

# **Dokumentacja projektu zaliczeniowego**

Przedmiot: Inżynieria oprogramowania

Temat: **Aplikacja wspomagająca pracę kierowcy autobusu linii numer 75 w Szczecinie**  
Autorzy: **Michał Kucznerowicz, Michał Niedzielski**  
Grupa: N1-20B  
Kierunek: Informatyka  
Rok akademicki: 2020/2021  
Poziom i semestr: I/4  
Tryb studiów: niestacjonarne

<b>1</b>	<b>Spis treści</b>	
<b>2.</b>	<b>Odnośniki do innych źródeł</b>	<b>3</b>
<b>3.</b>	<b>Słownik pojęć</b>	<b>3</b>
<b>4.</b>	<b>Wprowadzenie</b>	<b>5</b>
4.1.	Cel dokumentacji	5
4.2.	Przeznaczenie dokumentacji	5
4.3.	Opis organizacji lub analiza rynku	5
4.4.	Analiza SWOT organizacji	5
<b>5.</b>	<b>Specyfikacja wymagań</b>	<b>5</b>
5.1.	Charakterystyka ogólna	6
5.2.	Wymagania funkcjonalne	8
5.3.	Wymagania нефunkcjonalne	33
<b>6.</b>	<b>Zarządzanie projektem</b>	<b>34</b>
6.1.	Zasoby ludzkie	35
6.2.	Harmonogram prac	35
6.3.	Etapy/kamienie milowe projektu	35
<b>7.</b>	<b>Zarządzanie ryzykiem</b>	<b>35</b>
7.1.	Lista czynników ryzyka	36
7.2.	Ocena ryzyka	36
7.3.	Plan reakcji na ryzyko	37
<b>8.</b>	<b>Zarządzanie jakością</b>	<b>38</b>
8.1.	Scenariusze i przypadki testowe	39
<b>9.</b>	<b>Projekt techniczny</b>	<b>49</b>
9.1.	Opis architektury systemu	50
9.2.	Technologie implementacji systemu	50
9.3.	Diagramy UML	51
9.4.	Charakterystyka zastosowanych wzorców projektowych	75
9.5.	Projekt bazy danych	76
9.6.	Projekt interfejsu użytkownika	79
9.7.	Procedura wdrożenia	86
<b>10.</b>	<b>Dokumentacja dla użytkownika</b>	<b>87</b>
<b>11.</b>	<b>Podsumowanie</b>	<b>92</b>
11.1.	Szczegółowe nakłady projektowe członków zespołu	92
<b>12.</b>	<b>Inne informacje</b>	<b>93</b>

## **2 Odnośniki do innych źródeł**

- Zarządzanie projektem - JazzHub
- Wersjonowanie kodu - Github
- System obsługi defektów - Bitbucket
- Platforma kontroli jakości kodu - SonarQube
- Serwer automatyzacji budowania - Jenkins
- Narzędzie do zarządzania testowaniem - TestRail

### 3 Słownik pojęć

**Kierowca autobusu** - pracownik operujący autobusem, którym przewozi pasażerów po wyznaczonej trasie.

**Pasażer** - osoba podróżująca autobusem do wybranego przez nią przystanku znajdującego się na trasie.

**Koordynator ruchu** - pracownik zajmujący się organizacją trasy autobusu oraz pracy kierowców.

**Zespół techniczny** - grupa pracowników zajmująca się naprawą usterek oraz odholowywaniem uszkodzonych autobusów.

**Kierownik** - osoba odpowiedzialna za nadzorowanie najistotniejszych spraw dziejących się na trasie autobusu oraz zarządzająca pracą koordynatora ruchu i kierowców.

**Służba bezpieczeństwa** - służby takie jak policja, straż pożarna czy pogotowie ratunkowe.

**Product Owner** - osoba odpowiedzialna za tworzenie i udoskonalanie wizji produktu, praca nad wymaganiami, planowanie wydań, angażowanie klientów w proces tworzenia produktu, zarządzaniu budżetem, uczestniczenie w spotkaniach Scrum, współpracowanie z zespołem nad produktem.

**UX Designer** - osoba odpowiedzialna za projektowanie doświadczeń użytkownika. Jego celem jest zwiększenie satysfakcji klienta korzystającego z produktu poprzez poprawę jego funkcjonalności i użyteczności.

**Project Manager** - osoba odpowiedzialna za zarządzania projektem - jego planowaniem, realizacją, zamykaniem oraz zarządzaniem pracy deweloperów oraz testerów.

**Software Engineer** - deweloper odpowiedzialny za rozwój oprogramowania.

**QA Engineer** - tester odpowiedzialny za kontrolę jakości oprogramowania.

## 4 Wprowadzenie

### 4.1 Cel dokumentacji

Dokumentacja została stworzona, aby umożliwić zapoznanie się z wymaganiami oraz funkcjonalnościami aplikacji, jak również przedstawienia celu biznesowego. Zawiera ona opis sposobu działania, wykorzystanych technologii w procesie tworzenia oraz przeznaczenia i możliwościach dostarczonego oprogramowania.

### 4.2 Przeznaczenie dokumentacji

Dokumentacja jest przeznaczona dla użytkowników aplikacji, potencjalnych inwestorów oraz przyszłych deweloperów i testerów utrzymujących i rozwijających oprogramowanie.

### 4.3 Opis organizacji lub analiza rynku

System będzie realizowany dla Zarządu Dróg i Transportu Miejskiego w Szczecinie, organizacji zajmującej się obsługą transportu drogowego na terenie Szczecina m.in. kluczowej z perspektywy projektu linii numer 75. ZDiTM posiada rozbudowany system informacji pasażerskiej odpowiedzialny za informowanie użytkowników o opóźnieniach, czy formach płatności za przejazdy, dlatego wprowadzając nasz system organizacja chciałaby wkomponować go w już istniejące rozwiązania, rozszerzając jego funkcjonalności.

### 4.4 Analiza SWOT organizacji

<b>Mocne strony (S):</b> <ul style="list-style-type: none"><li>- monopol na obsługę transportu autobusowego na terenie Szczecina</li><li>- duże miasto z rozwijającą się aglomeracją</li></ul>	<b>Słabe strony (W):</b> <ul style="list-style-type: none"><li>- ograniczony rozwój ze względu na zależność od budżetu miejskiego</li><li>- trudność pokrycia całego miasta siatką połączeń ze względu na wielką powierzchnię</li></ul>
<b>Szanse (O):</b> <ul style="list-style-type: none"><li>- ograniczanie ruchu samochodowego w centrum miasta</li><li>- popularyzacja ekologicznego podejścia i chęć porzucania samochodów na rzecz transportu publicznego</li></ul>	<b>Zagrożenia (T):</b> <ul style="list-style-type: none"><li>- praca zdalna zmniejszająca zapotrzebowanie na transport osób</li><li>- chęć bycia niezależnym przez jednostki społecznie, więc mniejsza skłonność do transportu publicznego</li></ul>

## **5 Specyfikacja wymagań**

### **5.1 Charakterystyka ogólna**

#### **5.1.1 Definicja produktu**

Jest to system wspomagający pracę kierowcy autobusu linii numer 75 w Szczecinie.

#### **5.1.2 Podstawowe założenia**

System ma za zadanie wspomagać pracę kierowcy ułatwiając jego codzienną pracę. Będzie on służył do wyszukiwania objazdu w przypadku napotkania na korki lub wypadki, co zwiększy prędkość komunikacyjną linii. Kolejną funkcjonalnością będzie informowanie pasażerów o opóźnieniach w czasie rzeczywistym. Zapewni on szybki kontakt z koordynatorem w celu reorganizacji trasy lub pomocy technicznej oraz w sytuacji zagrożenia ułatwi wezwanie odpowiednich służb bezpieczeństwa. Dodatkowo system ułatwi pracę kierowcy, który w przejrzysty sposób będzie mógł zarządzać czasem swojej pracy oraz przerwami.

#### **5.1.3 Cel biznesowy**

Organizacja docelowa poprzez wdrożenie naszego systemu, chciałaby ułatwić codzienną pracę kierowcy autobusu, koordynatora ruchu oraz uzupełnić dotychczas istniejące rozwiązania w zakresie informacji pasażerskiej co przełożyłoby się bezpośrednio na polepszenie poziomu zadowolenia podróżnych ZDiTM ze świadczonych przez nich usług.

#### **5.1.4 Użytkownicy**

- [Kierowca autobusu](#)
- [Koordynator ruchu](#)
- [Pasażer](#)
- [Służby bezpieczeństwa](#)
- [Kierownik](#)
- [Zespół techniczny](#)

#### **5.1.5 Korzyści z systemu**

##### **5.1.5.1 Kierowca autobusu**

Kierowca autobusu uzyska możliwość wglądu w mapę trasy przejazdu w czasie rzeczywistym z informacjami o korkach i wypadkach oraz możliwych objazdach. Oprócz tego uzyska on możliwość szybkiego kontaktu ze służbami bezpieczeństwa, zarządzania swoim czasem pracy oraz przerwami, szybkiego kontaktu z zespołem technicznym w przypadku awarii lub usterki oraz otrzyma dostęp do informacji dotyczących stanu technicznego pojazdu, który obsługuje. Dodatkowo kierowca autobusu zostanie powiadomiony o przewidywanym czasie przyjazdu na kolejny przystanek.

#### **5.1.5.2 Koordynator ruchu**

Koordynator ruchu uzyska możliwość szybkiego kontaktu z kierowcą autobusu, możliwość zarządzania swoim czasem pracy i przerwami, otrzyma dostęp do stanu technicznego wszystkich pojazdów oraz możliwość nadpisywania, dodawania nowych oraz usuwania wszelkich danych technicznych w systemie, a także uzyska on dostęp do wszystkich przechowywanych statystyk dotyczących zapewnienia pojazdów. Oprócz tego zostanie on powiadomiony o próbie kontaktu ze służbami bezpieczeństwa oraz o wystąpieniu awarii lub usterki, a w obu z tych przypadków otrzyma informacje o lokalizacji pojazdu. Dodatkowo koordynator ruchu zostanie powiadomiony o zmianie trasy oraz opóźnieniu.

#### **5.1.5.3 Pasażer**

Pasażer uzyska możliwość dokładniejszej informacji o rzeczywistym czasie kursowania linii oraz możliwość kupna biletów na przejazd przez zewnętrzny system. Oprócz tego uzyska on podgląd możliwych przesiadek z wybranego kursu na inny środek transportu oraz aktualny podgląd zapewnienia autobusów na linii. Dodatkowo pasażer zostanie powiadomiony o zmianie trasy lub występującym opóźnieniu.

#### **5.1.5.4 Służby bezpieczeństwa**

Służby bezpieczeństwa uzyskają dostęp do dokładnej lokalizacji pojazdu w czasie zdarzenia oraz wgląd do kamer w celu uzyskania lepszego obrazu danej sytuacji.

#### **5.1.5.5 Kierownik**

Kierownik zyska dokładny wgląd w czas pracy swoich pracowników, dostęp do wszystkich przechowywanych statystyk dotyczących zapewnienia pojazdów oraz będzie informowany o najważniejszych wydarzeniach jak wezwanie służb bezpieczeństwa.

#### **5.1.5.6 Zespół techniczny**

Zespół techniczny uzyska system informacji o zgłoszeniach razem z dokładną lokalizacją pojazdu oraz dostęp do stanu technicznego wszystkich pojazdów wraz z możliwością nadpisywania istniejących już danych.

### **5.1.6 Ograniczenia projektowe i wdrożeniowe**

- **Aktualizacje systemów i bibliotek** - dostosowywanie oprogramowania do zmieniających się wersji używanych narzędzi wymaga dodatkowych nakładów pracy i może wprowadzać regresję.
- **RODO** - utrzymywanie zgodności z rozporządzeniami dotyczącymi przetwarzania danych osobowych wymaga kontaktów z radcą prawnym.
- **Wieloplatfomowość** - projektowanie funkcjonalności możliwych do zrealizowania zarówno na wielu wersjach systemu Android może ograniczyć możliwości i zwiększyć koszty wytwarzania oprogramowania.
- **Integralność** - zintegrowanie aplikacji z już istniejącym oprogramowaniem ZDiTM wymagać może dostosowania do specyficznych standardów i protokołów.

## **5.2 Wymagania funkcjonalne**

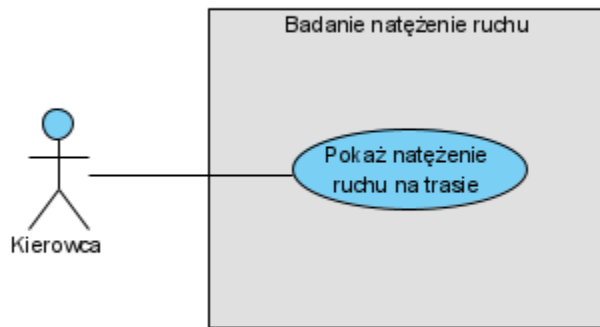
### **5.2.1 Lista wymagań**

1. System powinien pokazywać na bieżąco natężenia ruchu drogowego na trasie linii.
2. System powinien proponować najlepsze możliwe objazdy w przypadku korków lub wypadków na trasie.
3. System powinien informować o opóźnieniach w kursowaniu.
4. System powinien umożliwiać zarządzanie czasem pracy.
5. System powinien umożliwiać szybki kontakt ze służbami bezpieczeństwa.
6. System powinien umożliwiać kontakt z zespołem technicznym w przypadku awarii lub usterki.
7. System powinien umożliwiać sprzedaż biletów.
8. System powinien pokazywać informacje techniczne pojazdu (m.in.: numer rejestracyjny, ważność przeglądu, dowód rejestracyjny pojazdu, numer polisy).
9. System powinien pokazywać statystyki o zapelnieniu pojazdu.
10. System powinien pokazywać możliwość przesiadki na inny autobus, tramwaj lub pociąg.
11. System powinien umożliwiać modyfikowanie danych technicznych pojazdu.

