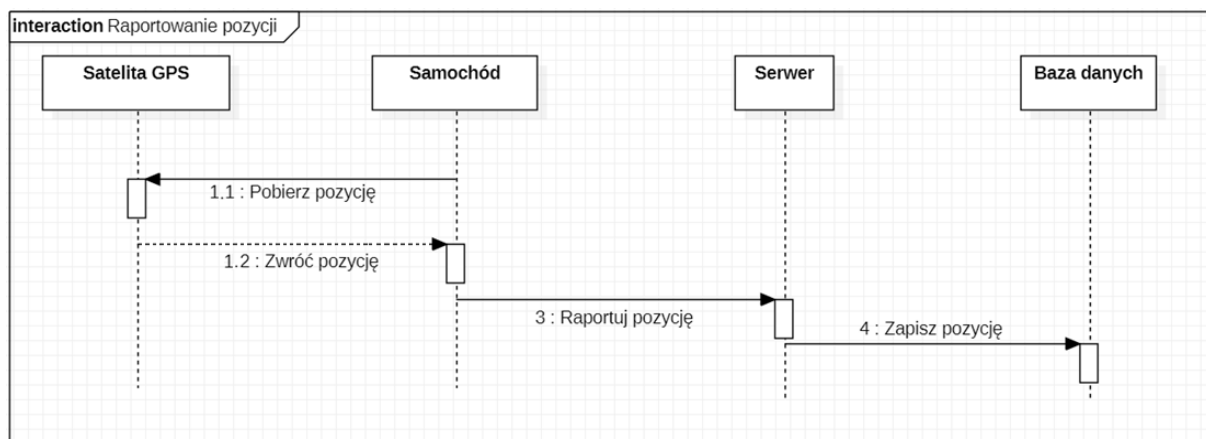


Rysunek 6. Diagram wdrożenia serwisu "PKkierowca"

4. Diagramy sekwencji

4.1 Raportowanie pozycji i prędkości

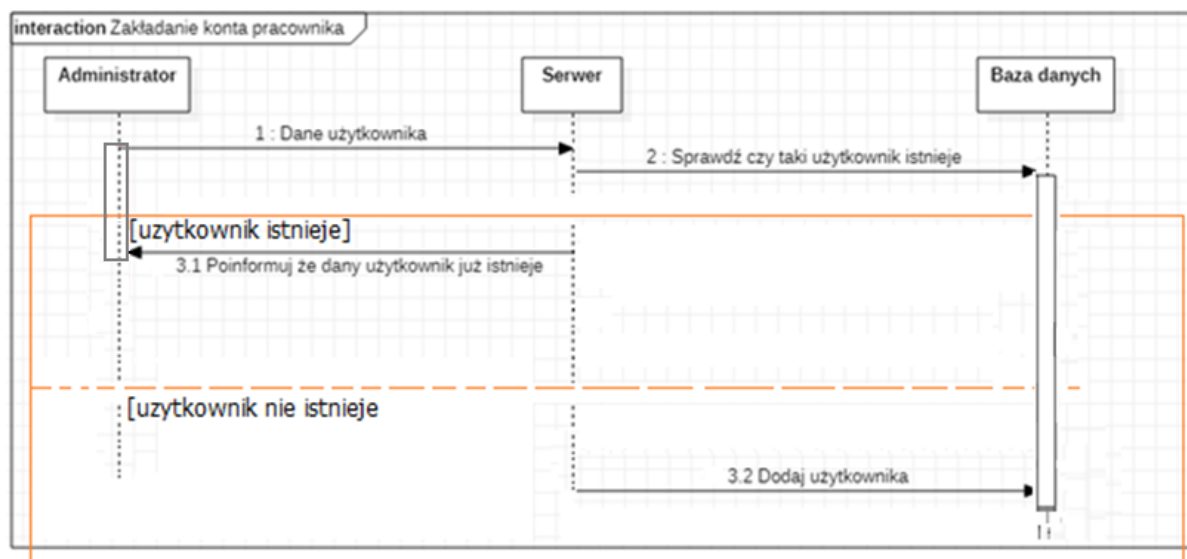
Na rysunku siódmym został przedstawiony diagram sekwencji dla czynność „Raportowanie pozycji i prędkości pojazdu”



Rysunek 7. Diagram sekwencji - Raportowanie pozycji i prędkości pojazdu

4.2 Zakładanie kont użytkowników

Na rysunku ósmym został przedstawiony diagram sekwencji dla czynność „Zakładanie kont użytkowników”.

