



# Specyfikacja wymagań dla projektu Samochody ciężarowe, wersja 0.0.8

**Autor: Igor Nowicki**

## Opis systemu

System służy do przechowywania definicji pojazdów ciężarowych i ich zespołów, wraz z informacjami o załadunku. Przewidziana jest możliwość przechowywania pojazdów różnego rodzaju (samochody z przestrzenią ładunkową, ciągniki siodłowe, przyczepy, naczepy), z różnego rodzaju przestrzeniami ładunkowymi (zdefiniowane jako skrzyniowa, kontenerowa, cysterna). Program przewiduje mechanizm przechowywania danych na temat pojazdu pomiędzy sesjami poprzez zapis i odczyt z i do pliku dyskowego w formacie JSON.

Program realizuje następujące działania:

- Logowanie i wylogowanie operatora
- Logowanie i wylogowanie administratora
- Wczytanie stanu systemu z repozytorium do pamięci
- Zapis stanu systemu z pamięci do repozytorium
- Wprowadzenie nowego pojazdu
- Skasowanie istniejącego pojazdu
- Zarządzanie pojazdami przez operatora:
  - dodawanie pojazdów
  - edycja pojazdów
  - usuwanie pojazdów
  - wyświetlanie danych o pojazdach
  - łączenie i rozłączanie pojazdów w zespoły pojazdów według zdefiniowanych zasad
  - ładunek i rozładunek pojazdów z przestrzenią ładunkową według zdefiniowanych zasad
- Zarządzanie operatorami przez administratora:
  - dodawanie operatorów

- edycja operatorów
- usuwanie operatorów
- wyświetlanie danych o operatorach

## Słownik dziedziny problemu

- Pojazd (Vehicle) - abstrakcyjny obiekt określający dowolną jednostkę jeżdżącą.
- Pojazd silnikowy (Motorized Vehicle) - abstrakcyjny obiekt dziedziczący z Pojazdu, opisujący dowolny pojazd posiadający silnik.
- Pojazd bezsilnikowy (Non Motorized Vehicle) - abstrakcyjny obiekt dziedziczący z Pojazdu, opisujący dowolny pojazd nieposiadający silnika.
- Pojazd ładunkowy (Loadable) - abstrakcyjny pojazd posiadający możliwość przechowywania ładunku.
- Pojazd przyczepowy (Trailerable) - abstrakcyjny pojazd posiadający możliwość doczepienia przyczepy.
- Pojazd naczepowy (Semi-Trailerable) - abstrakcyjny pojazd posiadający możliwość doczepienia naczepy.
- Przyczepa (Trailer) - pojazd bezsilnikowy, ładunkowy, z możliwością bycia przyczepionym do pojazdu przyczepowego.
- Naczepa (SemiTrailer) - pojazd bezsilnikowy, ładunkowy, z możliwością bycia przyczepionym do pojazdu naczepowego.
- Ciągnik siodłowy (Truck) - pojazd silnikowy, naczepowy.
- Samochód ciężarowy (TrailerCar) - pojazd silnikowy, przyczepowy.
- Samochód z przestrzenią ładunkową (Cargo Space Car) - pojazd silnikowy, ładunkowy.
- Operator - użytkownik z możliwością zarządzania bazą danych pojazdów.
- Administrator - użytkownik z możliwością zarządzania bazą danych użytkowników.

## Obiekty Biznesowe

### Pojazd (Vehicle)

Opis: Abstrakcyjny obiekt określający dowolną jednostkę jeżdżącą.

Każdy pojazd opisywany jest zestawem parametrów:

- Id (liczba całkowita) - Identyfikator (liczba całkowita dodatnia).
- Name (tekst) - Nazwa (tekst na podstawie pozycji katalogowej producenta).
- VIN (tekst) - Vehicle Identification Number (15 lub 17 znaków alfanumerycznych).
- Year - Rok produkcji (czterocyfrowa liczba całkowita w zakresie od roku 1900 do roku bieżącego).

## Pojazd silnikowy (MotorizedVehicle)

Opis: Abstrakcyjny obiekt dziedziczący z Pojazdu, opisujący dowolny pojazd posiadający silnik.

- Engine (Engine) - parametry silnika (poniżej).

## Silnik (Engine)

Opis: Obiekt definiowany wewnątrz pojazdu silnikowego. Definiowany następującymi parametrami:

- Name (tekst) - Nazwa (tekst na podstawie pozycji katalogowej producenta).
- Capacity (liczba całkowita) - Pojemność (liczba całkowita dodatnia, cm3).
- Power (liczba całkowita) - Liczba koni mechanicznych (liczba całkowita dodatnia).

## Pojazd bezsilnikowy (NonMotorizedVehicle)

Opis: Abstrakcyjny obiekt dziedziczący z Pojazdu, opisujący dowolny pojazd nieposiadający silnika.

## Pojazd ładunkowy (Loadable)

Opis: Interfejs zawierający przestrzeń ładunkową dla pojazdu.

- Storage (Storage) - Przestrzeń ładunkowa (poniżej)

## Przestrzeń ładunkowa (Storage)

Opis: Abstrakcyjny obiekt definiowany wewnątrz pojazdu z którego dziedziczą opisane poniżej typy przestrzeni ładunkowej.

- StorageDescription (tekst) - Opis przestrzeni ładunkowej. Definiowany na poziomie

definicji klasy.

- Items (List of Items) - Lista towarów.
- IsSingle (bool) - Czy przestrzeń ładunkowa może zawierać jeden typ towarów.

## Towar (Item)

Abstrakcyjny obiekt opisujący dowolną jednostkę towarową. Wyszczególnione są następujące obiekty pochodne:

- BoxItem - Towar typu skrzyniowego
- ContainerItem - Towar typu kontenerowego
- LiquidItem - Towar typu płynnego

Parametry:

- Name (tekst) - Nazwa towaru
- GrossMass (decimal) - Waga brutto towaru
- Description (tekst) - Opis towaru

Metody:

- GetDimensions (tekst) - abstrakcyjna funkcja definiowana na poziomie funkcji pochodnych zwracająca opis fizycznej przestrzeni zajmowanej przez towar.