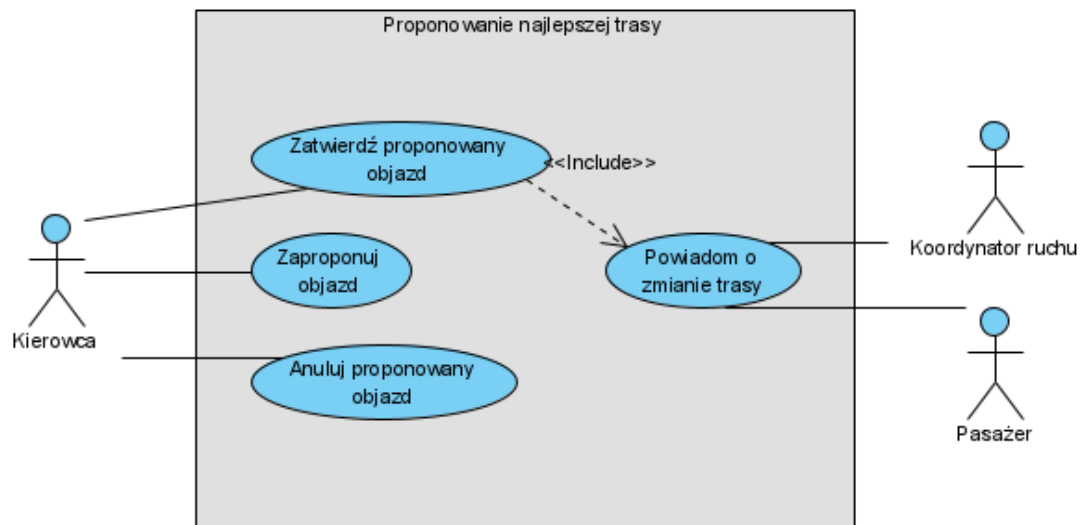


### 5.2.2 Diagramy przypadków użycia

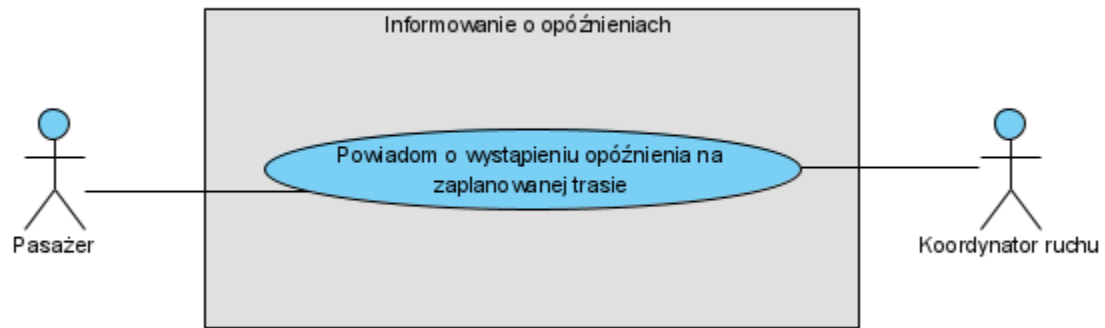
1. System powinien pokazywać na bieżąco natężenia ruchu drogowego na trasie linii.



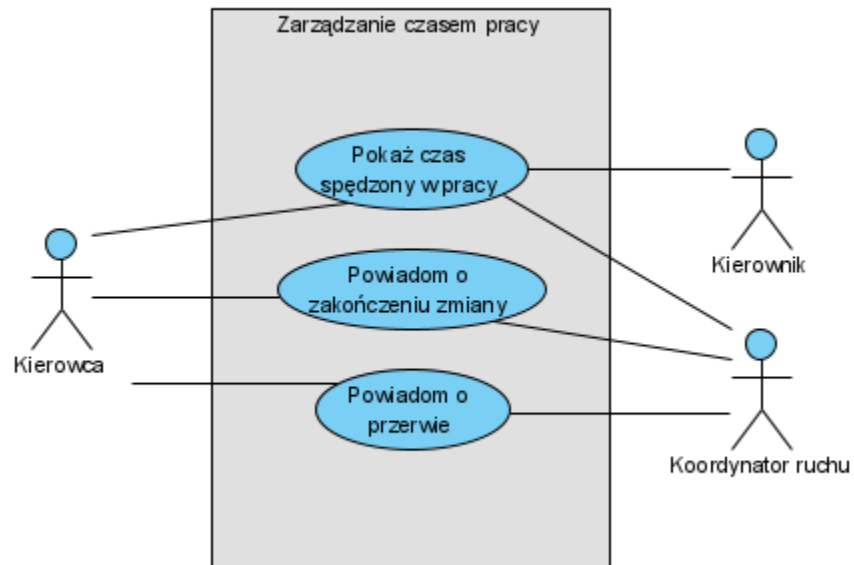
2. System powinien proponować najlepsze możliwe objazdy w przypadku korków lub wypadków na trasie.



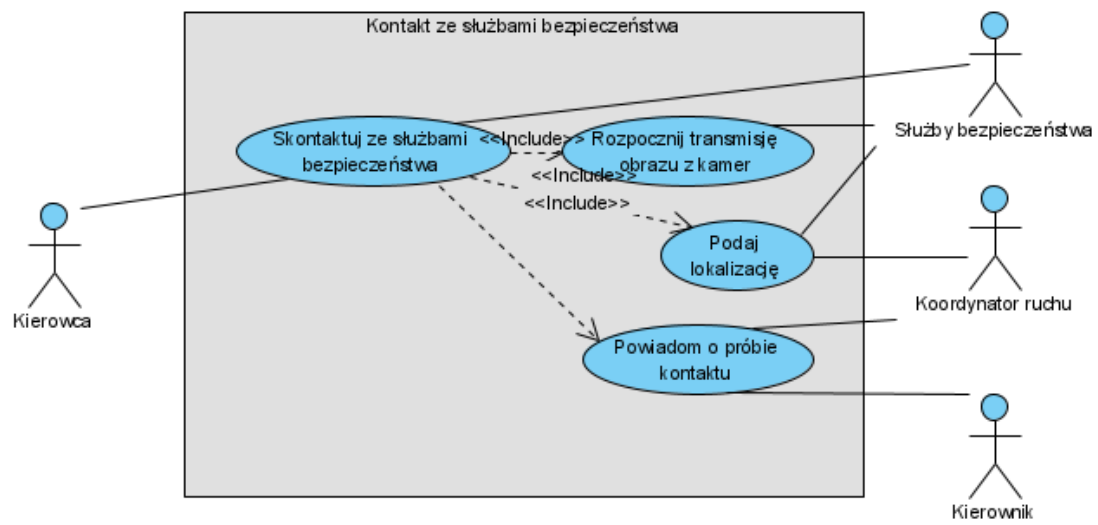
3. System powinien informować o opóźnieniach w kursowaniu.



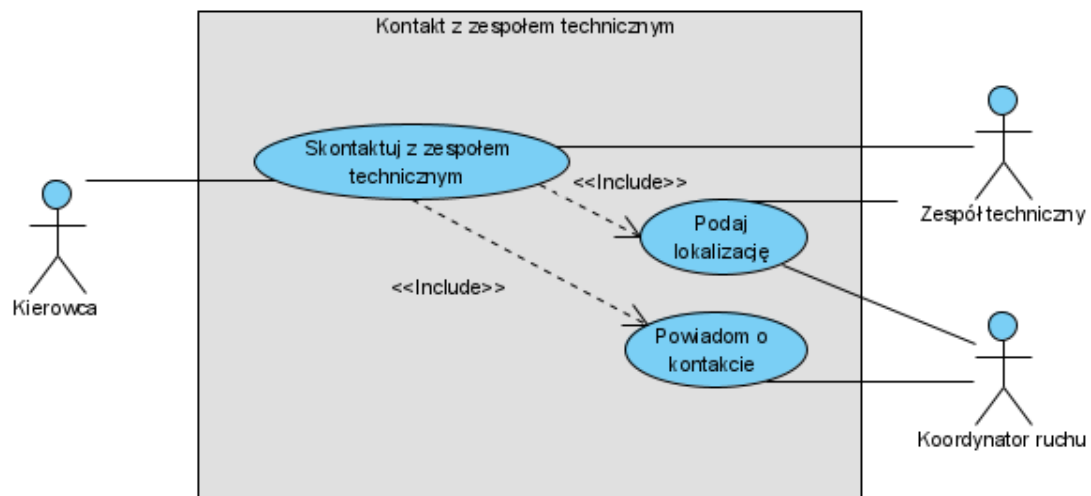
4. System powinien umożliwiać zarządzanie czasem pracy.



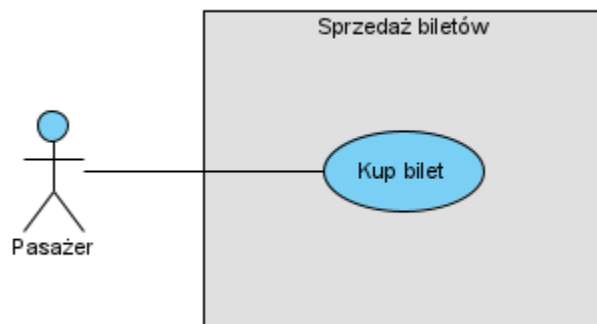
5. System powinien umożliwiać szybki kontakt ze służbami bezpieczeństwa.



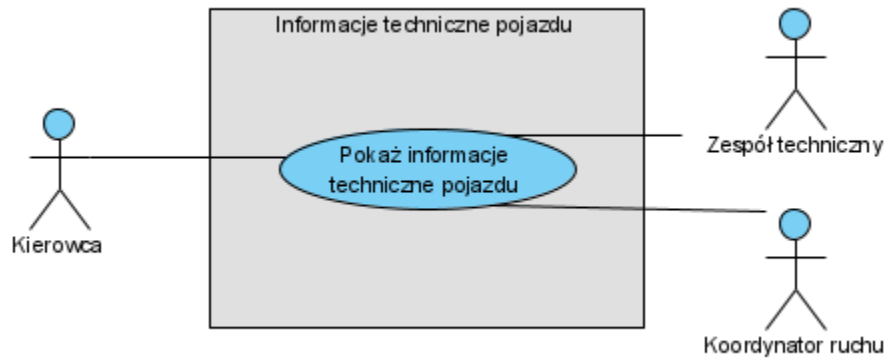
6. System powinien umożliwiać kontakt z zespołem technicznym w przypadku awarii lub usterki.



7. System powinien umożliwiać sprzedaż biletów.

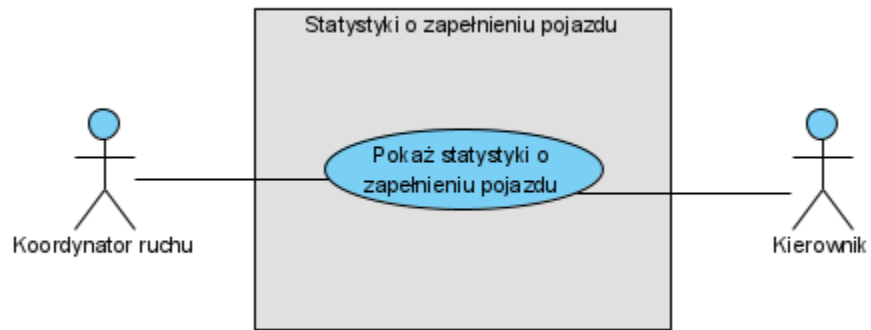


8. System powinien pokazywać informacje techniczne pojazdu.

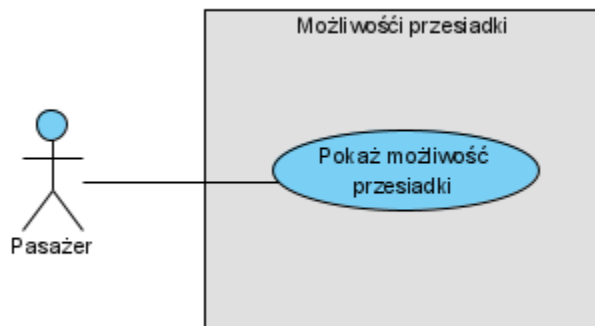




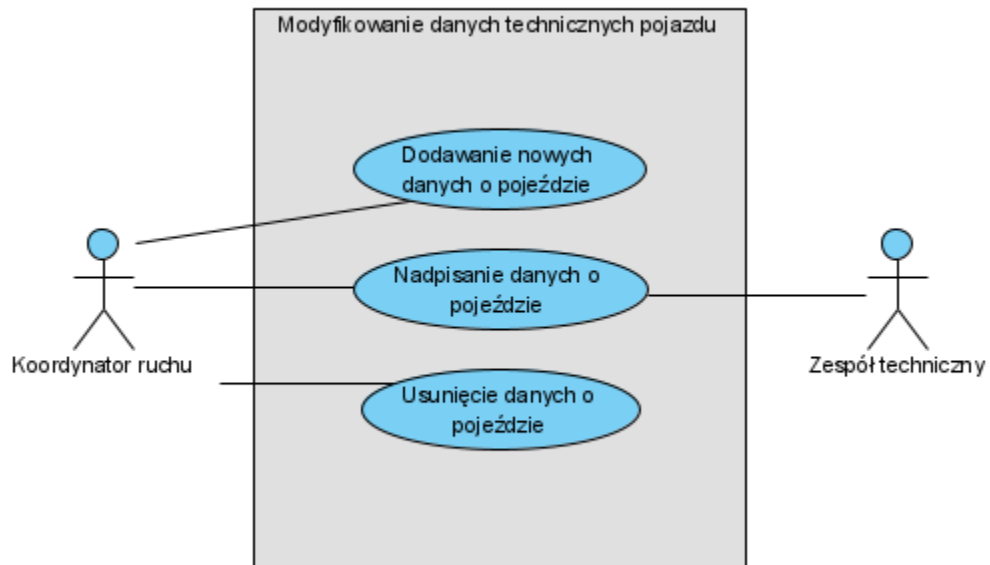
9. System powinien pokazywać statystyki o zapelnieniu pojazdu.



10. System powinien pokazywać możliwość przesiadki na inny autobus, tramwaj lub pociąg.



11. System powinien umożliwiać modyfikowanie danych technicznych pojazdu.



### 5.2.3 Szczegółowy opis wymagań

#### 1. System powinien pokazywać na bieżąco natężenia ruchu drogowego na trasie linii

##### Uzasadnienie biznesowe:

- 5.1.5.1 - Kierowca autobusu uzyska możliwość podglądu natężenia ruchu w czasie rzeczywistym.

##### Użytkownicy:

- Kierowca

##### Scenariusz główny:

- **Warunki początkowe:**
  - Uruchomione oprogramowanie
  - Dostęp do internetu
- **Przebieg działań:**
  1. Użytkownik loguje się do oprogramowania na profil kierowcy.
  2. System autentykuje użytkownika.
  3. Użytkownik przechodzi z głównego ekranu na widok natężenia ruchu.
  4. System wyświetla natężenie ruchu w czasie rzeczywistym, pobrane przez internet.
- **Efekt:**
  - Oprogramowanie wyświetla natężenie ruchu w okolicy widoczne dla użytkownika.
- **Wymagania нефunkcjonalne:**
  - Mapa natężenia ruchu będzie odświeżana co 10s.
  - Interfejs jest zgodny z Material Design.
  - Szkolenie dla użytkowników.
- **Częstotliwość:** 2
- **Istotność:** 4

## **2. System powinien proponować najlepsze możliwe objazdy w przypadku korków lub wypadków na trasie**

### **Uzasadnienie biznesowe:**

- 5.1.5.1 - Kierowca autobusu uzyska możliwość podglądu najlepszego dopasowane przez system objazdu.
- 5.1.5.3 - Pasażer uzyska powiadomienie o zmianie trasy.
- 5.1.5.2 - Koordynator ruchu zostanie powiadomiony o zmianie trasy.