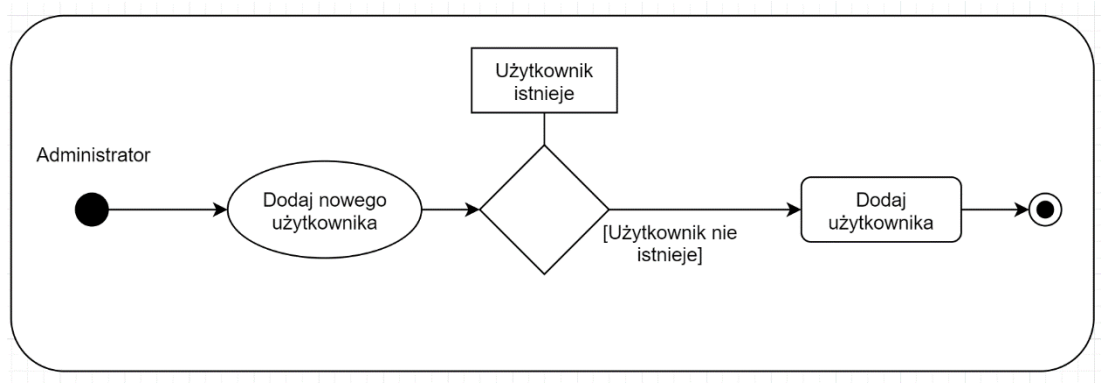


Rysunek 8. Diagram sekwencji - Zakładanie kont użytkowników

5. Diagramy aktywności

5.1 Zakładanie kont użytkowników

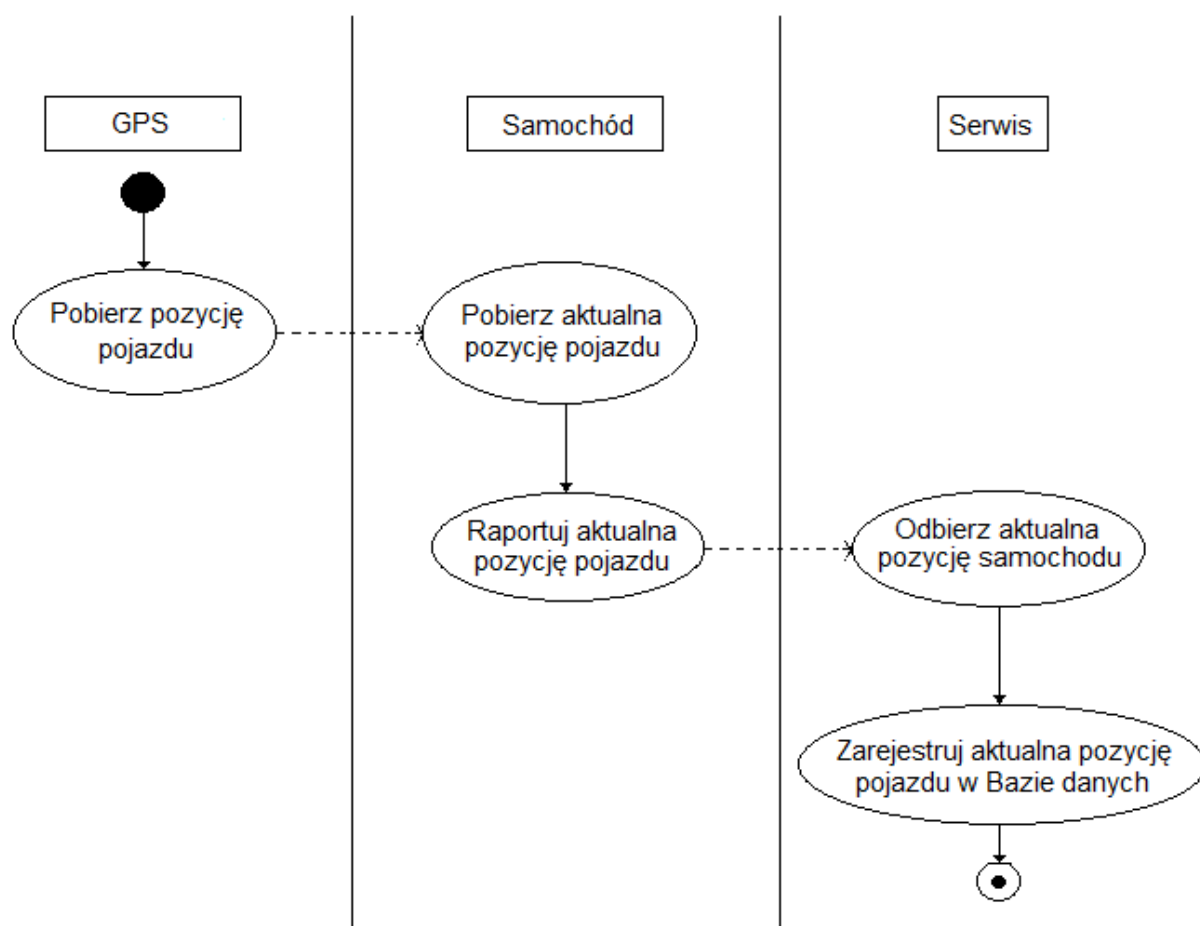
Na rysunku dziewiątym został przedstawiony diagram aktywności dla czynność „Zakładanie kont użytkowników”.