## Skrzyniowa przestrzeń ładunkowa (BoxStorage)

Opis: Typ przestrzeni ładunkowej. Umożliwia załadunek towaru typu skrzyniowego.

- StorageDescription (tekst) - parametr, zwraca "Przestrzeń skrzyniowa"
- IsSingle (bool) - zwraca "fałsz"

## Towar typu skrzyniowego (BoxItem)

Typ towaru, ładowany do skrzyniowej przestrzeni ładunkowej.

- Nazwa (Name) (tekst na podstawie pozycji katalogowej producenta)
- Masa brutto (GrossMass) (liczba zmiennoprzecinkowa, w tonach).
- Width - Szerokość (w cm)
- Height - Wysokość (w cm)
- Length - Długość (w cm)

## Kontenerowa przestrzeń ładunkowa (ContainerStorage)

Opis: Typ przestrzeni ładunkowej. Umożliwia załadunek towaru typu kontenerowego. Opisany następującymi parametrami:

- StorageDescription (tekst) - parametr, zwraca "Przestrzeń kontenerowa"
- IsSingle (bool) - parametr, zwraca "prawda"

## Towar typu kontenerowego (ContainerItem)

Typ towaru, ładowany do kontenerowej przestrzeni ładunkowej. Metody:

- GetDimensions() - zwraca "-" (standardowy wymiar kontenera)

## Cysternowa przestrzeń ładunkowa (TankStorage)

Opis: Typ przestrzeni ładunkowej. Umożliwia załadunek towaru typu płynnego.

- StorageDescription (tekst) - parametr, zwraca "Cysterna".

## Towar typu płynnego (LiquidItem)

Typ towaru, ładowany do cysternowej przestrzeni ładunkowej. Opisany listą parametrów:

- Volume - Objętość (w litrach, opisywana liczbą całkowitą dodatnią)

## Aktorzy Biznesowi

Aktorzy biznesowi przetwarzają obiekty biznesowe lub są z nimi w interakcji.

Id	Nazwa	Opis
A1	Operator	Podstawowy użytkownik aplikacji. Może przeglądać listy pojazdów i informacje na temat każdego z pojazdów. Może przeprowadzać załadunek i rozładunek pojazdów oraz łączenie i rozłączanie zespołów pojazdów.
A2	Administrator	Użytkownik z uprawnieniami administratora. Może zarządzać użytkownikami systemu – dodawać, usuwać, edytować informacje i wyświetlać listę zarejestrowanych użytkowników i administratorów

Id	Nazwa	Opis
		systemu.

## Reguły Biznesowe

ID	Definicja reguły	Typ	Źródło
A1	Do skrzyniowej przestrzeni ładunkowej można ładować jedynie towary typu skrzyniowego	ograniczenie	Definicja
A2	Do kontenerowej przestrzeni ładunkowej można ładować jedynie towary typu kontenerowego	ograniczenie	Definicja
A3	Do cysternowej przestrzeni ładunkowej można ładować jedynie towary typu płynnego	ograniczenie	Definicja
B1	Skrzyniowa przestrzeń ładunkowa może przechowywać wiele towarów.	definicja	Rozmowa w trakcie zajęć
B2	Kontenerowa przestrzeń ładunkowa może przechowywać tylko jeden towar.	ograniczenie	Rozmowa w trakcie zajęć
B3	Cysterna może być załadowana jednym rodzajem płynu.	ograniczenie	Rozmowa w trakcie zajęć
C1	Przyczepę można przyczepić jedynie do pojazdu typu przyczepowego	ograniczenie	-
C2	Naczepę można przyczepić jedynie do pojazdu typu naczepowego	ograniczenie	-

## Wymagania Funkcjonalne

Rozdział zawiera spis wymagań funkcjonalnych dla tworzonego produktu informatycznego.

ID	Nazwa	Priorytet	Proces biznesowy
F1	Operator może wyświetlać	Wysoki	Operator może wyświetlać listę

ID	Nazwa	Priorytet	Proces biznesowy
	listę pojazdów.		pojazdów.
F2	Operator może wyświetlać dane pojedynczego pojazdu.	Wysoki	Operator może wyświetlać dane pojedynczego pojazdu.
F3	Operator może stworzyć nowy pojazd.	Wysoki	Operator może stworzyć nowy pojazd.
F4	Operator może zaktualizować dane pojazdu.	Wysoki	Operator może zaktualizować dane pojazdu.
F5	Operator może skasować pojazd.	Wysoki	Operator może skasować pojazd.
F6	Złączanie pojazdów (przyczepa)	Wysoki	Operator może złączyć pojazd typu ciągnikowego z przyczepą.
F7	Rozłączanie pojazdów (przyczepa)	Wysoki	Operator może rozłączyć pojazd typu ciągnikowego z przyczepą.
F8	Złączanie pojazdów (naczepa)	Wysoki	Operator może złączyć pojazd typu siodłowego z naczepą.
F9	Rozłączanie pojazdów (naczepa)	Wysoki	Operator może rozłączyć pojazd typu siodłowego z naczepą.
F10	Załadunek pojazdu	Wysoki	Operator może przeprowadzić załadunek pojazdu z przestrzenią ładunkową.
F11	Rozładunek pojazdu	Wysoki	Operator może przeprowadzić rozładunek pojazdu z przestrzenią ładunkową.
F12	Zapis i odczyt z pliku	Wysoki	Operator może zapisać/odczytać dane do/z pliku.

ID	Nazwa	Priorytet	Proces biznesowy
F13	Autoryzacja użytkownika	Wysoki	Operator autoryzuje się przed rozpoczęciem pracy.
F14	Dodawanie użytkownika	Wysoki	Administrator może dodać użytkownika do systemu
F14	Usuwanie użytkownika	Wysoki	Administrator może usunąć użytkownika z systemu.