### **Użytkownicy:**

- Kierowca
- Pasażer
- Koordynator ruchu

### **Scenariusz główny:**

- **Warunki początkowe:**
  - Uruchomione oprogramowanie
  - Dostęp do internetu
- **Przebieg działań:**
  1. Użytkownik loguje się do oprogramowania na odpowiadający mu profil.
  2. System autentykuje użytkownika.
  3. Użytkownik przechodzi z głównego ekranu na widok natężenia ruchu.
  4. System wyświetla natężenie ruchu w czasie rzeczywistym, pobrane przez internet.
  5. System analizuje trasę przejazdu i szuka najlepszego rozwiązania.
  6. System wyświetla kierowcy propozycję najlepszego przejazdu.
- Scenariusz alternatywny - odrzucenie propozycji:
  - 1-6. Jak w scenariuszu głównym.
  7. Kierowca odrzuca propozycję objazdu.
- Scenariusz alternatywny - akceptacja propozycji:
  - 1-6. Jak w scenariuszu głównym.
  7. Kierowca akceptuje propozycję przejazdu
  8. System aktualizuje trasę przejazdu na mapie oraz czas przyjazdu na poszczególne przystanki.
  9. Użytkownicy otrzymują powiadomienie o zmianie trasy przejazdu autobusu.
- **Efekt:**
  - System proponuje najlepszy możliwy objazd dla kierowcy.
  - System informuje pasażerów oraz koordynatora ruchu o zmianie trasy przejazdu autobusu.
- **Wymagania нефunkcjonalne:**
  - Mapa natężenia ruchu będzie odświeżana co 10s.
  - System będzie analizował trasę co 10s.
  - Interfejs jest zgodny z Material Design.

- Szkolenie dla użytkowników.
- Powiadomienia muszą dotrzeć do użytkowników w czasie nie dłuższym niż 30s.

- **Częstotliwość: 3**
- **Istotność: 3**

### 3. System powinien informować o opóźnieniach w kursowaniu

#### Uzasadnienie biznesowe:

- 5.1.5.3 - Pasażer zostanie powiadomiony o opóźnieniu.
- 5.1.5.2 - Koordynator ruchu zostanie powiadomiony o opóźnieniu.

#### Użytkownicy:

- Pasażer
- Koordynator ruchu

#### Scenariusz główny:

- **Warunki początkowe:**
  - Uruchomione oprogramowanie
  - Dostęp do internetu
- **Przebieg działań:**
  1. Użytkownik loguje się do oprogramowania na odpowiadający mu profil.
  2. System autentykuje użytkownika.
  3. Użytkownik przechodzi z głównego ekranu na widok zaplanowanej trasy.
  4. Użytkownik wybiera interesujący go kurs z dostępnej listy.
  5. System wyświetla szacowany czas przyjazdu wybranego kursu na poszczególne przystanki wraz z informacją o opóźnieniach w czasie rzeczywistym, pobrane przez internet.
- Scenariusz alternatywny - obserwowanie wybranego kursu:
  - 1-3. Jak w scenariuszu głównym.
  4. Użytkownik oznacza gwiazdką interesujący go kurs z dostępnej listy.
  5. Użytkownik wychodzi z aplikacji na ekran główny urządzenia.
  6. System wysyła powiadomienie o wystąpieniu opóźnienia.
- **Efekt:**
  - System informuje użytkowników o opóźnieniach w zaplanowanym czasie dojazdu.
- **Wymagania niefunkcjonalne:**
  - Informację będą odświeżane co 15s.
  - Powiadomienie zostanie wysłane jeśli opóźnienie będzie większe lub równe 120s.
  - Interfejs jest zgodny z Material Design.
  - Szkolenie dla użytkowników.
- **Częstotliwość:** 4
- **Istotność:** 4

#### **4. System powinien umożliwiać zarządzanie czasem pracy**

##### **Uzasadnienie biznesowe:**

- 5.1.5.1 - Kierowca autobusu uzyska możliwość zarządzania swoim czasem pracy oraz przerwami.
- 5.1.5.5 - Kierownik uzyska wgląd do czasu pracy wszystkich pracowników.
- 5.1.5.2 - Koordynator ruchu uzyska możliwość zarządzania swoim czasem pracy oraz przerwami.

##### **Użytkownicy:**

- Kierowca
- Kierownik
- Koordynator ruchu

##### **Scenariusz główny:**

- **Warunki początkowe:**
  - Uruchomione oprogramowanie
  - Dostęp do internetu
- **Przebieg działań:**
  1. Użytkownik loguje się do oprogramowania na odpowiadający mu profil.
  2. System autentykuje użytkownika.
  3. Użytkownik przechodzi z głównego ekranu na widok zarządzania czasem pracy
  4. Użytkownik widzi dane o swoim czasie pracy lub wszystkich pracownikach zależnie od polityki związanej z profilem.
- **Scenariusz alternatywny - edycja danych:**
  - 1-4. Jak w scenariuszu głównym.
  5. Użytkownik wybiera opcję "Edytuj dane".
  6. System przenosi użytkownika do widoku edycji danych związanych z czasem pracy.
  7. Użytkownik uzupełnia dane o czasie pracy.
- **Efekt:**
  - System przedstawia Kierowcy i Koordynatorowi ruchu informacje o czasie spędzonym w pracy.
  - System powiadamia Kierowcę i Koordynatora ruchu o zakończeniu zmiany.
  - System powiadamia Kierowcę i Koordynatora ruchu o przerwie w pracy.
- **Wymagania нефункционалне:**
  - Informację będą odświeżane co 15s.
  - Interfejs jest zgodny z Material Design.
  - Szkolenie dla użytkowników.
- **Częstotliwość:** 1
- **Istotność:** 3



## **5. System powinien umożliwiać szybki kontakt ze służbami bezpieczeństwa**

### **Uzasadnienie biznesowe:**

- 5.1.5.1 - Kierowca autobusu uzyska możliwość szybkiego kontaktu ze służbami bezpieczeństwa.
- 5.1.5.2 - Koordynator ruchu zostanie powiadomiony o próbie kontaktu ze służbami bezpieczeństwa oraz otrzyma informacje o lokalizacji pojazdu.
- 5.1.5.4 - Służby bezpieczeństwa zostaną połączone z kierowcą pojazdu, otrzymają one informację o lokalizacji pojazdu oraz dostęp do kamer znajdujących się wewnątrz autobusu.
- 5.1.5.5 - Kierownik otrzyma powiadomienie o próbie kontaktu ze służbami bezpieczeństwa.

### **Użytkownicy:**

- Kierowca
- Służby bezpieczeństwa
- Kierownik
- Koordynator ruchu

### **Scenariusz główny:**

- **Warunki początkowe:**
  - Uruchomione oprogramowanie
  - Dostęp do internetu
- **Przebieg działań:**
  1. Użytkownik loguje się na odpowiadające mu konto.
  2. System autentykuje użytkownika.
  3. Użytkownik używa przycisku szybkiego kontaktu ze służbami bezpieczeństwa.
  4. System wysyła informację o próbie kontaktu do koordynatora ruchu i kierownika.
  5. System łączy kierowcę z odpowiednimi służbami.
  6. System wysyła aktualną lokalizację pojazdu do koordynatora ruchu i służb bezpieczeństwa.
- **Efekt:**
  - System łączy kierowcę z odpowiednimi służbami
  - Kierownik oraz koordynator ruchu są poinformowani o próbie kontaktu.
  - Koordynator ruchu oraz służby są poinformowani o aktualnej lokalizacji pojazdu.
  - Służby otrzymują dostęp do kamer monitoringu znajdujących się wewnątrz autobusu.
- **Wymagania нефunkcjonalne:**
  - System umożliwia kontakt w czasie natychmiastowym.
  - System przesyła lokalizację pojazdu z dokładnością do 20m.
  - Interfejs jest zgodny z Material Design.

- Szkolenie dla użytkowników.
- **Częstotliwość:** 1
- **Istotność:** 5

## **6. System powinien umożliwiać kontakt z zespołem technicznym w przypadku awarii lub usterki**

### **Uzasadnienie biznesowe:**

- 5.1.5.1 - Kierowca autobusu uzyska możliwość szybkiego kontaktu z zespołem technicznym w przypadku awarii lub usterki.
- 5.1.5.6 - Zespół techniczny uzyska system informacji o zgłoszeniach razem z dokładną lokalizacją pojazdu.
- 5.1.5.2 - Koordynator ruchu zostanie poinformowany o wystąpieniu awarii oraz miejscu jej wystąpienia na mapie.

### **Użytkownicy:**

- Kierowca
- Zespół techniczny
- Koordynator ruchu

### **Scenariusz główny:**

- **Warunki początkowe:**
  - Uruchomione oprogramowanie
  - Dostęp do internetu
- **Przebieg działań:**
  1. Użytkownik loguje się na odpowiadające mu konto.
  2. System autentykuje użytkownika.
  3. Użytkownik używa przycisku szybkiego kontaktu z zespołem technicznym.
  4. System wysyła informację o próbie kontaktu do koordynatora ruchu.
  5. System łączy kierowcę z zespołem technicznym.
  6. System przesyła aktualną lokalizację pojazdu do koordynatora ruchu i zespołu technicznego.
- **Efekt:**
  - System łączy kierowcę z zespołem technicznym
  - Koordynator ruchu zostaje poinformowany o próbie kontaktu.
  - Koordynator ruchu oraz zespół techniczny są poinformowani o aktualnej lokalizacji pojazdu.
- **Wymagania нефunkcjonalne:**
  - System umożliwia kontakt w czasie natychmiastowym.
  - System przesyła lokalizację pojazdu z dokładnością do 20m.
  - Interfejs jest zgodny z Material Design.
  - Szkolenie dla użytkowników.
- **Częstotliwość:** 1
- **Istotność:** 4

## **7. System powinien umożliwiać sprzedaż biletów**

### **Uzasadnienie biznesowe:**

- 5.1.5.3 - Pasażer uzyska możliwość kupowania biletów.

### **Użytkownicy:**

- Pasażer

### **Scenariusz główny:**

- **Warunki początkowe:**
  - Uruchomione oprogramowanie
  - Dostęp do internetu
- **Przebieg działań:**
  1. Pasażer przechodzi z ekranu głównego do widoku planowania trasy.
  2. Pasażer naciska opcję “Kup bilet” przy wybranym kursie.
  3. System przenosi użytkownika do zewnętrznej aplikacji, która zapewnia sprzedaż biletów.
- **Efekt:**
  - Pasażer ma zakupiony bilet.
  - Pasażer ma podgląd biletu w celu okazania go kontrolerowi biletów.
- **Wymagania нефunkcjonalne:**
  - System natychmiast przenosi użytkownika do zewnętrznej aplikacji.
  - Interfejs jest zgodny z Material Design.
- **Częstotliwość:** 2
- **Istotność:** 3

## **8. System powinien pokazywać informacje techniczne pojazdu**

### **Uzasadnienie biznesowe:**

- 5.1.5.1 - Kierowca autobusu otrzyma informacje dotyczące stanu technicznego pojazdu, który obsługuje.
- 5.1.5.6 - Zespół techniczny otrzyma dostęp do stanu technicznego wszystkich pojazdów.
- 5.1.5.2 - Koordynator ruchu otrzyma dostęp do stanu technicznego wszystkich pojazdów.

### **Użytkownicy:**

- Kierowca
- Zespół techniczny
- Koordynator ruchu

### **Scenariusz główny:**

- **Warunki początkowe:**
  - Uruchomione oprogramowanie
  - Dostęp do internetu
- **Przebieg działań:**
  1. Użytkownik loguje się na odpowiadające mu konto.
  2. System autentykuje użytkownika.
  3. Użytkownik przechodzi do widoku informacji technicznych
  4. Użytkownik wybiera interesujący go pojazd z dostępnej listy
  5. Użytkownik wybiera datę z dostępnej listy, z której mają pochodzić dane.
  6. System wyświetla tabele z danymi
- **Efekt:**
  - Użytkownik widzi dane techniczne wybranego przez siebie pojazdu, aktualne dla wybranego dnia.
- **Wymagania niefunkcjonalne:**
  - Informacje będą odświeżane raz dziennie, po wprowadzeniu aktualizacji danych przez koordynatora.
  - Interfejs jest zgodny z Material Design.
  - Szkolenie dla użytkowników.
- **Częstotliwość: 1**
- **Istotność: 5**

## 9. System powinien pokazywać statystyki zapelnienia pojazdu

### Uzasadnienie biznesowe:

- 5.1.5.5 - Kierownik uzyska dostęp do wszystkich przechowywanych statystyk dotyczących zapelnienia pojazdów.
- 5.1.5.2 - Koordynator ruchu uzyska dostęp do wszystkich przechowywanych statystyk dotyczących zapelnienia pojazdów.

### Użytkownicy:

- Kierownik
- Koordynator ruchu

### Scenariusz główny:

- **Warunki początkowe:**
  - Uruchomione oprogramowanie
  - Dostęp do internetu
- **Przebieg działań:**
  1. Użytkownik loguje się na odpowiadające mu konto.
  2. System autentykuje użytkownika.
  3. Użytkownik przechodzi do widoku statystyk zapelnienia pojazdu.
  4. Użytkownik wybiera interesujący go pojazd z dostępnej listy.
  5. Użytkownik wybiera datę z dostępnej listy, z której mają pochodzić dane.
  6. System wyświetla dane.
- **Efekt:**
  - Użytkownik widzi statystyki zapelnienia wybranego przez siebie pojazdu, aktualne dla wybranego dnia.
- **Wymagania нефunkcjonalne:**
  - Statystyki będą odświeżane co odjazd autobusu z przystanku
  - Interfejs jest zgodny z Material Design.
  - Szkolenie dla Kierownika i Koordynatora ruchu dotyczące analizy statystyk.
- **Częstotliwość:** 2
- **Istotność:** 2

## **10. System powinien pokazywać możliwość przesiadki na inny autobus, tramwaj lub pociąg**

### **Uzasadnienie biznesowe:**

- 5.1.5.3 - Pasażer uzyska podgląd możliwych przesiadek z wybranego kursu na inny środek transportu.

### **Użytkownicy:**

- Pasażer

### **Scenariusz główny:**

- **Warunki początkowe:**
  - Uruchomione oprogramowanie
  - Dostęp do internetu
- **Przebieg działań:**
  1. Pasażer przechodzi z ekranu głównego do widoku możliwych przesiadek.
  2. Pasażer wybiera interesujący go kurs z filtrowanej listy.
  3. System wyświetla listę wszystkich kolejnych przystanków na trasie wraz z 3 najszybciej odjeżdżającym z niego środkami transportu po przewidywanym przyjeździe wybranego wcześniej kursu.
- **Scenariusz alternatywny:**
  - 1-3. Jak w scenariuszu głównym.
  4. Pasażer naciska na konkretny przystanek.
  5. System wyświetla chronologiczną przewijaną listę wszystkich kolejnych odjazdów z wybranego przystanku do końca dnia.
- **Efekt:**
  - Pasażer otrzymuje propozycje możliwych przesiadek z wybranego kursu.
- **Wymagania нефunkcjonalne:**
  - Informacje będą odświeżane co 15s.
  - Interfejs jest zgodny z Material Design.
- **Częstotliwość:** 3
- **Istotność:** 3

## **11. System powinien umożliwiać modyfikowanie danych technicznych pojazdu.**

### **Uzasadnienie biznesowe:**

- 5.1.5.6 - Zespół techniczny uzyska możliwość nadpisywania istniejących danych technicznych pojazdów w systemie.
- 5.1.5.2 - Koordynator ruchu uzyska możliwość nadpisywania, dodawania nowych oraz usuwania wszelkich danych technicznych pojazdów w systemie.