Rysunek 9. Diagram aktywności - Zakładanie kont użytkowników

5.2 Raportowanie pozycji pojazdu

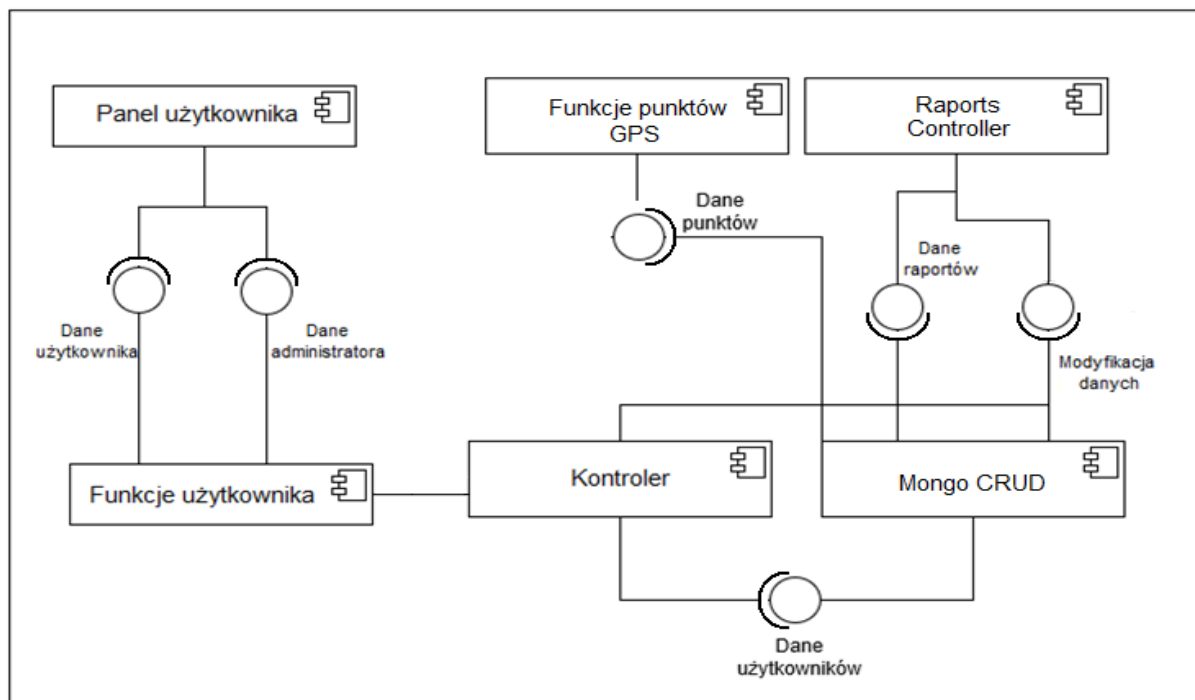
Na rysunku dziesiątym został przedstawiony diagram aktywności dla czynność „Raportowanie pozycji pojazdu”.