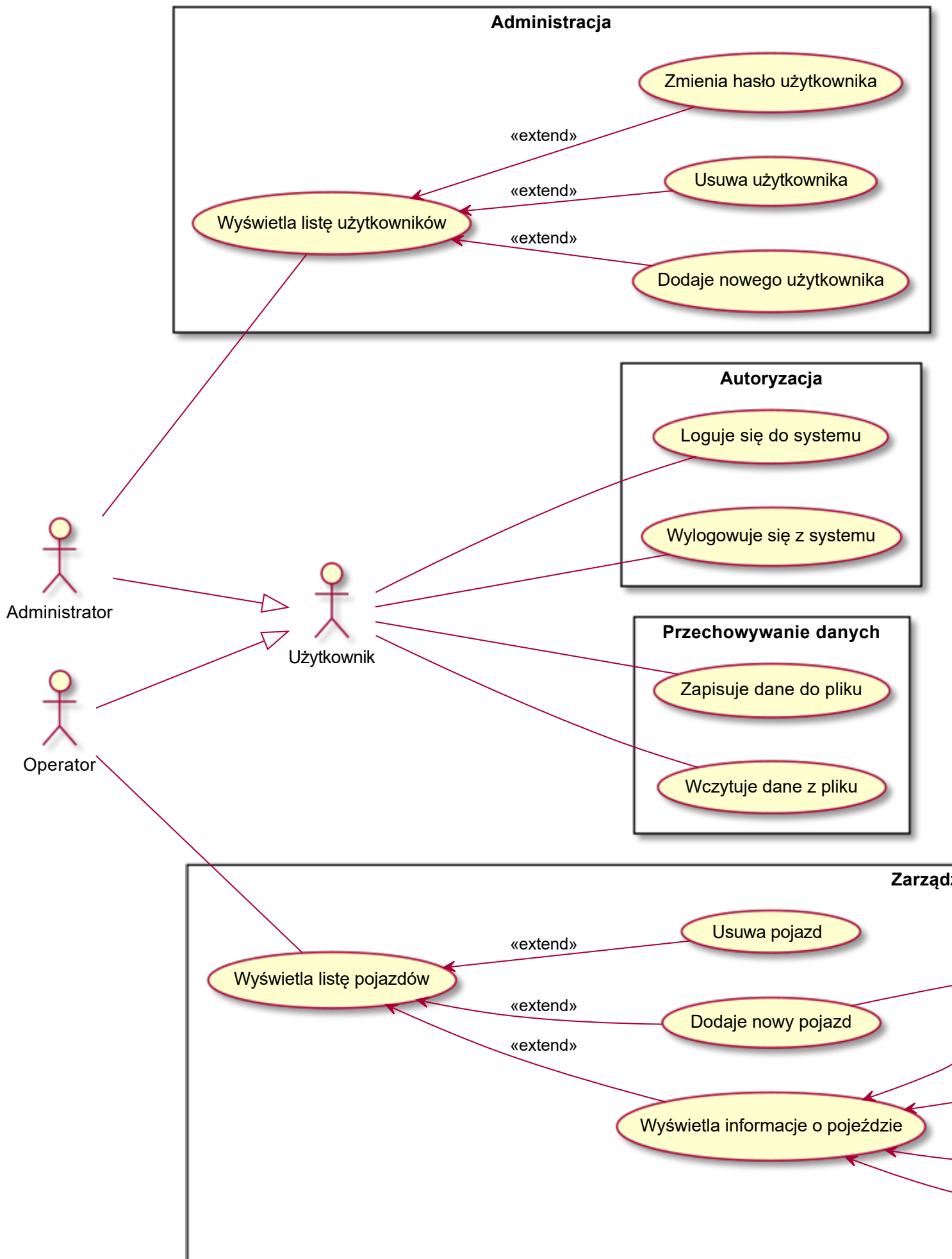
## Wymagania Niefunkcjonalne

W tym rozdziale należy umieścić wszystkie wymagania niefunkcjonalne pamiętając, aby tak je sformułować, aby w przyszłości była możliwość ich weryfikacji.

ID	Nazwa	Priorytet	Proces biznesowy
N1	Platforma Windows 10	Wysoki	Program ma działać na platformie Windows 10
N2	Windows Presentation Foundation	Wysoki	Program ma zostać zaprojektowany dla platformy Windows Presentation Foundation
N3	Interfejs graficzny	Wysoki	Interfejs programu ma być w postaci graficznej (GUI)
N4	Wymagania zapisu	Wysoki	Program ma automatycznie zapisywać stan magazynu przy zmianach
N5	Wymagania odczytu	Wysoki	Program ma automatycznie wczytywać stan magazynu przy zmianach