### **Użytkownicy:**

- Zespół techniczny
- Koordynator ruchu

### **Scenariusz główny:**

- **Warunki początkowe:**
  - Uruchomione oprogramowanie
  - Dostęp do internetu
- **Przebieg działań:**
  1. Użytkownik loguje się na odpowiadające mu konto.
  2. System autentykuje użytkownika.
  3. Użytkownik przechodzi do widoku informacji technicznych.
  4. Użytkownik wybiera opcję "Edytuj dane".
  5. Użytkownik wybiera interesujący go pojazd z dostępnej listy.
  6. System przenosi użytkownika do ekranu edycji danych technicznych wybranego pojazdu.
  7. Użytkownik modyfikuje dane zgodnie z polityką związaną z jego profilem.
- **Efekt:**
  - Użytkownik zmodyfikował dane techniczne wybranego przez siebie pojazdu.
- **Wymagania нефunkcjonalne:**
  - Interfejs jest zgodny z Material Design.
  - Szkolenie dla użytkowników.
- **Częstotliwość:** 1
- **Istotność:** 4



### **5.3 Wymagania niefunkcjonalne**

#### **1. Wydajność**

- Aplikacja nie może ważyć więcej niż 256 MB, nie wliczając danych lokalnych użytkownika.
- Aplikacja będzie odświeżała mapę natężenia ruchu co 10s.
- Aplikacja będzie analizowała trasę przejazdu co 10s.
- Powiadomienia z aplikacji muszą dotrzeć do użytkowników w czasie nie dłuższym niż 30s.
- Aplikacja będzie odświeżała informacje o czasie przyjazdu co 15s.

#### **2. Bezpieczeństwo**

- Aplikacja daje możliwość dezaktywacji konta użytkownika przez administratora w przypadku zgubienia urządzenia.
- Serwer z danymi jest zabezpieczony przed utratą danych dzięki RAID.

#### **3. Zabezpieczenia**

- Dane użytkowników będą szyfrowane za pomocą SHA-3.
- Aplikacja będzie używała Proguard dla systemu Android oraz iXGuard dla systemu iOS do obfuskacji kodu.
- Możliwość skorzystania z aplikacji jedynie po autentykacji (nie dotyczy trybu korzystania z produktu jako pasażer).

#### **4. Adaptowalność / Elastyczność**

- Aplikacja będzie dostosowywana do najnowszych wersji systemów operacyjnych oraz używanych bibliotek w czasie do 3 miesięcy.
- Aplikacja będzie wspierała 5 ostatnich pełnych wersji systemów operacyjnych.
- Aplikacja będzie dostosowywana do najnowszych wytycznych dotyczących designu maksymalnie w przeciągu roku.

#### **5. Dostępność**

- Aplikacja w wersji dla podróżnych będzie dostępna dla każdego w Google Play oraz App Store.
- Aplikacja będzie dostępna dla użytkowników 24/7/365 przynajmniej przez 99.7% czasu.

#### **6. Poprawność / Awaryjność**

- Do kolejnej wersji aplikacji może przejść co najwyżej 8% błędów z poprzedniej wersji.

#### **7. Łatwość konserwacji**

- Aplikacja będzie wspierana przez deweloperów przez cały okres jej istnienia.
- Architektura aplikacji musi umożliwiać łatwą skalowalność.
- Wszyscy użytkownicy będą mieli możliwość zgłaszania błędów oraz opinii o aplikacji.
- Wszystkie błędy będą zarządzane w systemie obsługi defektów.

#### **8. Przenośność**

- Aplikacja będzie dostępna na systemy Android oraz iOS jako osobno napisane produkty bez możliwości użycia i przeniesienia na inne systemy.

- Aplikacja będzie przechowywała dane na zewnętrznym serwerze z możliwością odczytu z dowolnego autoryzowanego środowiska.

#### **9. Testowalność**

- Wszystkie części aplikacji muszą mieć napisane testy jednostkowe.
- Aplikacja będzie zawierała specjalny ukryty ekran dla testerów z opcjami ułatwiającymi testowanie.
- Wszystkie historyjki użytkownika muszą mieć napisane przypadki testowe.

#### **10. Użyteczność**

- Aplikacja w wersji na system iOS będzie zgodna z Human Interface Guidelines.
- Aplikacja w wersji na system Android będzie zgodna z Material Design.
- Aplikacja będzie zawierała podstawowy tutorial przy pierwszym uruchomieniu.
- Aplikacja będzie pokazywała użytkownikowi ekran informujący o nowościach przy wprowadzeniu nowej wersji.

## 6 Zarządzanie projektem

### 6.1 Zasoby ludzkie

1. [Product Owner](#)
2. [UX Designer](#)
3. [Project Manager](#)
4. [Software Engineer](#) × 6
5. [QA Engineer](#) × 2

### 6.2 Harmonogram prac

Etapy	Zadania \ Tygodnie	1 - 4	5 - 8	9 - 10	11 - 22	23	24 - 34	35 - 36	37	38 - 40	41 - 50	51	52
Doprecyzowanie projektu oprogramowania	Uzupełnienie dokumentacji												
	Poprawienie wersji koncepcyjnej												
Rozwój wersji alfa	Wytwarzanie oprogramowania												
	Testowanie oprogramowania												
	Wytworzenie wersji alfa												
Rozwój wersji beta	Wytwarzanie oprogramowania												
	Testowanie oprogramowania												
	Wytworzenie wersji beta												
Przeszkolenie pracowników ZDiTM													
	Zbieranie informacji zwrotnych od beta testerów												
Rozwój wersji stabilnej	Dostosowywanie oprogramowania												
	Testowanie oprogramowania												
	Wydanie wersji stabilnej - 1.0.0												
Zainstalowanie i skonfigurowanie systemu dla pracowników ZDiTM													

### 6.3 Etapy/kamienie milowe projektu

1. Podpisanie umowy
2. Przygotowanie projektu oprogramowania
3. Wytworzenie wersji alfa
4. Wytworzenie wersji beta
5. Przeszkolenie użytkowników
6. Wydanie wersji stabilnej
7. Instalacja i konfiguracja systemu

## 7 Zarządzanie ryzykiem

### 7.1 Lista czynników ryzyka

1. Konkurencja w postaci innej aplikacji zapewniającej podobne funkcjonalności.
2. Rezygnacja klienta z subskrypcji systemu.
3. Brak chęci korzystania z aplikacji przez użytkowników.
4. Zakończenie współpracy z jedną z firm dostarczających usługi potrzebne funkcjonowaniu aplikacji.
5. Zmiana przepisów regulujących przetwarzanie danych.
6. Zmniejszenie popytu na transport publiczny.
7. Niewłaściwa kontrola nad pracownikami i ich nieetycznymi zachowaniami.
8. Awaria systemu.
9. Utrata finansowania projektu.

### 7.2 Ocena ryzyka

1. **Konkurencja w postaci innej aplikacji zapewniającej podobne funkcjonalności:**
  - Prawdopodobieństwo: 50%
  - Wpływ: Konsekwencją konkurencji na rynku może być potrzeba obniżenia ceny subskrypcji systemu oraz podwyższenie jakości świadczonych usług, aby wyróżnić się na tle innych aplikacji.
2. **Rezygnacja klienta z subskrypcji systemu:**
  - Prawdopodobieństwo: 5%
  - Wpływ: Całkowita utrata zysków oraz konieczność pozyskania nowych klientów.
3. **Brak chęci korzystania z aplikacji przez użytkowników:**
  - Prawdopodobieństwo: 10%
  - Wpływ: Zmniejszone wpływy oraz potrzeba zmian systemu, aby dostosować ją do potrzeb użytkowników.
4. **Zakończenie współpracy z jedną z firm dostarczających usługi potrzebne funkcjonowaniu aplikacji:**
  - Prawdopodobieństwo: 5%
  - Wpływ: Utrata jednej z funkcjonalności systemu.
5. **Zmiana przepisów regulujących przetwarzanie danych:**
  - Prawdopodobieństwo: 35%
  - Wpływ: Konieczność zmian polityki aplikacji oraz dostosowanie powiązanych funkcjonalności.
6. **Zmniejszenie popytu na transport publiczny:**
  - Prawdopodobieństwo: 75%

- Wpływ: Zmniejszone zapotrzebowanie na system, czego konsekwencją może być ograniczenie dochodów.

**7. Niewłaściwa kontrola nad pracownikami i ich nieetycznymi zachowaniami:**

- Prawdopodobieństwo: 15%
- Wpływ: Zmniejszone tempo prac, obniżona jakość oprogramowania lub możliwy wyciek poufnych informacji.

**8. Awaria systemu:**

- Prawdopodobieństwo: 25%
- Wpływ: Ograniczenie zaufania klienta oraz użytkowników wobec systemu.

**9. Utrata finansowania projektu:**

- Prawdopodobieństwo: 5%
- Wpływ: Możliwe zatrzymanie projektu w fazie planowania oraz potrzeba poszukiwań nowych źródeł finansowania.

### **7.3 Plan reakcji na ryzyko**

**1. Konkurencja w postaci innej aplikacji zapewniającej podobne funkcjonalności:**

- przeznaczenie większych funduszy na marketing i reklamę
- dostarczenie nowych funkcjonalności wyróżniających się od konkurencji

**2. Rezygnacja klienta z subskrypcji systemu:**

- pozyskanie nowego klienta
- renegotiacja warunków umowy

**3. Brak chęci korzystania z aplikacji przez użytkowników:**

- poprawa marketingu oraz reklam
- dodanie nowych funkcjonalności
- odświeżenie interfejsu aplikacji

**4. Zakończenie współpracy z jedną z firm dostarczających usługi potrzebne funkcjonowaniu aplikacji:**

- pozyskanie nowych sojuszników strategicznych
- renegotiacja warunków dotychczasowej współpracy

**5. Zmiana przepisów regulujących przetwarzanie danych:**

- dostosowanie się do nowych przepisów

**6. Zmniejszenie popytu na transport publiczny:**

- współpraca marketingowa z klientem
- współpraca badawczo-rozwojowa z władzami miasta bądź państwa

**7. Niewłaściwa kontrola nad pracownikami oraz ich nieetyczne zachowania:**

- wyciąganie konsekwencji od pracowników w postaci kary pieniężnej bądź ich zwolnienie
- dokładniejszy proces wprowadzania w zakres obowiązków
- przystępniejsze przedstawienie zagadnień, które obejmuje klauzula poufności

**8. Awaria systemu:**

- zwiększenie nakładów na zapewnianie jakości oprogramowania
- jak najszybsze przywrócenie działania systemu

**9. Utrata finansowania projektu:**

- pozyskanie nowych inwestorów
- skorzystanie z funduszu awaryjnego

## 8 Zarządzanie jakością

### 8.1 Scenariusze i przypadki testowe

ID	1		
Nazwa	Wyświetlanie natężenia ruchu		
Kategoria	Integracyjne		
Opis	Sprawdzenie czy system poprawnie pokazuje bieżące natężenie ruchu drogowego na trasie linii.		
Tester	Tester z projektowego zespołu QA		
Termin	Do wersji stabilnej 1.0.0		
Narzędzia wspomagające	- TestRail		
Przebieg działań	1.	Tester naciska kafelek “Natężenie ruchu” na ekranie głównym.	System wyświetla mapę z aktualnym natężeniem ruchu.
Założenia	Środowisko: Android Warunki wstępne: tester jest zalogowany do aplikacji i ma połączenie z internetem Dane wejściowe: testowe konto kierowcy (driver0123.test@firma.com / ungun)		
Zestaw danych testowych	Brak danych do wprowadzenia.		

ID	2		
Nazwa	Proponowanie objazdu		
Kategoria	Integracyjne		
Opis	Sprawdzenie czy system poprawnie proponuje najlepszy możliwy objazd w przypadku wypadku.		
Tester	Tester z projektowego zespołu QA		
Termin	Do wersji stabilnej 1.0.0		
Narzędzia wspomagające	<ul style="list-style-type: none"> <li>- Charles Proxy</li> <li>- TestRail</li> </ul>		
Przebieg działań	1.	Tester naciska kafelek “Natężenie ruchu” na ekranie głównym.	System wyświetla mapę z aktualnym natężeniem ruchu.
	2.	Tester podmienia dane poprzez oprogramowanie “Charles Proxy”, aby zasymulować wystąpienie wypadku.	System wyświetla dialog z informacją o możliwym objeździe.
	3.	Tester akceptuje zaproponowany przez system objazd.	System odświeża wygląd mapy, przedstawiając nową trasę.
Założenia	<p>Środowisko: Android</p> <p>Warunki wstępne: tester jest zalogowany do aplikacji i ma połączenie z internetem</p> <p>Dane wejściowe: testowe konto kierowcy (driver0123.test@firma.com / ungung)</p>		
Zestaw danych testowych	Brak danych do wprowadzenia.		



ID	3		
Nazwa	Wyświetlanie informacji o opóźnieniach		
Kategoria	Integracyjne		
Opis	Sprawdzenie czy system poprawnie informuje użytkownika o wystąpieniu opóźnień w kursowaniu.		
Tester	Tester z projektowego zespołu QA		
Termin	Do wersji stabilnej 1.0.0		
Narzędzia wspomagające	<ul style="list-style-type: none"> <li>- Charles Proxy</li> <li>- TestRail</li> </ul>		
Przebieg działań	1.	Tester wychodzi z aplikacji na ekran główny urządzenia.	System przechodzi w tryb pracy w tle.
	2.	Tester podmienia dane poprzez oprogramowanie "Charles Proxy", aby zasymulować wystąpienie opóźnienia.	System wyświetla powiadomienie o wystąpieniu opóźnienia.
Założenia	<p>Środowisko: Android</p> <p>Warunki wstępne: tester jest zalogowany do aplikacji i ma połączenie z internetem</p> <p>Dane wejściowe: testowe konto pasażera (passenger0123.test@firma.com / ungunung)</p>		
Zestaw danych testowych	<p>{ delay: 180 } - system wyświetla powiadomienie, ponieważ opóźnienie jest większe niż 120 sekund.</p> <p>{ delay: 0 } - brak reakcji systemu</p> <p>{ delay: null } - brak reakcji systemu</p> <p>{ delay: -120 } - brak reakcji systemu</p> <p>{ delay: 60 } - brak reakcji systemu</p>		

ID	4		
Nazwa	Zarządzanie czasem pracy		
Kategoria	Integracyjne		
Opis	Sprawdzenie czy system poprawnie pozwala na uzupełnienie danych o czasie pracy z danego miesiąca.		
Tester	Tester z projektowego zespołu QA		
Termin	Do wersji stabilnej 1.0.0		
Narzędzia wspomagające	- TestRail		
Przebieg działań	1.	Tester naciska kafelek “Zarządzanie czasem pracy” na ekranie głównym,.	System wyświetla widok zarządzania czasem pracy.
	2.	Tester naciska opcję “Zarządzaj”	System wyświetla menu z dostępnymi opcjami.
	3.	Tester naciska opcję “Uzupełnij czas pracy”	System wyświetla widok uzupełniania czasu pracy z aktualnego miesiąca.
	4.	Tester uzupełnia i zapisuje dane o czasie pracy.	System zapisuje podane przez użytkownika dane lub wyświetla dialog z informacją o błędnych danych.
Założenia	Środowisko: Android Warunki wstępne: tester jest zalogowany do aplikacji i ma połączenie z internetem Dane wejściowe: testowe konto kierowcy (driver0123.test@firma.com / ungung)		
Zestaw danych testowych	Brak danych do wprowadzenia.		