Rysunek 10. Diagram aktywności - raportowanie pozycji pojazdu

6. Diagram komponentów

Na rysunku jedenastym został przedstawiony diagram komponentów serwisu „PKkierowca”



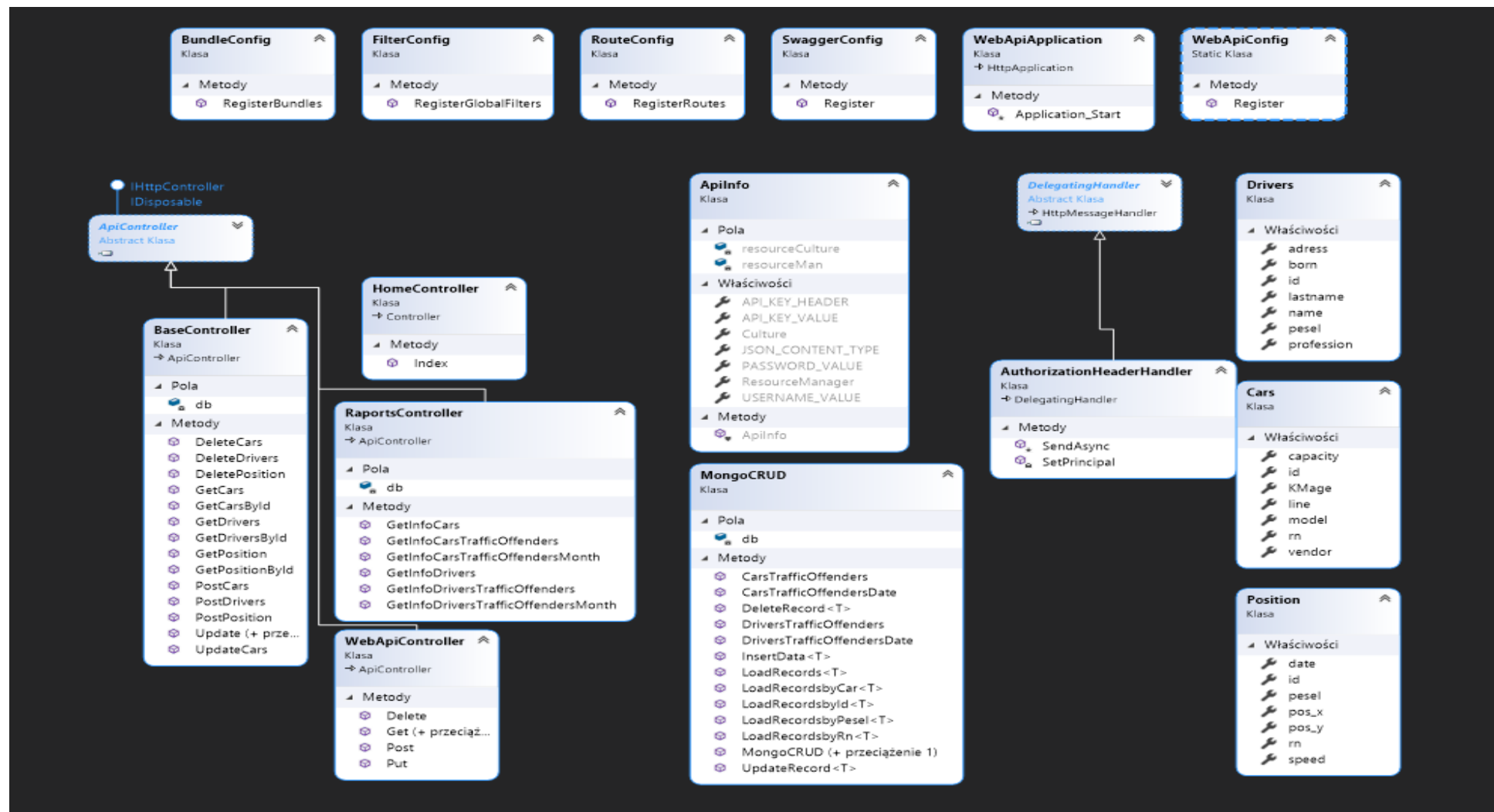
Rysunek 11. Diagram komponentów serwisu "PKkierowca"