# Diagrammy Use Case

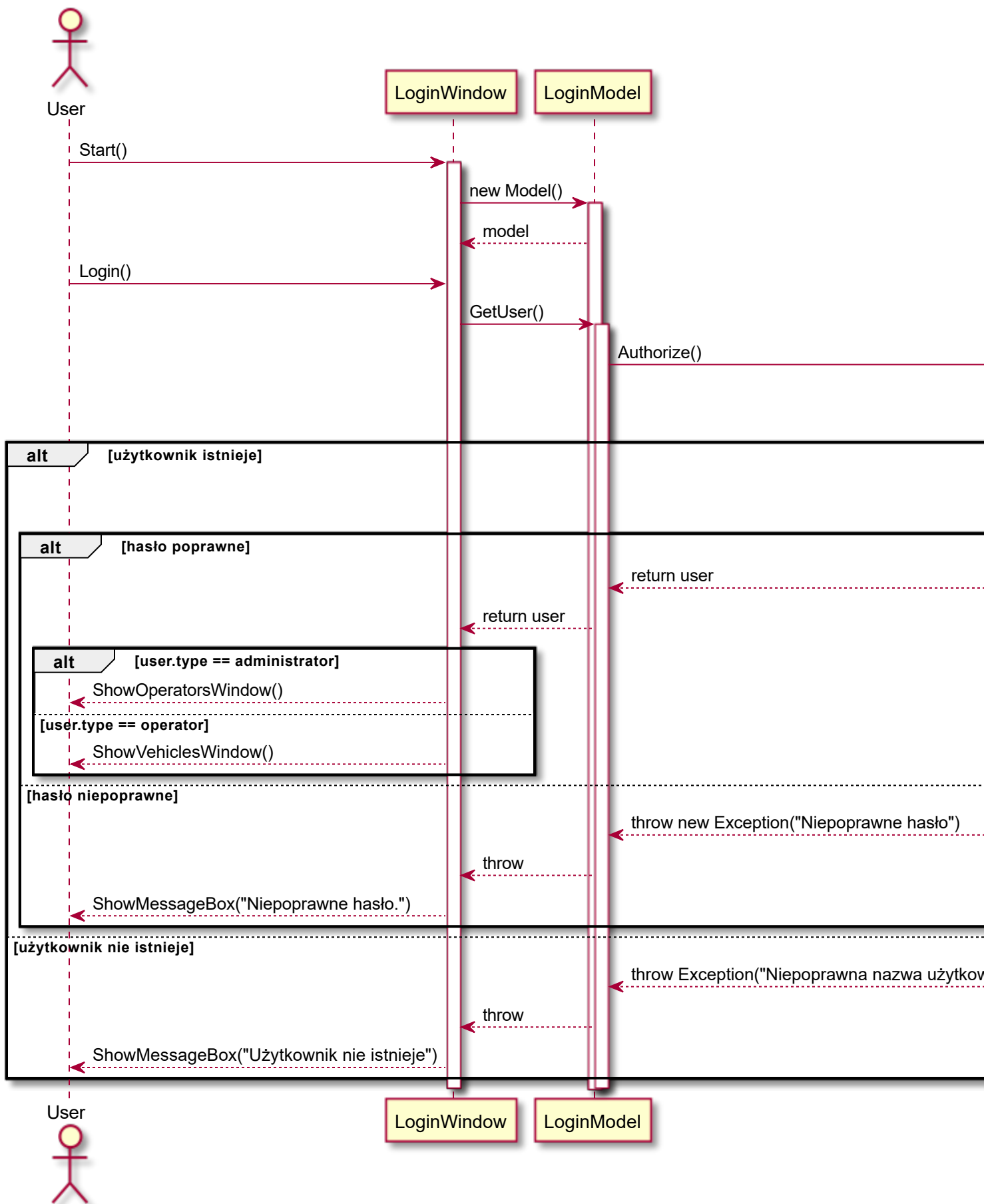


# Diagram klas

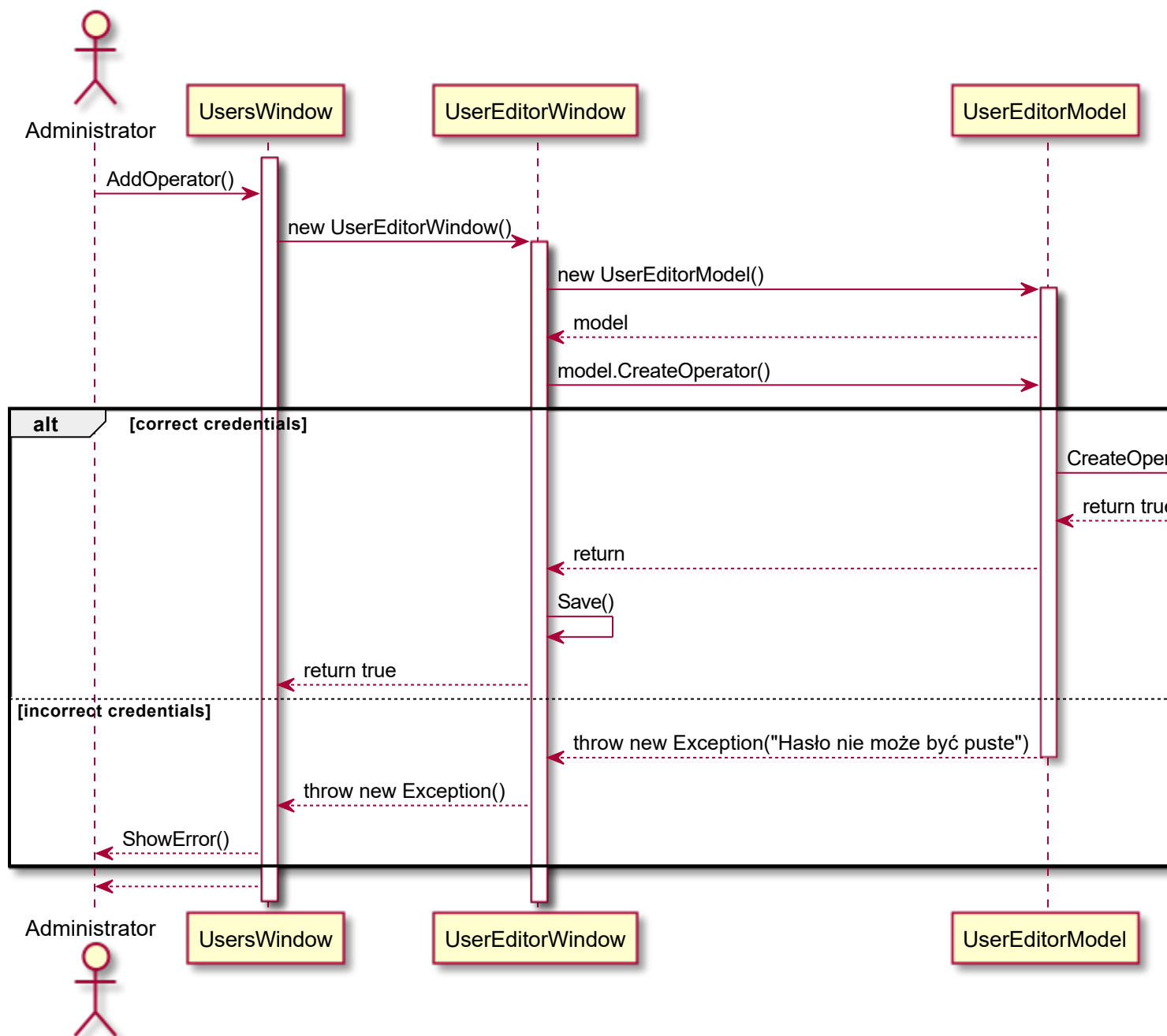
```
classDiagram
class VehicleBuilder
VehicleBuilder .. VehicleProxy
class VehicleProxy
VehicleProxy -- Vehicle
class ILoadable{ <<interface>> ILoadable }
class Vehicle{
<<abstract>> # Vehicle(); # Vehicle(Vehicle); + Id: int; + Name: string; + VIN: string; + Year: int;
+ TypeDescription*: string; + ToString(): string; }
class MotorizedVehicle{ <<abstract>> #
MotorizedVehicle(); # MotorizedVehicle(Vehicle vehicle); # MotorizedVehicle(MotorizedVehicle
vehicle); +Engine Engine; }
class NonMotorizedVehicle{ <<abstract>> #
NonMotorizedVehicle(); # NonMotorizedVehicle(Vehicle vehicle); }
Vehicle <|--
MotorizedVehicle
Vehicle <|-- NonMotorizedVehicle
MotorizedVehicle <|-- TrailerCar
MotorizedVehicle <|-- Truck
MotorizedVehicle <|-- CargoSpaceCar
CargoSpaceCar --|>
ILoadable
SemiTrailer --|> ILoadable
Trailer --|> ILoadable
NonMotorizedVehicle <|--
SemiTrailer
NonMotorizedVehicle <|-- Trailer
class ISemiTrailerable{ <<interface>> +int
SemiTrailerId }
Truck --|> ISemiTrailerable
class ITrailerable{ <<interface>> +int TrailerId }
TrailerCar ..|> ITrailerable
ILoadable -- Storage
class Storage{ <<abstract>> +
StorageDescription*: string; + Items: List<Item> ; + IsSingle*: bool ; }