ID	5		
Nazwa	Kontakt ze służbami bezpieczeństwa		
Kategoria	Integracyjne		
Opis	Sprawdzenie czy system poprawnie łączy kierowcę z wybranymi służbami bezpieczeństwa.		
Tester	Tester z projektowego zespołu QA		
Termin	Do wersji stabilnej 1.0.0		
Narzędzia wspomagające	- TestRail		
Przebieg działań	1.	Tester naciska przycisk szybkiego kontaktu ze służbami bezpieczeństwa na ekranie głównym.	System przenosi użytkownika do wbudowanej aplikacji "Telefon" z rozpoczętym połączeniem. Dodatkowo system wysyła lokalizację pojazdu oraz udostępniania obraz z kamer.
Założenia	Środowisko: Android Warunki wstępne: tester jest zalogowany do aplikacji i ma połączenie z internetem Dane wejściowe: testowe konto kierowcy (driver0123.test@firma.com / ungung)		
Zestaw danych testowych	Brak danych do wprowadzenia.		

ID	6		
Nazwa	Kontakt z zespołem technicznym		
Kategoria	Integracyjne		
Opis	Sprawdzenie czy system poprawnie łączy kierowcę z zespołem technicznym.		
Tester	Tester z projektowego zespołu QA		
Termin	Do wersji stabilnej 1.0.0		
Narzędzia wspomagające	- TestRail		
Przebieg działań	1.	Tester naciska przycisk szybkiego kontaktu z zespołem technicznym na ekranie głównym.	System przenosi użytkownika do wbudowanej aplikacji "Telefon" z rozpoczętym połączeniem. Dodatkowo system wysyła lokalizację pojazdu.
Założenia	Środowisko: Android Warunki wstępne: tester jest zalogowany do aplikacji i ma połączenie z internetem Dane wejściowe: testowe konto kierowcy (driver0123.test@firma.com / unguno)		
Zestaw danych testowych	Brak danych do wprowadzenia.		

ID	7		
Nazwa	Kupno biletu		
Kategoria	Integracyjne		
Opis	Sprawdzenie czy system poprawnie obsługuje sprzedaż biletów.		
Tester	Tester z projektowego zespołu QA		
Termin	Do wersji stabilnej 1.0.0		
Narzędzia wspomagające	- TestRail		
Przebieg działań	1.	Tester naciska na kafelek "Kup bilet" na ekranie głównym.	System wyświetla widok listy możliwych do kupienia biletów.
	2.	Tester naciska przycisk "Kup" przy wybranym bilecie.	System przenosi użytkownika do strony zewnętrznego serwisu poprzez domyślną przeglądarkę w celu dokonania zakupu.
	3.	Tester powraca do aplikacji po dokonaniu zakupu.	System wyświetla zakładkę "Moje bilety" pokazując właśnie zakupiony bilet.
Założenia	Środowisko: Android Warunki wstępne: tester jest zalogowany do aplikacji i ma połączenie z internetem Dane wejściowe: testowe konto pasażera (passenger0123.test@firma.com / ungun)		
Zestaw danych testowych	Brak danych do wprowadzenia.		

ID	8		
Nazwa	Wyświetlanie informacji technicznych pojazdu		
Kategoria	Integracyjne		
Opis	Sprawdzenie czy system poprawnie wyświetla informacje o aktualnym stanie technicznym pojazdu.		
Tester	Tester z projektowego zespołu QA		
Termin	Do wersji stabilnej 1.0.0		
Narzędzia wspomagające	- TestRail		
Przebieg działań	1.	Tester naciska kafelek "Informacje techniczne" na ekranie głównym.	System wyświetla ekran wyboru pojazdu oraz daty.
	2.	Tester wybiera konkretny pojazd oraz datę stanu danych.	System wyświetla informacje techniczne o wybranym pojeździe z wybranego dnia.
Założenia	Środowisko: Android Warunki wstępne: tester jest zalogowany do aplikacji i ma połączenie z internetem Dane wejściowe: testowe konto zespołu technicznego (technicalservice0123.test@firma.com / ungung)		
Zestaw danych testowych	{ side_number: 2813, date: 2021-05-15} - wyświetlenie informacji technicznych { side_number: 2813, date: 2021-09-15} - informacja o błędnej dacie { side_number: 4444, date: 2021-05-15} - informacja o błędnym numerze bocznym pojazdu		

ID	9		
Nazwa	Wyświetlanie statystyk o zapelnieniu pojazdu		
Kategoria	Integracyjne		
Opis	Sprawdzenie czy system poprawnie wyświetla statystyki o zapelnieniu pojazdu.		
Tester	Tester z projektowego zespołu QA		
Termin	Do wersji stabilnej 1.0.0		
Narzędzia wspomagające	- TestRail		
Przebieg działań	1.	Tester naciska kafelek "Statystyki zapelnienia" na ekranie głównym.	System wyświetla ekran wyboru brygady oraz daty i godziny.
	2.	Tester wybiera konkretną brygadę oraz datę i godzinę.	System wyświetla statystyki o zapelnieniu wybranej brygady z wybranego dnia i godziny.
Założenia	Środowisko: Android Warunki wstępne: tester jest zalogowany do aplikacji i ma połączenie z internetem Dane wejściowe: testowe konto kierownika (manager0123.test@firma.com / unguno)		
Zestaw danych testowych	{ brigade:7 , date: 2021-05-15} - wyświetlenie statystyk o zapelnieniu { brigade: -1, date: 2021-05-15} - informacja o błędnej brygadzie na linii { brigade:7, date: 2021-09-15} - informacja o błędnej dacie		

ID	10		
Nazwa	Wyświetlanie możliwości przesiadki		
Kategoria	Integracyjne		
Opis	Sprawdzenie czy system poprawnie wyświetla informacje o możliwych przesiadkach.		
Tester	Tester z projektowego zespołu QA		
Termin	Do wersji stabilnej 1.0.0		
Narzędzia wspomagające	<ul style="list-style-type: none"> <li>- Charles Proxy</li> <li>- TestRail</li> </ul>		
Przebieg działań	1.	Tester naciska kafelek “Przesiadki” na ekranie głównym.	System wyświetla ekran z filtrowaną listą do wyboru kursu.
	2.	Tester wybiera konkretny kurs z listy i podmienia dane poprzez oprogramowanie “Charles Proxy”, aby zasymulować konkretne odjazdy..	System wyświetla listę wszystkich kolejnych przystanków na trasie wraz z trzema najbliższymi z niego odjazdami środków transportu po przewidywanym przyjeździe wybranego wcześniej kursu.
Założenia	<p>Środowisko: Android</p> <p>Warunki wstępne: tester jest zalogowany do aplikacji i ma połączenie z internetem</p> <p>Dane wejściowe: testowe konto pasażera (passenger0123.test@firma.com / ungun)</p>		
Zestaw danych testowych	<p>{line: 53, direction:Pomorzany Dobrzyńska, stop:Wernyhory, nearest_departure:7} - wyświetlenie na liście odjazdu za 7min</p> <p>{line: 53, direction:Pomorzany Dobrzyńska, stop:qwerty, nearest_departure:7} - brak przedstawienia kursu ze względu na błędne dane</p> <p>{line: 53, direction:Pomorzany Dobrzyńska, stop:Wernyhory, nearest_departure:null} - wyświetlenie statycznego czasu przyjazdu np. 20:53</p>		



ID	11		
Nazwa	Modyfikowanie danych technicznych pojazdu		
Kategoria	Integracyjne		
Opis	Sprawdzenie czy system poprawnie pozwala na zmodyfikowanie istniejących już danych technicznych pojazdu.		
Tester	Tester z projektowego zespołu QA		
Termin	Do wersji stabilnej 1.0.0		
Narzędzia wspomagające	- TestRail		
Przebieg działań	1.	Tester naciska kafelek "Informacje techniczne" na ekranie głównym.	System wyświetla ekran wyboru pojazdu oraz daty.
	2.	Tester wybiera konkretny pojazd oraz datę stanu danych.	System wyświetla informacje techniczne o wybranym pojeździe z wybranego dnia.
	3.	Tester naciska opcję "Edytuj".	System wyświetla ekran edycji danych technicznych pojazdu.
	4.	Tester podmienia dane i naciska opcję "Zapisz".	System zapisuje w bazie danych dane techniczne pojazdu.
Założenia	Środowisko: Android Warunki wstępne: tester jest zalogowany do aplikacji i ma połączenie z internetem Dane wejściowe: testowe konto zespołu technicznego (technicalservice0123.test@firma.com / ungung)		
Zestaw danych testowych	{last_inspection_date: 2021-05-15} - poprawne nadpisanie daty przeglądu technicznego {last_inspection_date: 2021-99-15} - wyświetlenie informacji o błędnej dacie przeglądu technicznego		

## 9 Projekt techniczny

### 9.1 Opis architektury systemu

System będzie oparty o architekturę MVVM, która zakłada podział na trzy elementy: modelu, widoku, viewmodelu. Model jest odpowiedzialny za przechowywanie danych i nie powinien posiadać żadnych elementów logiki biznesowej. Widok to warstwa wizualna do której dane są “bindowane” poprzez przypisaną instancję viewmodelu. Viewmodel natomiast przetwarza dane wykorzystywane do obsługi interakcji wywołanych przez użytkownika, przechowuje referencje do modelu oraz jest “bindowany” jako kontekst do widoku. Dzięki tam zastosowanej architekturze aplikacja będzie zawierała jawną separację warstw.

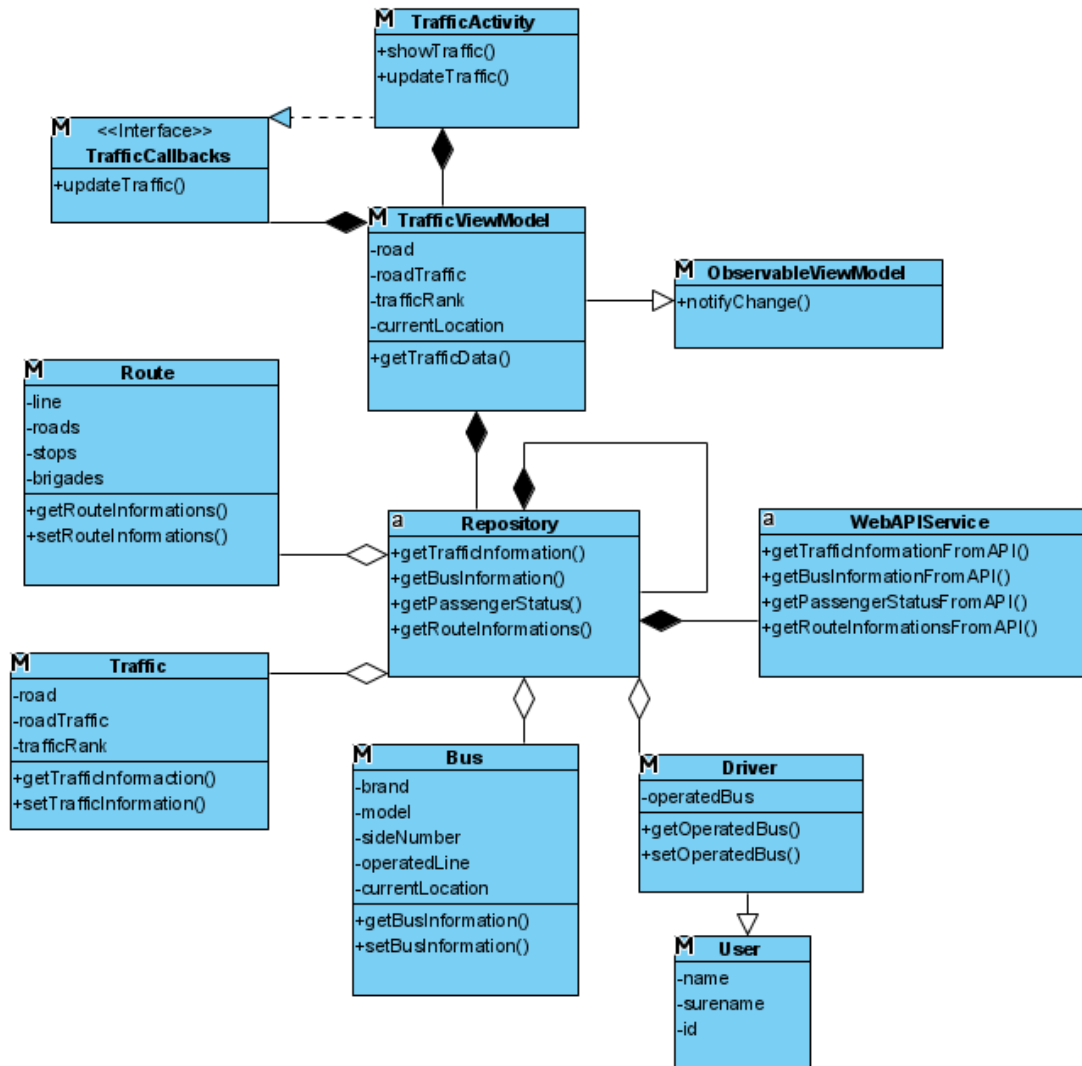
### 9.2 Technologie implementacji systemu

Kotlin	Język programowania przystosowany do tworzenia aplikacji na system Android.
Android Jetpack	Zestaw bibliotek wspomagających programowanie na wszystkie wersje systemu Android.
Retrofit	Biblioteka ułatwiająca wymianę danych z API.
Coroutines	Biblioteka do obsługi programowania asynchronicznego.
Koin	Biblioteka do wstrzykiwania zależności.
Jenkins	Serwer do zautomatyzowanego zarządzania budowaniem aplikacji.
SonarQube	Platforma do zautomatyzowanej kontroli jakości kodu.