Funkcja użytkownika wyświetla potrzebne dane w odpowiednich miejscach oraz przechowuje zapytania tworzone przez użytkownika. Moduł funkcji użytkownika odpowiada za tworzenie zapytań oraz przechwytywanie odpowiedzi od serwera. Moduł Sieć zapewnia szyfrowane połączenie o potrzebnych parametrach. Kontroler natomiast wychwytuje próby logowania użytkowników i odpowiada na nie by w późniejszym etapie realizować funkcjonalność serwisu.

Moduł Funkcje punktów współrzędnych GPS oraz raportów realizuje główne funkcje związane z dodawaniem punktów współrzędnych pojazdów i raportów na temat ich miejsca w czasie rzeczywistym jak i przekroczeń prędkości. Baza danych to moduł odpowiedzialny za bazę danych w której przechowywane są dane aplikacji.

7. Diagram klas

Na rysunkach poniżej znajdują się diagram klasy aplikacji. Diagram przedstawia klasy obiektów w programie, został on wygenerowany bezpośrednio z kodu programu, stanowiąc jego jawne odzwierciedlenie w postaci graficznej pokazuje strukturę systemu.

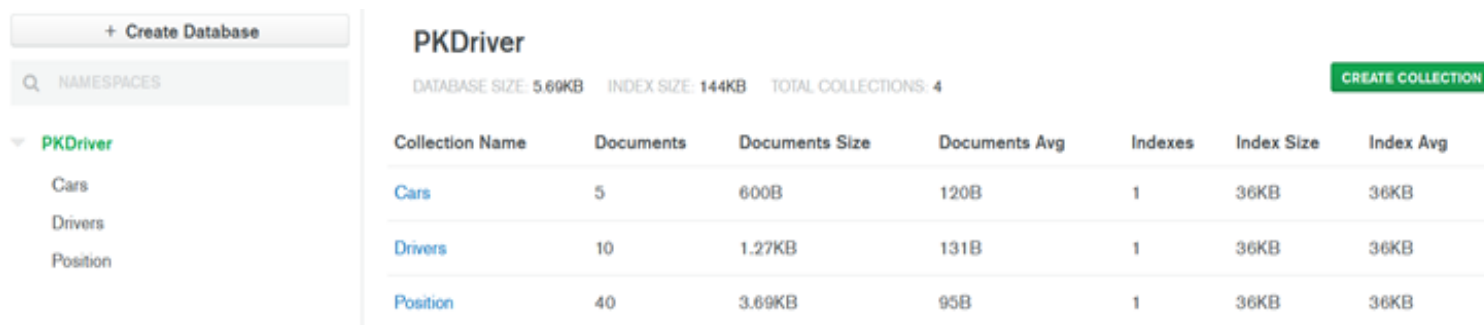


Rysunek 12. Diagram klas

8. Baza Danych

Aplikacja będzie wykorzystywała nierelacyjną bazę danych – MongoDB. MongoDB charakteryzuje się dużą skalowalnością, wydajnością oraz brakiem ściśle zdefiniowanej struktury obsługiwanych baz danych. Zamiast tego dane składowane są jako dokumenty w stylu JSON, co umożliwia aplikacjom bardziej naturalne ich przetwarzanie, przy zachowaniu możliwości tworzenia hierarchii oraz indeksowania.