Storage <|--
ContainerStorage
Storage <|-- BoxStorage
Storage <|-- TankStorage
class Engine{ +Engine()
+Engine(Engine) +string Name +int Capacity +int Power }
Engine --* MotorizedVehicle
class ContainerStorage{ + ContainerStorage(); + ContainerStorage(ContainerStorage); + string
StorageDescription + bool IsSingle }
class BoxStorage{ + BoxStorage(); +
BoxStorage(BoxStorage ); + StorageDescription: string + IsSingle: bool }
class TankStorage{ +
TankStorage(); + TankStorage(TankStorage); + StorageDescription : string + IsSingle: bool }
class Item{ <<abstract>> Item +Name : string +GrossMass : float }
class LiquidItem{ +
LiquidItem(); + LiquidItem(LiquidItem item) ; + Volume: decimal; + GetDimensions(): string; }
class BoxItem{ +Width: int +Height: int +Length: int }
ContainerStorage -- ContainerItem
BoxStorage -- BoxItem
TankStorage -- LiquidItem
ContainerItem --|> Item
BoxItem --|> Item
LiquidItem --|> Item
class Vehicle{ +Id : int +Name : string +VIN : string +Year : int +Engine :
Engine +CargoSpace : Storage }
class ItemProxy{ +Id: int +ItemProxy(Item item) +Item: Item
+Name: string +GrossMass: string +Dimensions: string +TypeDescription: string
-GetTypeDescription(): string -GetDimensions(): string }
Item -- ItemProxy
```