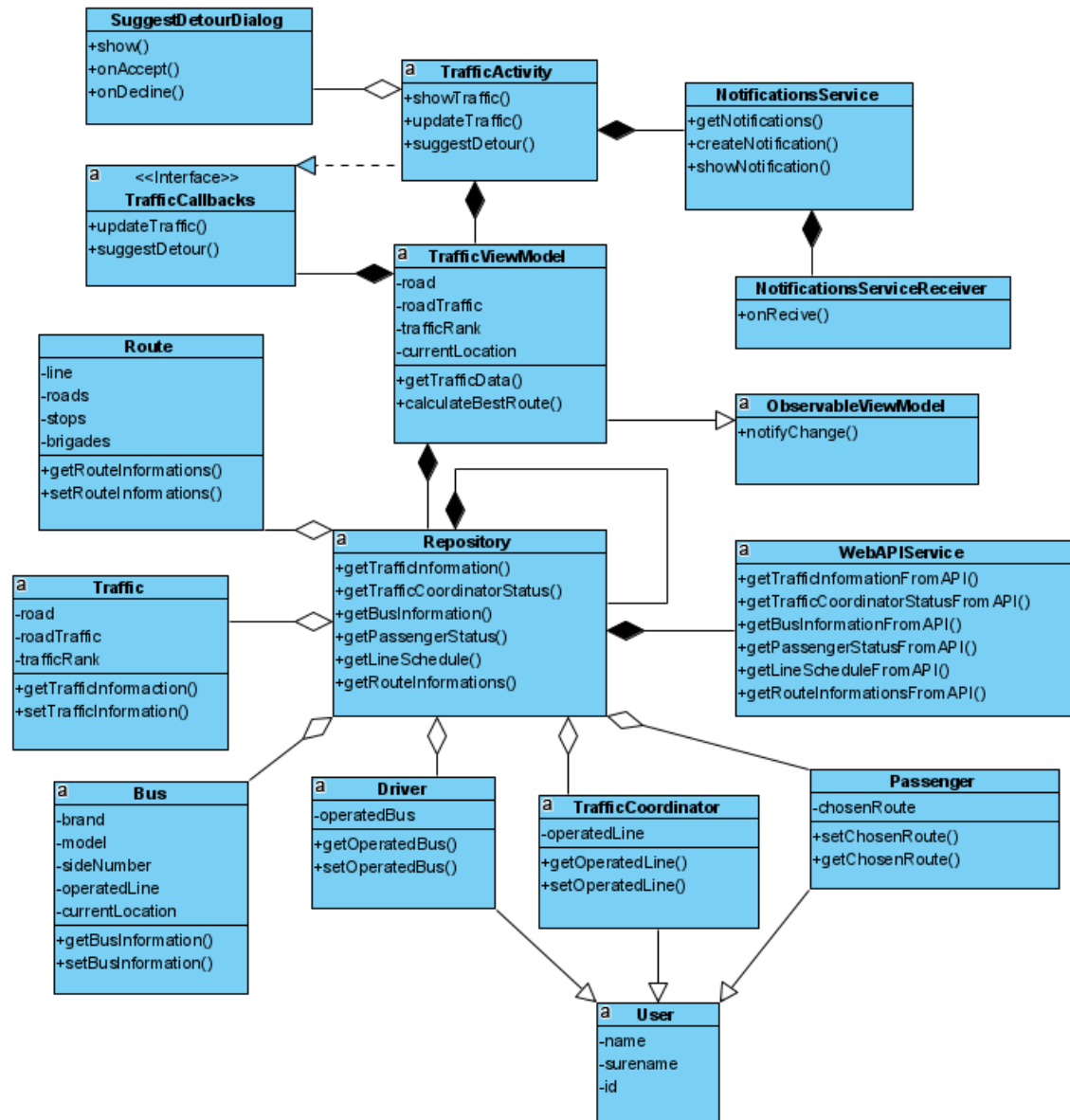
## 9.3 Diagramy UML

### 9.3.1 Diagramy klas

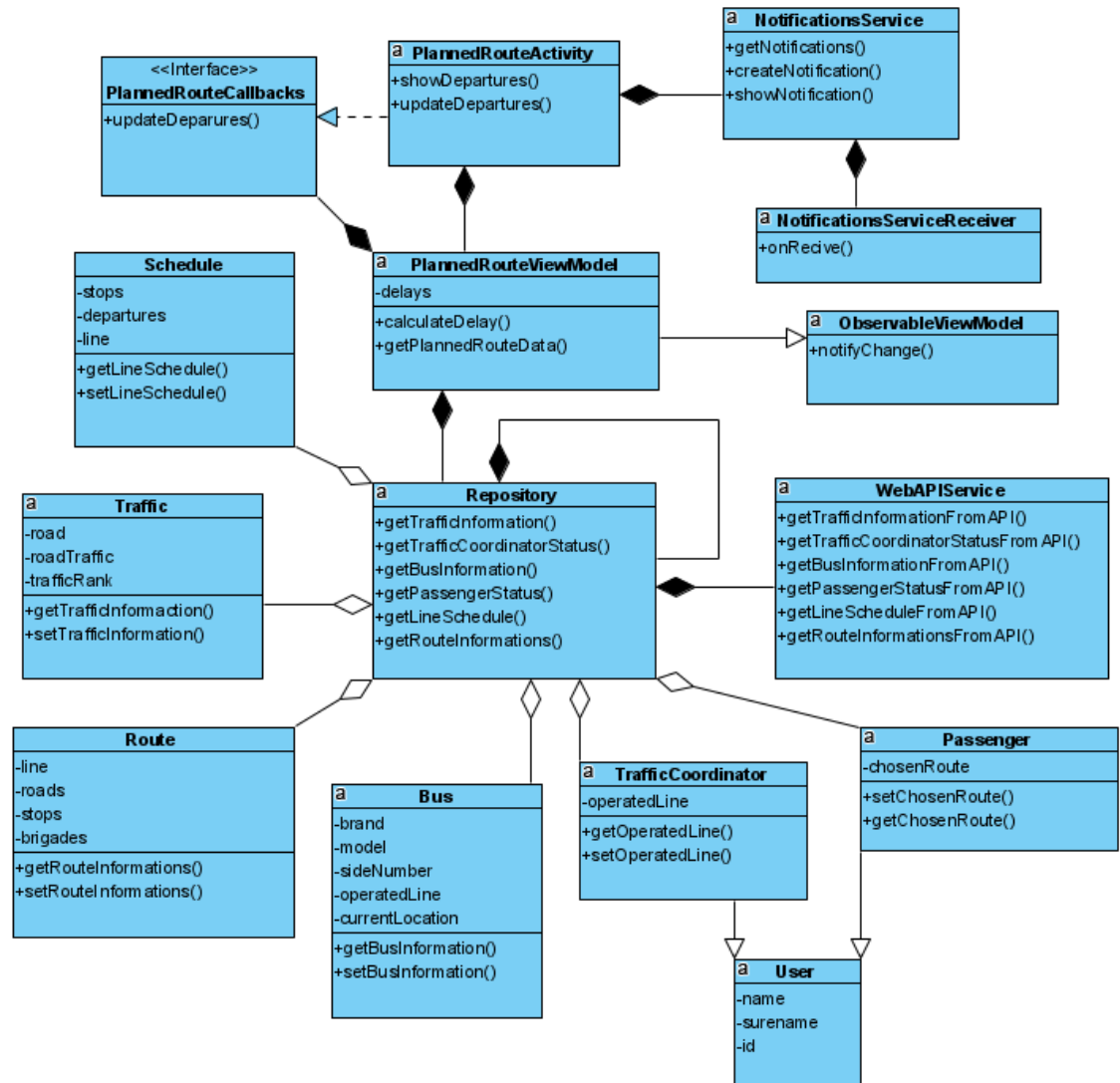
1. System powinien pokazywać na bieżąco natężenia ruchu drogowego na trasie linii.



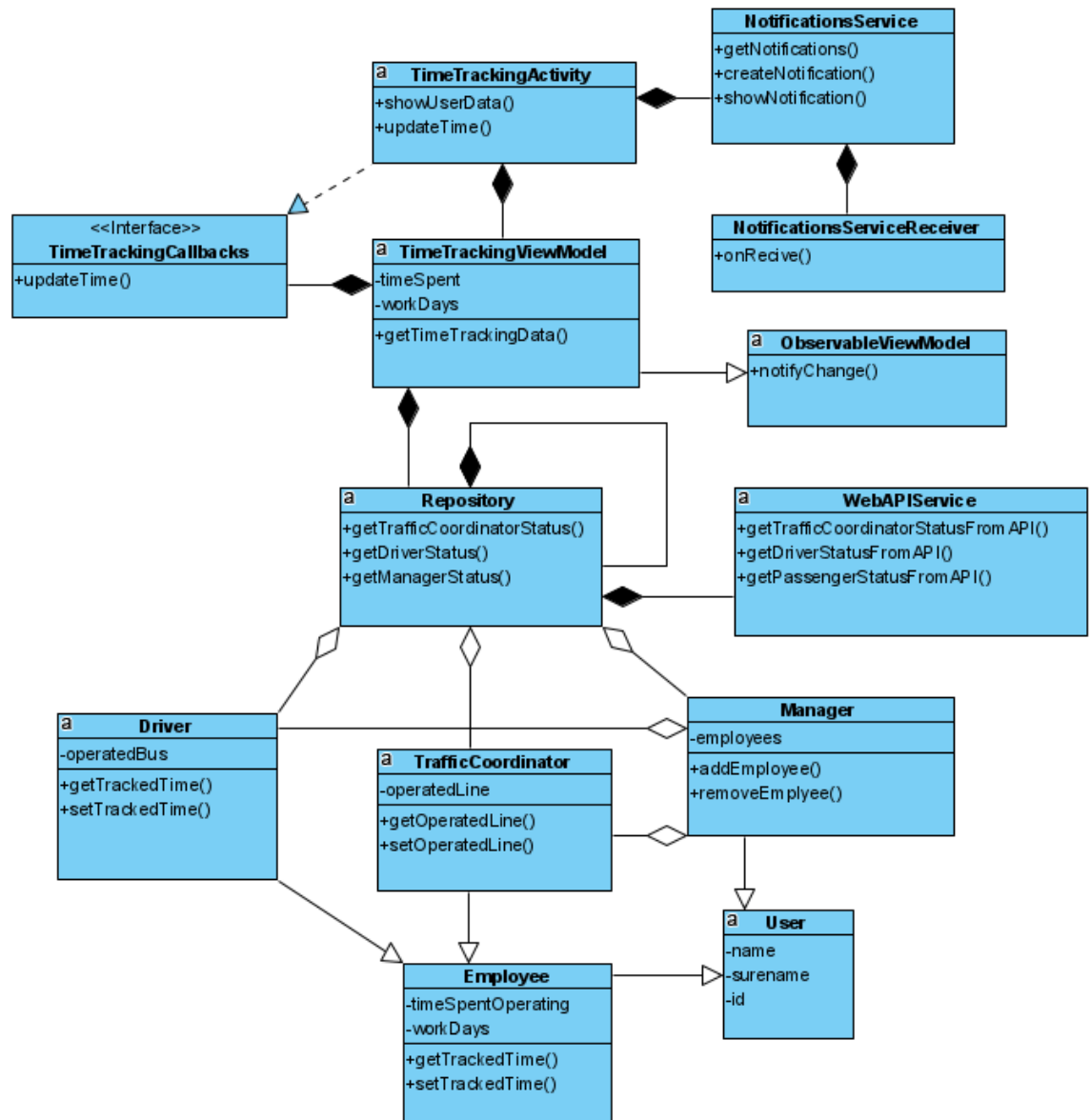
2. System powinien proponować najlepsze możliwe objazdy w przypadku korków lub wypadków na trasie.



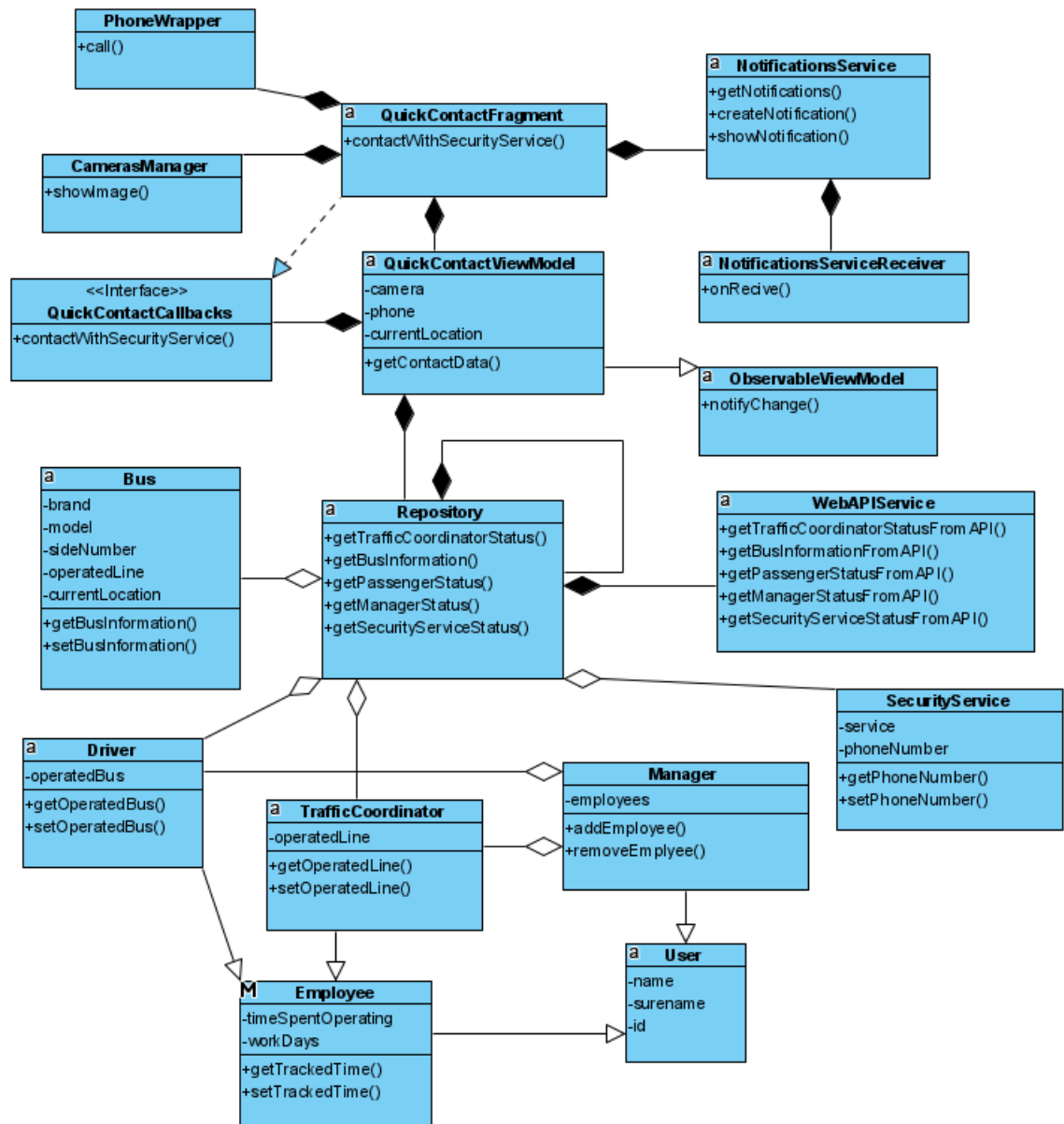
3. System powinien informować o opóźnieniach w kursowaniu.