Baza danych aplikacji – PKDriver składać się będzie z trzech kolekcji i będzie tworzona w tak zwanej chmurze Atlas. MongoDB Atlas to w pełni zarządzana baza danych w chmurze opracowana przez te same osoby, które tworzą MongoDB. Atlas obsługuje całą złożoność wdrażania, zarządzania Bazą danych.



The screenshot shows the MongoDB Atlas interface for a database named 'PKDriver'. On the left, there is a sidebar with a search bar labeled 'NAMESPACES' and a list of collections: 'Cars', 'Drivers', and 'Position'. The main area displays a table with the following data:

| Collection Name | Documents | Documents Size | Documents Avg | Indexes | Index Size | Index Avg |
|-----------------|-----------|----------------|---------------|---------|------------|-----------|
| Cars | 5 | 600B | 120B | 1 | 36KB | 36KB |
| Drivers | 10 | 1.27KB | 131B | 1 | 36KB | 36KB |
| Position | 40 | 3.69KB | 95B | 1 | 36KB | 36KB |

At the top right of the main area, there is a green button labeled 'CREATE COLLECTION'. Above the table, the database statistics are shown: 'DATABASE SIZE: 5.69KB', 'INDEX SIZE: 144KB', and 'TOTAL COLLECTIONS: 4'.

Rysunek 13. Widok kolekcji w usłudze MongoDB Atlas

Kolekcje bazy danych PKDriver:

- Cars
- Drivers
- Position

Każda kolekcja będzie przechowywać odpowiednie dane.

Na rysunku piętnastym został zaprezentowany model danych w bazie danych serwisu „PKkierowca” to właśnie w tych kolekcjach aplikacja będzie składowała dane z których będzie korzystać podczas pracy



Rysunek 14. Model bazy danych

Kolekcja „Cars” – dane na temat:

- id(objectid) – identyfikator dodawany automatycznie
- vendor(string) - marki samochodu
- model varchar) - modelu samochodu
- capacity(double) - pojemności silnika
- rn varchar) - numeru rejestracyjnego
- KMage(double) - licznika kilometrów

```
_id: ObjectId("5e9f09df1c9d4400009be6b5")
vendor: "Ford"
model: "Focus"
line: "ST-line"
capacity: 1.8
KMage: 36000
rn: "ZKL101PW"
```


Kolekcja „Drivers” – dane na temat:

- id (objectId)– identyfikator dodawany automatycznie
- name(varchar) - imie kierowcy
- lastname(varchar) - nazwisko kierowcy
- born(date) - data urodzenia
- address(varchar) - adres zamieszkania
- profession(varchar) - wydział zatrudnienia
- pesel(string) - identyfikatorze pracownika kierującego pojazdem

```
_id: ObjectId("5e9f092a1c9d4400009be6b4")
name: "Adrian"
lastname: "Ameran"
born: 1995-04-08T22:00:00.000+00:00
adress: "Pionierów 13/8"
profession: "IT"
```

Kolekcja „Position” – dane na temat:

- id(objectid) – identyfikator dodawany automatycznie
- pos_x(double) - wartości współrzędnych punktu, mogą mieć wartość zmiennoprzecinkową
- pos_y(double) - wartości współrzędnych punktu, mogą mieć wartość zmiennoprzecinkową
- rn (varchar)- numerze rejestracyjnym pojazdu
- pesel(string) - identyfikatorze pracownika kierującego pojazdem
- speed (int)- prędkości pojazdu
- date (date)- czasie rzeczywistym

```
_id: ObjectId("5ea070591c9d4400000227e5")
pos_x: 15
pos_y: 17
rn: "ZKL101PW"
pesel: "9405923858"
speed: 87
date: 2020-03-03T13:56:00.000+00:00
```