# **Diagramy sekwencji**

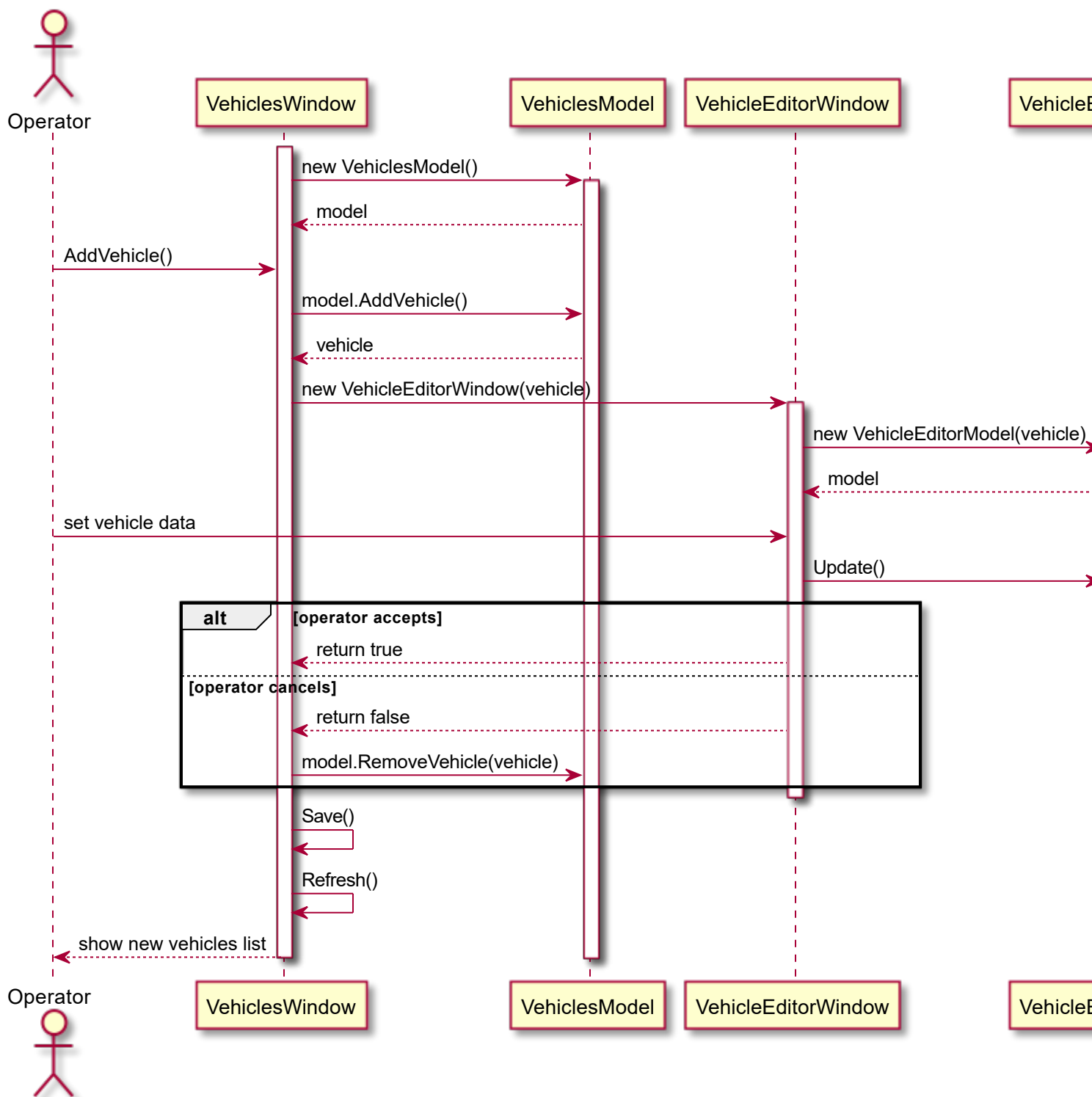
## **Logowanie do systemu**



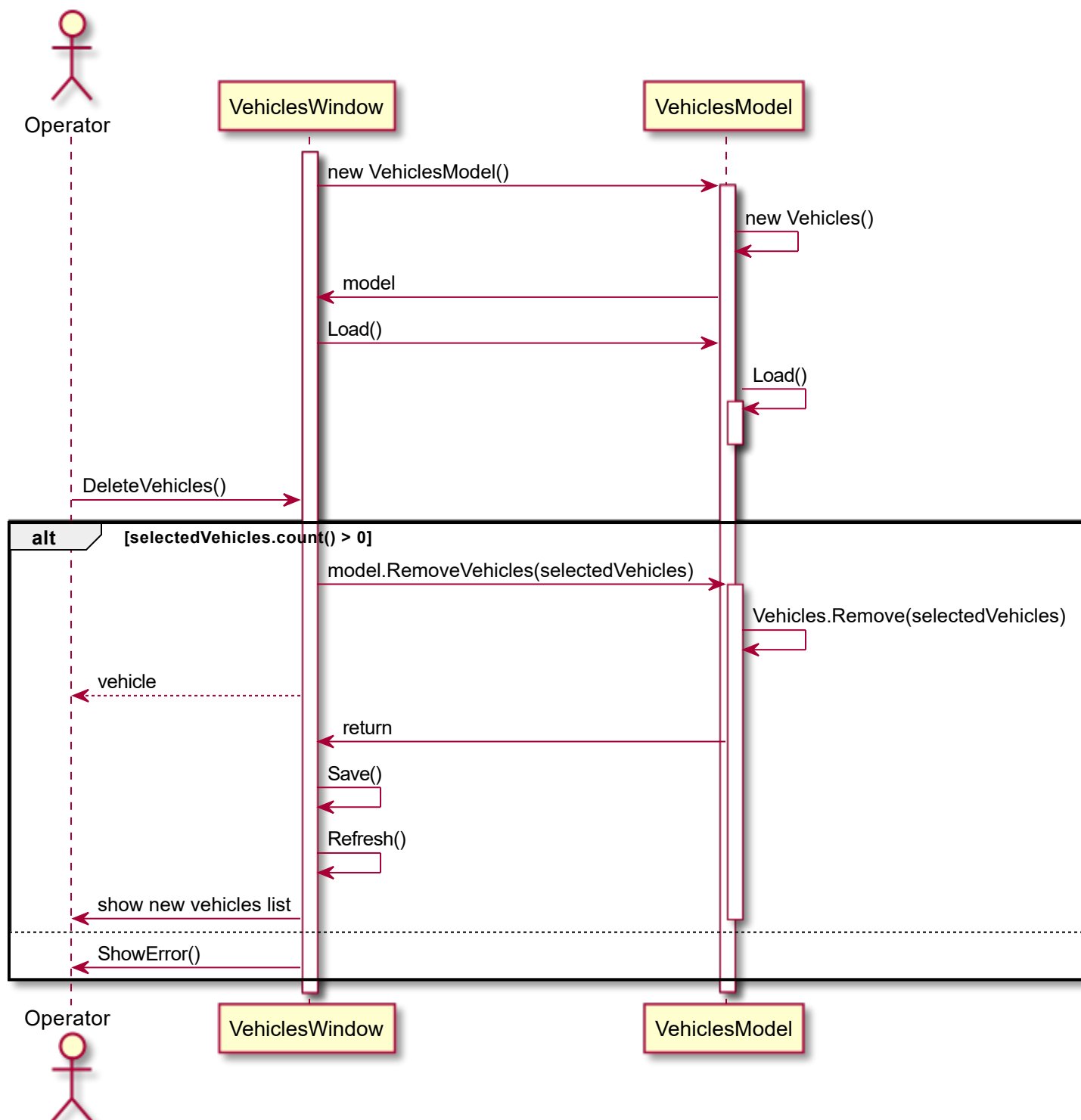
## Dodaj operatora



# Dodaj samochód

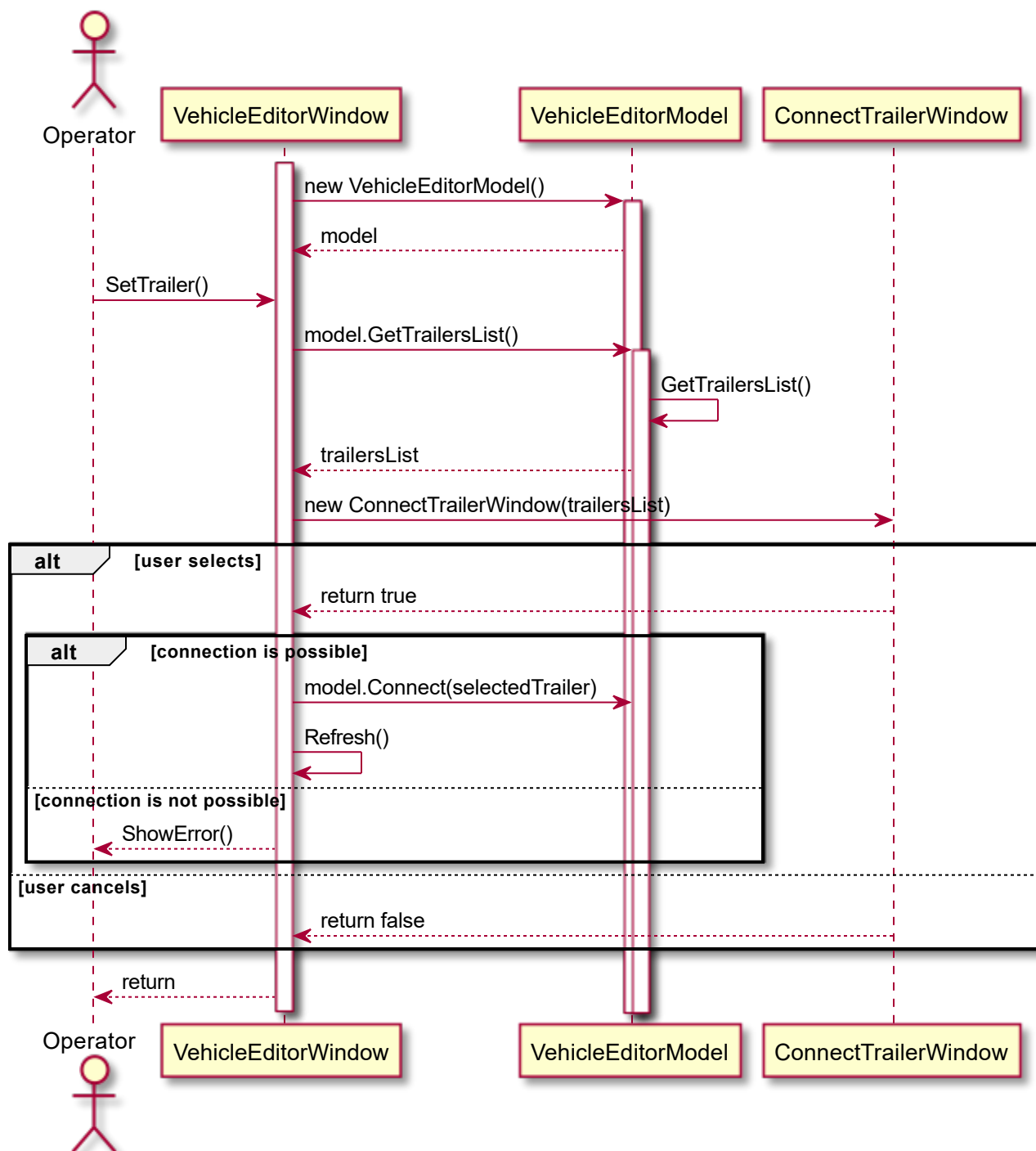


# Usuń samochody

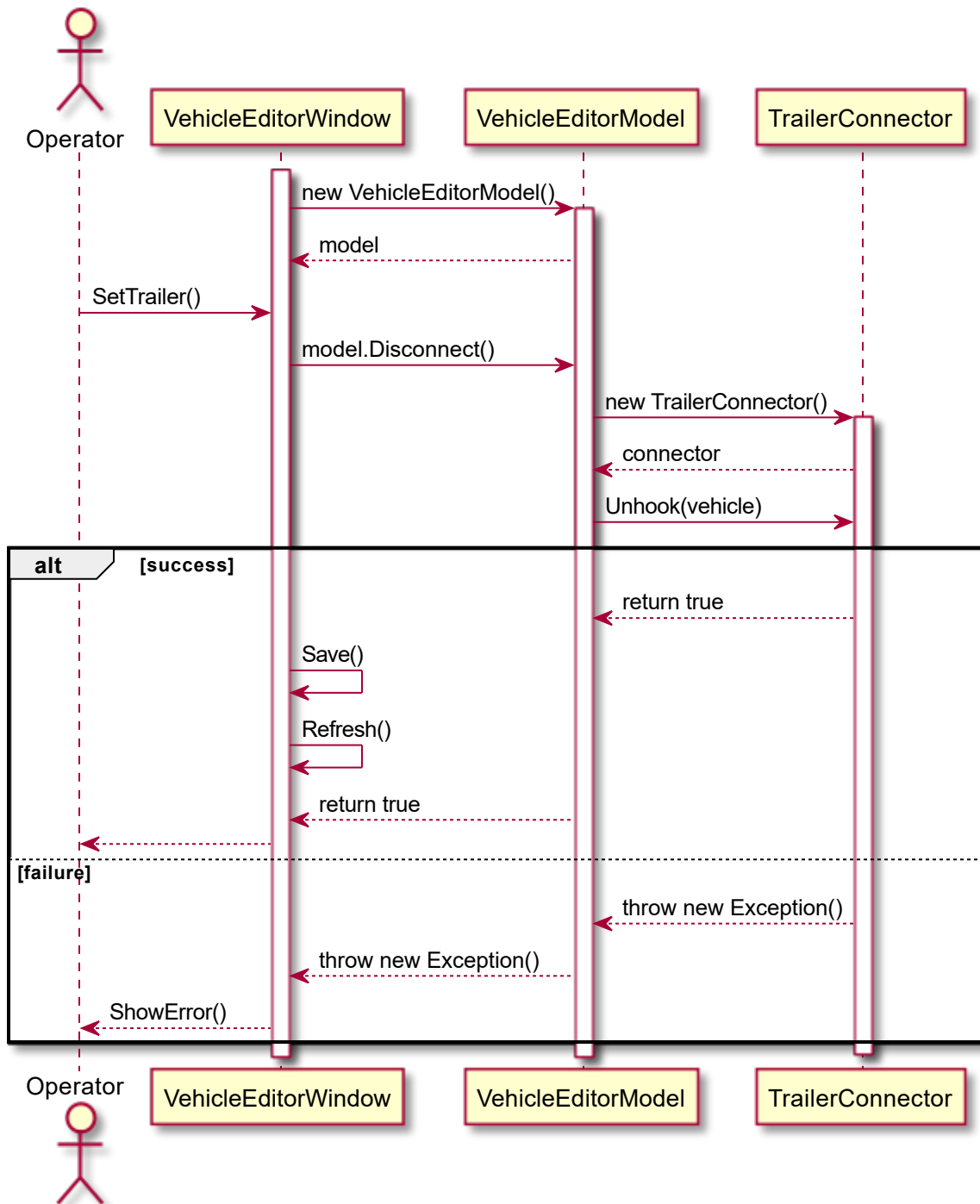




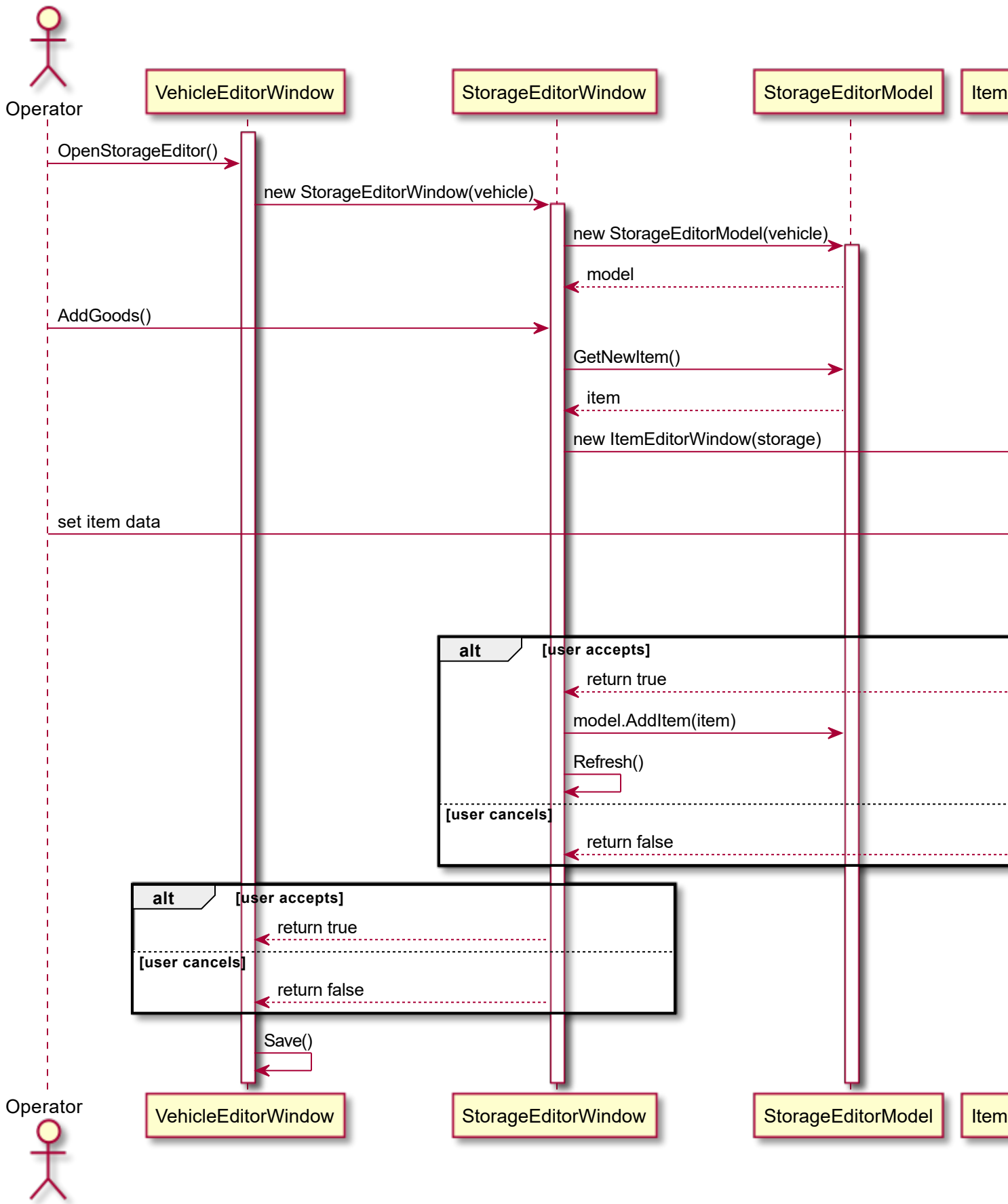
# Łącz pojazdy



# Rozłącz pojazdy



**Przeprowadź załadunek przestrzeni ładunkowej**



# Diagramy maszyny stanowej i czynności

## Logowanie do programu

stateDiagram-v2  
start: Logowanie vehicles: Wyświetlanie listy pojazdów operators:  
Wyświetlanie listy operatorów finish: Zamknięcie programu [\*] --> start  
start --> vehicles : Logowanie operatora start --> operators : Logowanie administratora start --> start : Błąd logowania operators --> start: Wylogowanie administratora vehicles --> start: Wylogowanie operatora start --> finish: Zakończenie programu finish --> [\*]