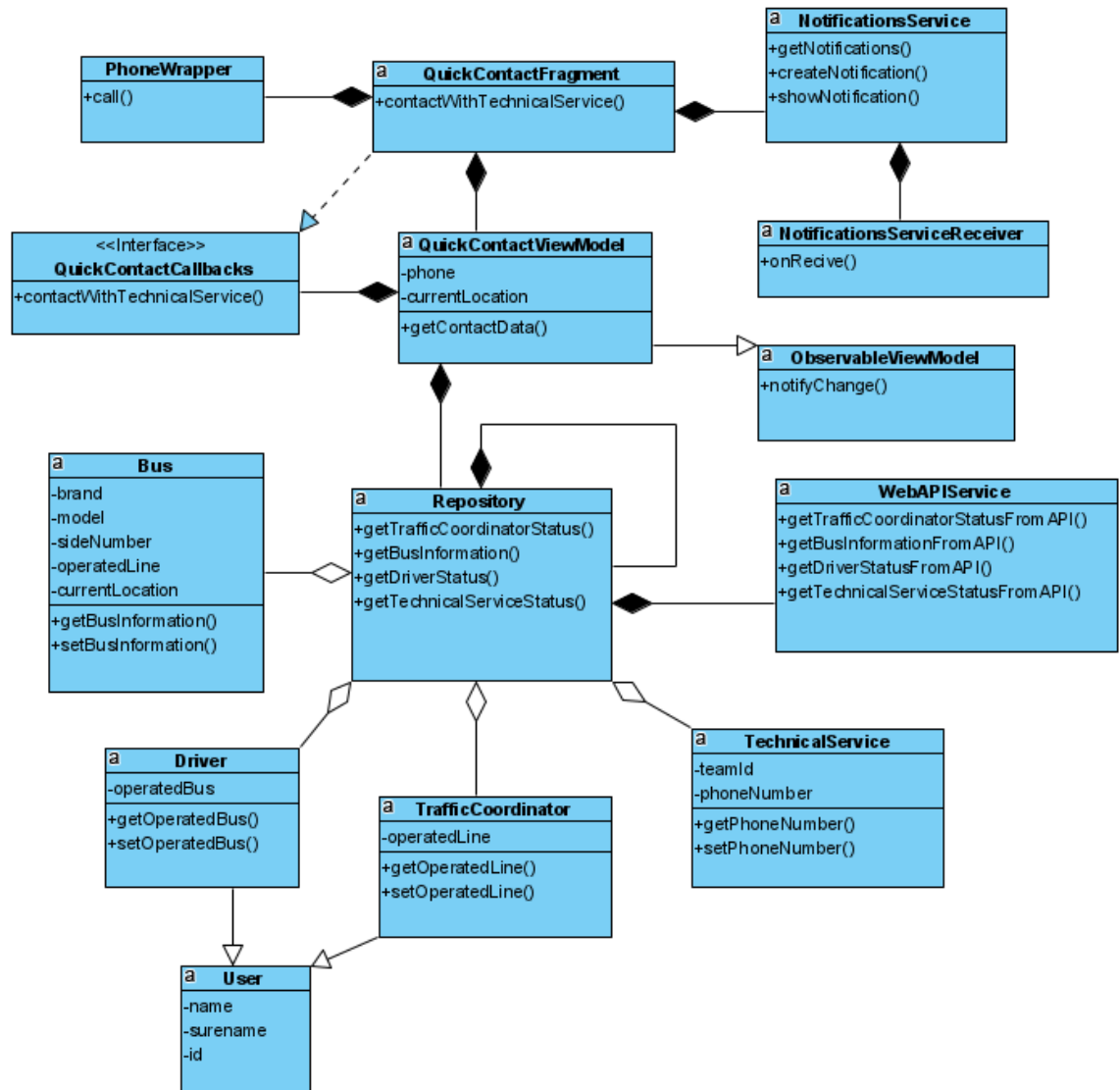
4. System powinien umożliwiać zarządzanie czasem pracy.



5. System powinien umożliwiać szybki kontakt ze służbami bezpieczeństwa.

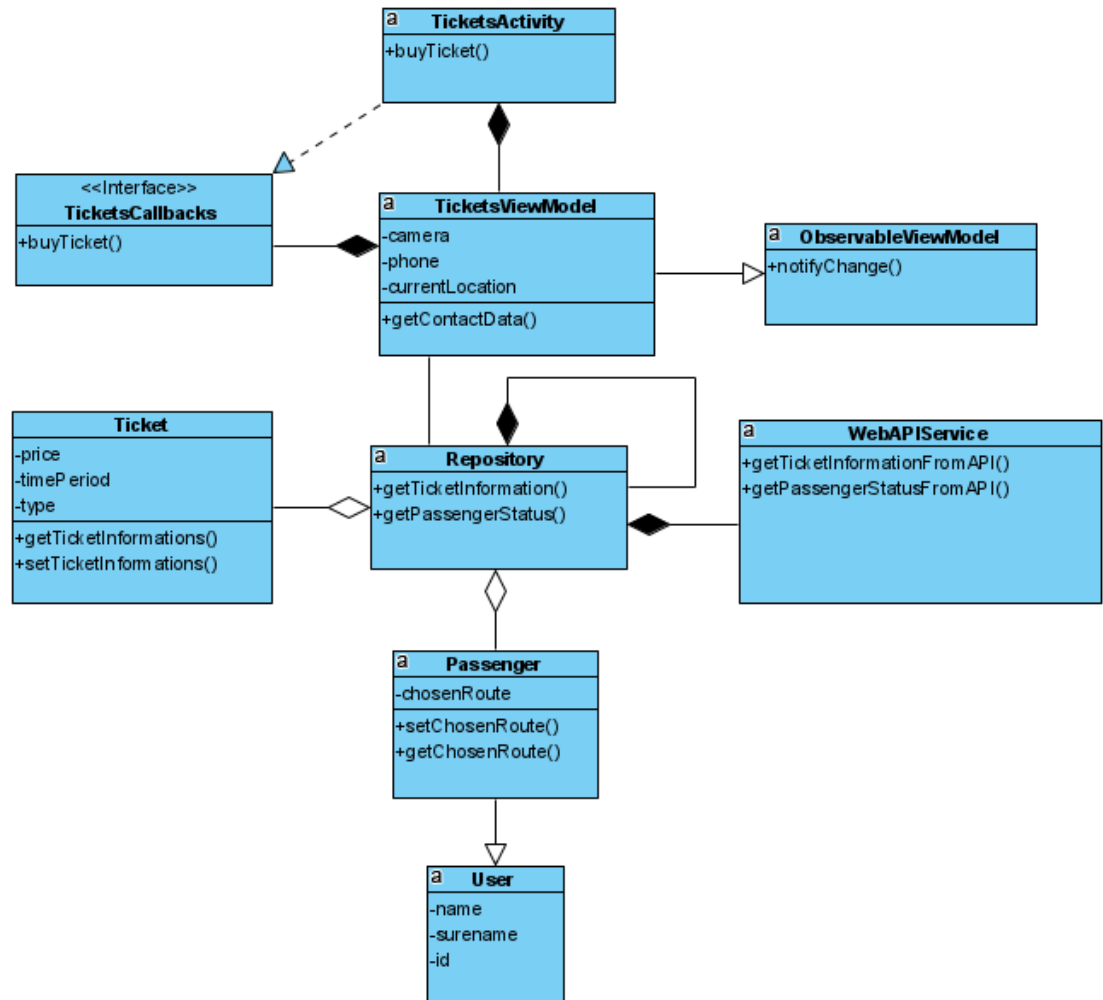


6. System powinien umożliwiać kontakt z zespołem technicznym w przypadku awarii lub usterki.

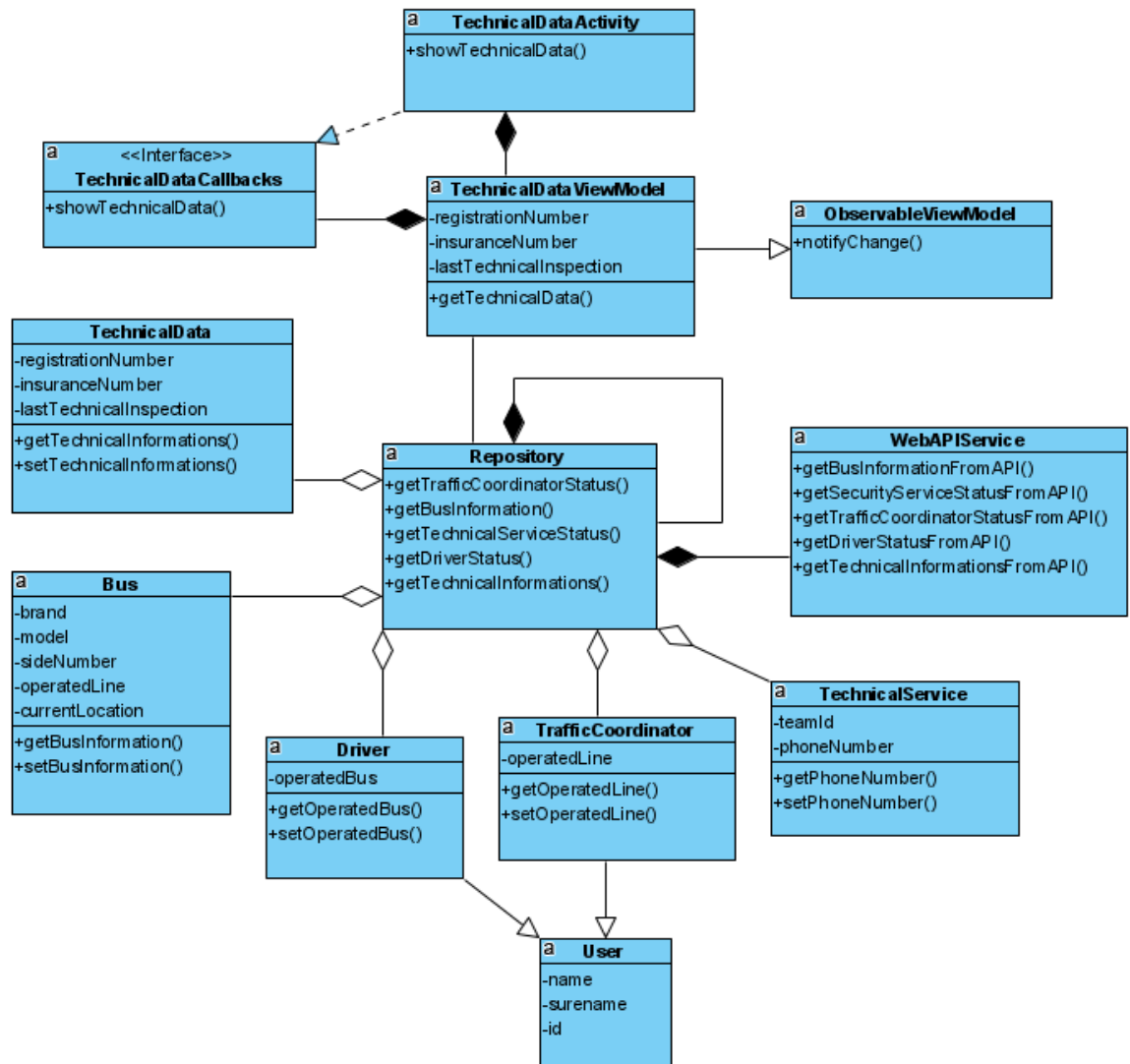




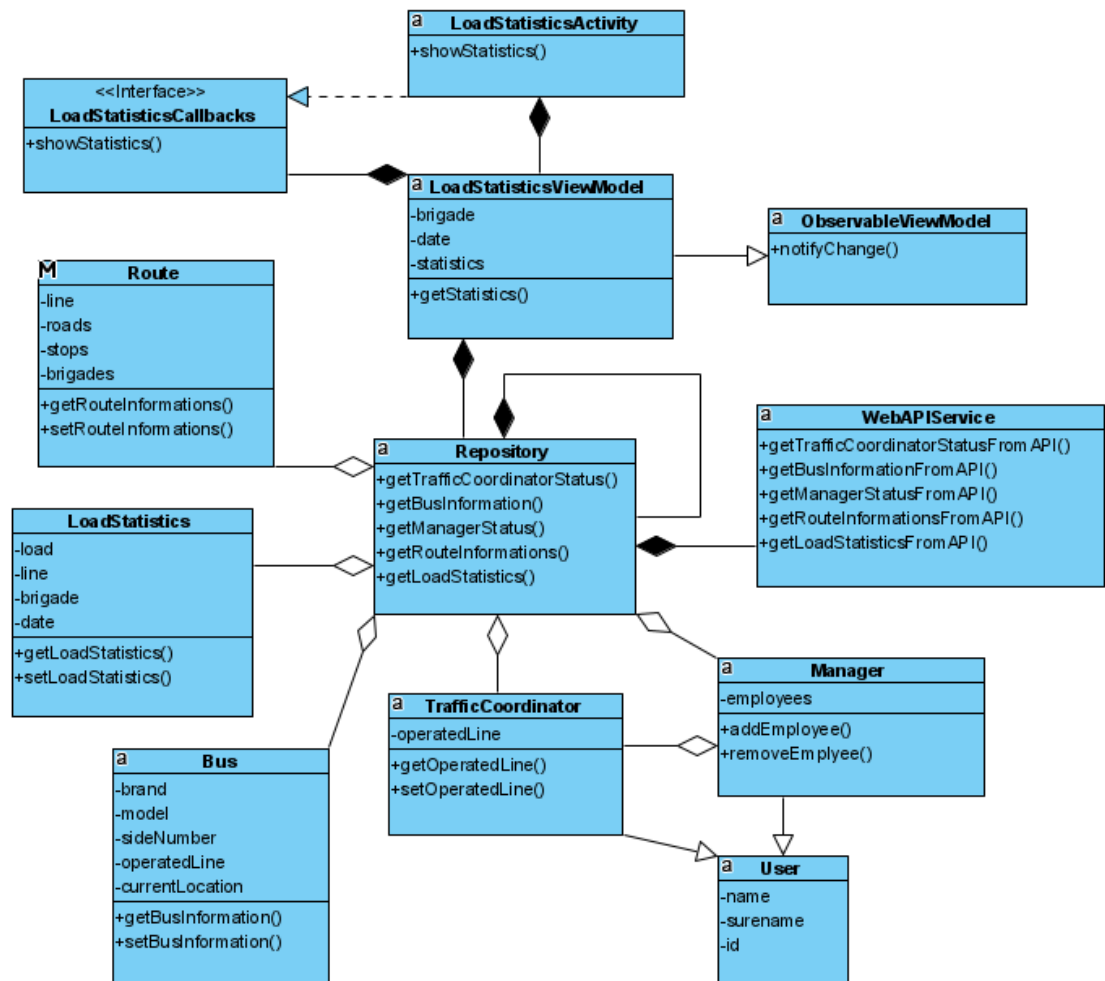
7. System powinien umożliwiać sprzedaż biletów.