## Okno administratora

stateDiagram-v2  
operators\_list: Wyświetlanie listy operatorów edit\_operator: Edycja operatora [\*] --> operators\_list: Logowanie jako administrator operators\_list --> edit\_operator: Dodaj operatora operators\_list --> edit\_operator: Edytuj operatora edit\_operator --> operators\_list: Zapisz operatora operators\_list --> operators\_list: Usuń operatora edit\_operator --> operators\_list: Anuluj edycję

## Okno operatora pojazdów

stateDiagram-v2  
vehicles\_list: Wyświetlanie listy pojazdów edit\_vehicle: Edytowanie pojazdu join\_trailer: Łączenie pojazdów disjoin\_trailer: Rozłączanie pojazdów [\*] --> vehicles\_list: Logowanie jako operator vehicles\_list --> edit\_vehicle: Dodaj pojazd vehicles\_list --> edit\_vehicle: Edytuj pojazd edit\_vehicle --> vehicles\_list: Zapisz pojazd edit\_vehicle --> vehicles\_list: Anuluj edycję pojazdu vehicles\_list --> vehicles\_list: Usuń pojazd edit\_vehicle --> join\_trailer: Połącz z pojazdem join\_trailer --> edit\_vehicle: Edytuj pojazd edit\_vehicle --> disjoin\_trailer: Rozłącz z pojazdem disjoin\_trailer --> edit\_vehicle: Edytuj pojazd

## Okno dodawania towarów

stateDiagram-v2  
edit\_vehicle: Edycja pojazdu edit\_item: Edycja towaru show\_storage: Wyświetlanie przestrzeni ładunkowej [\*] --> edit\_vehicle: Edycja pojazdu edit\_vehicle --> show\_storage: Otwórz przestrzeń ładunkową show\_storage --> edit\_item: Dodawanie towaru show\_storage --> edit\_item: Edycja towaru edit\_item --> show\_storage: Zapisz towar edit\_item --> show\_storage: Anuluj dodawanie towaru

## Diagram klas 2