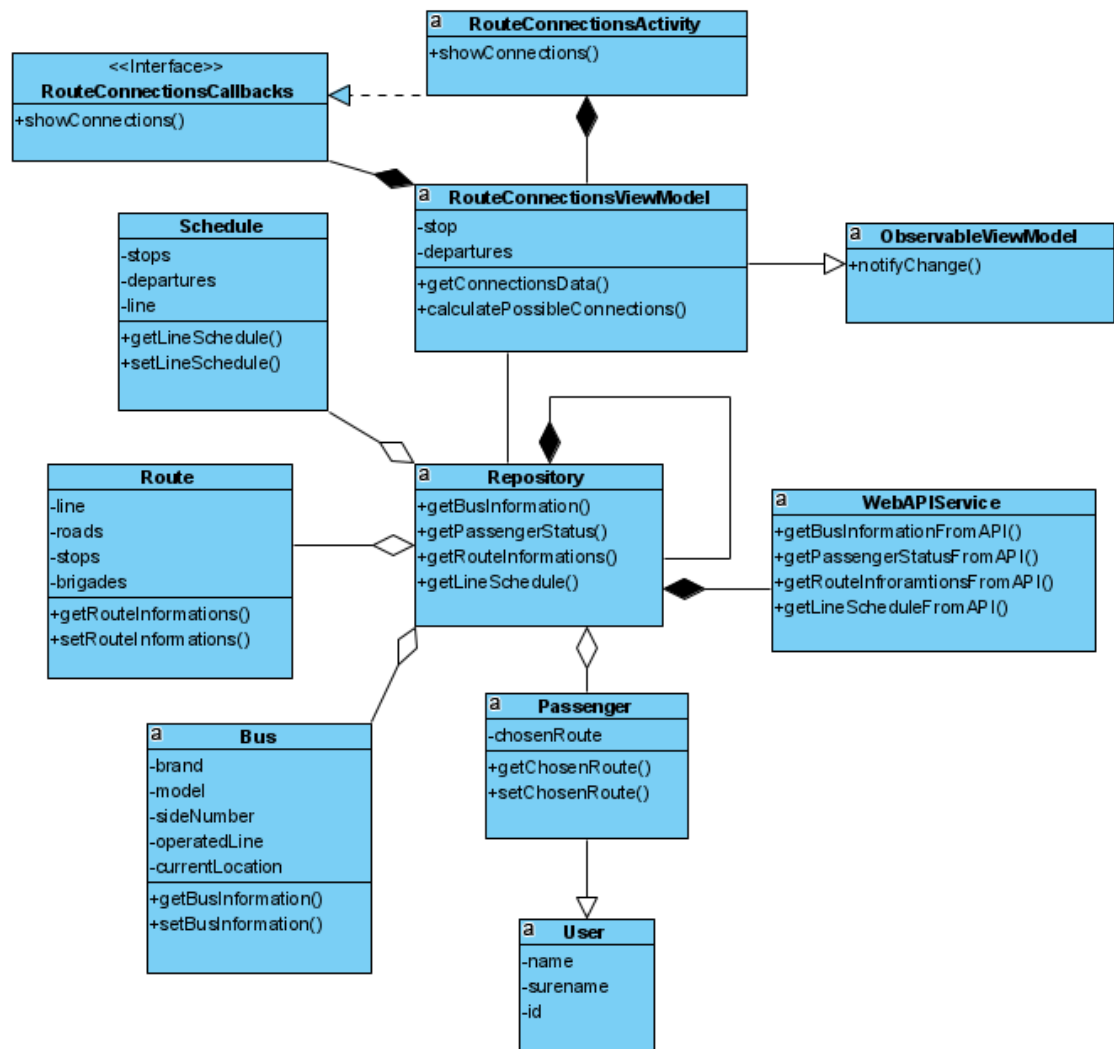
8. System powinien pokazywać informacje techniczne pojazdu.



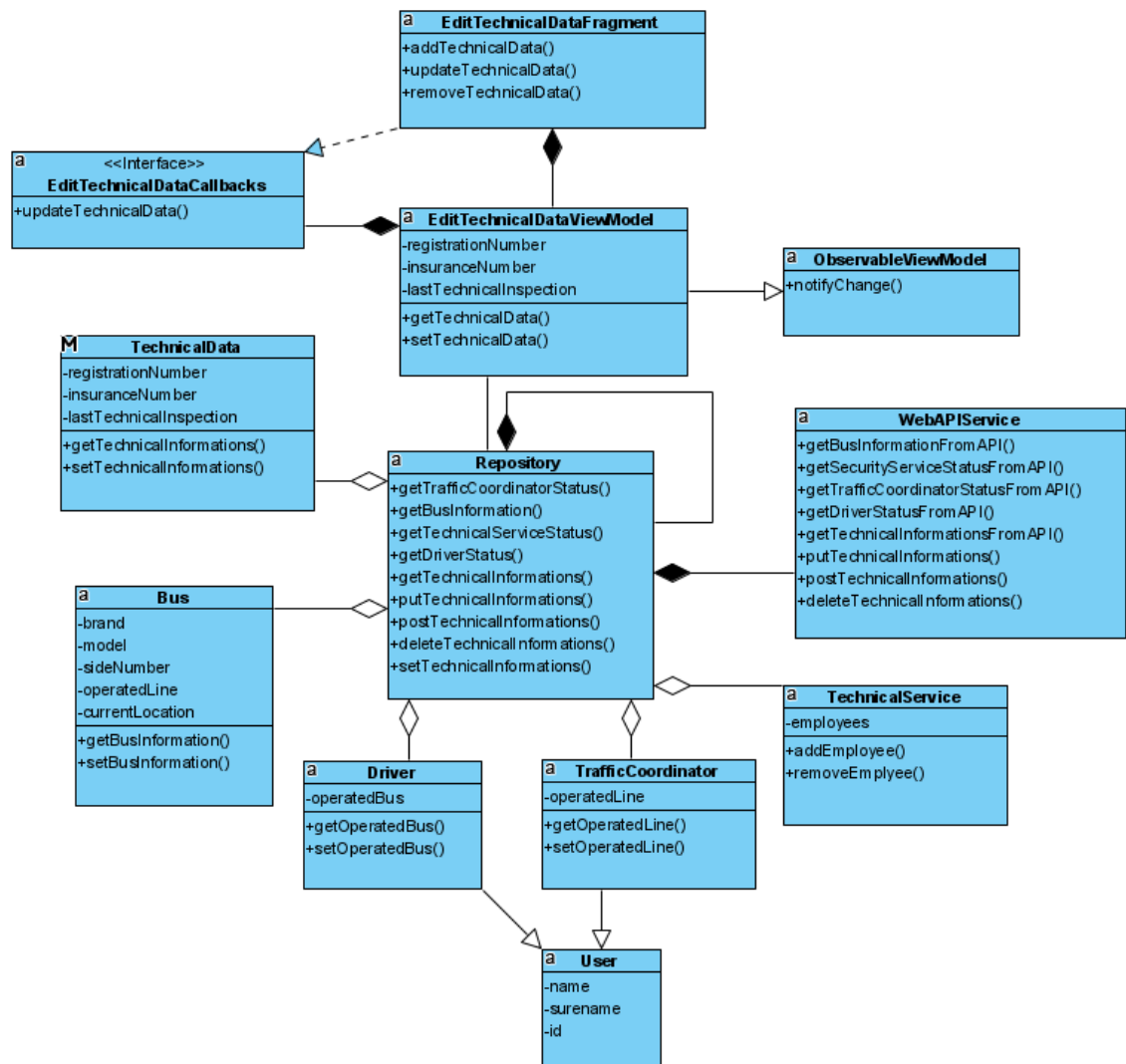
9. System powinien pokazywać statystyki o zapelnieniu pojazdu.



10. System powinien pokazywać możliwość przesiadki na inny autobus, tramwaj lub pociąg.

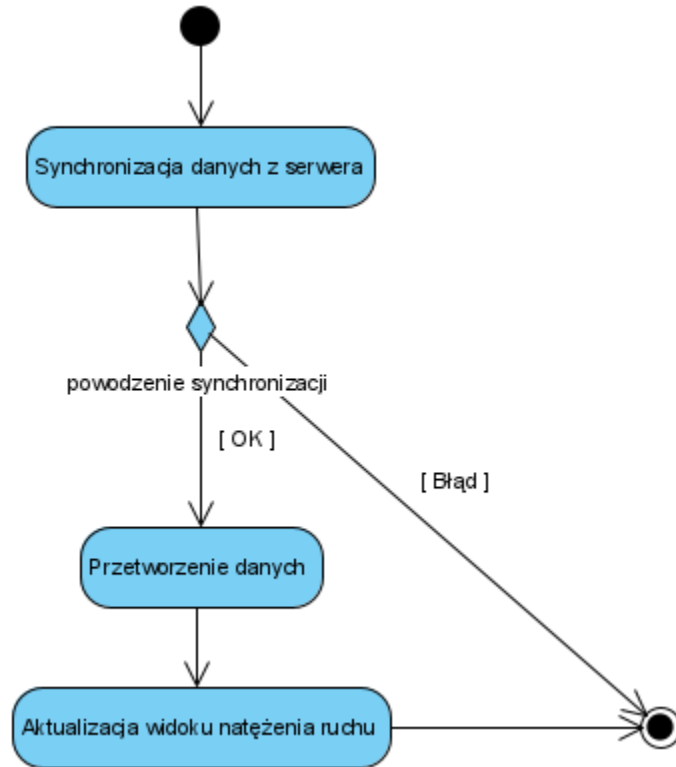


11. System powinien umożliwiać modyfikowanie danych technicznych pojazdu.

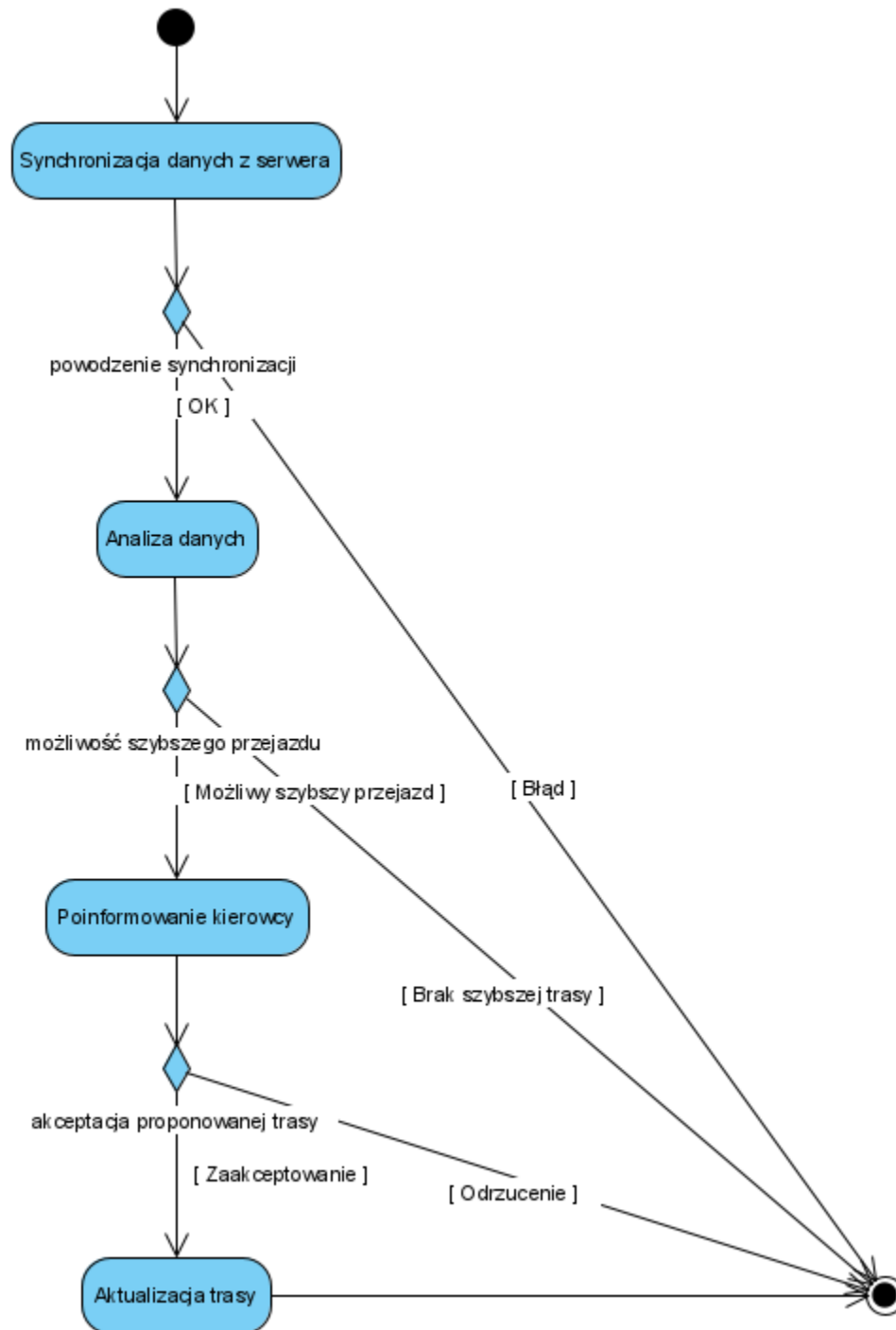


### 9.3.2 Diagramy czynności

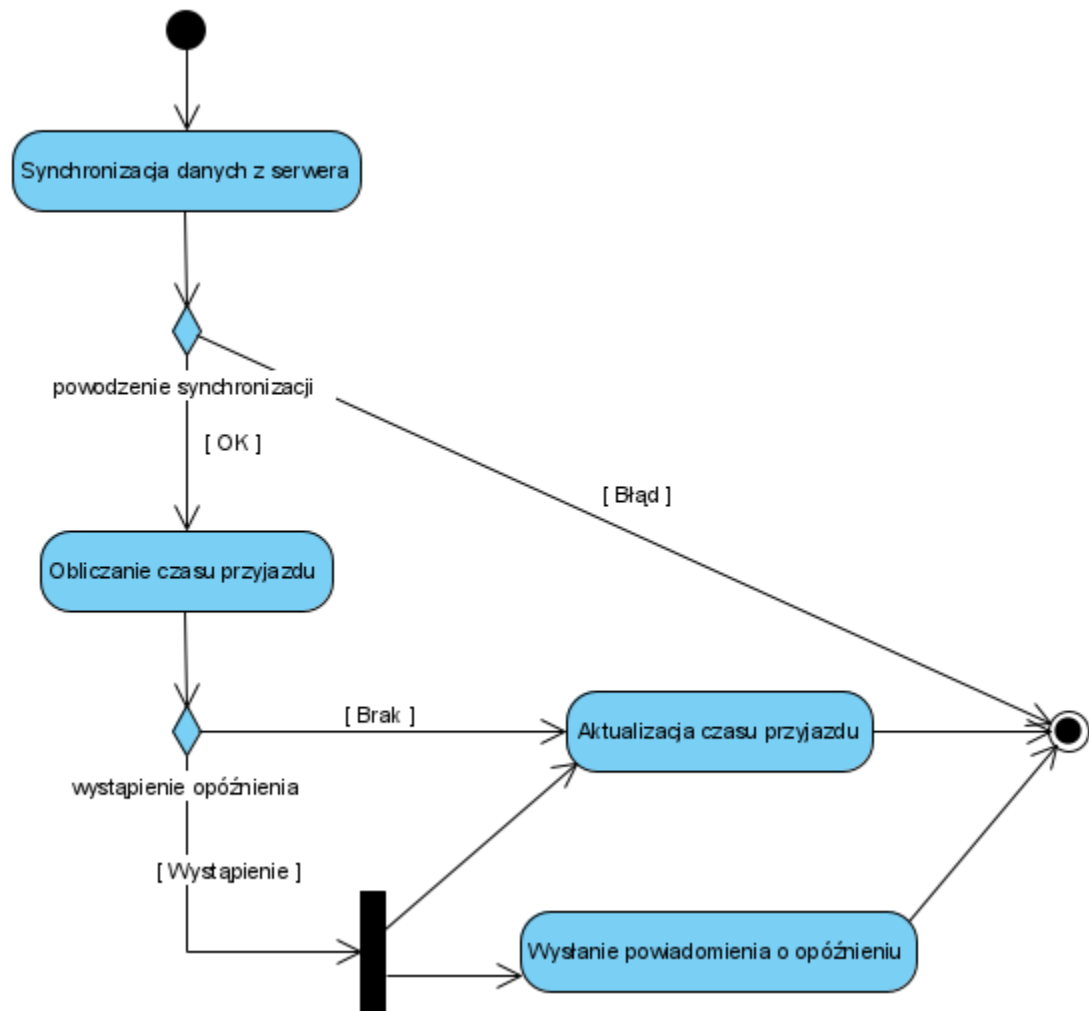
1. System powinien pokazywać na bieżąco natężenia ruchu drogowego na trasie linii.



2. System powinien proponować najlepsze możliwe objazdy w przypadku korków lub wypadków na trasie.

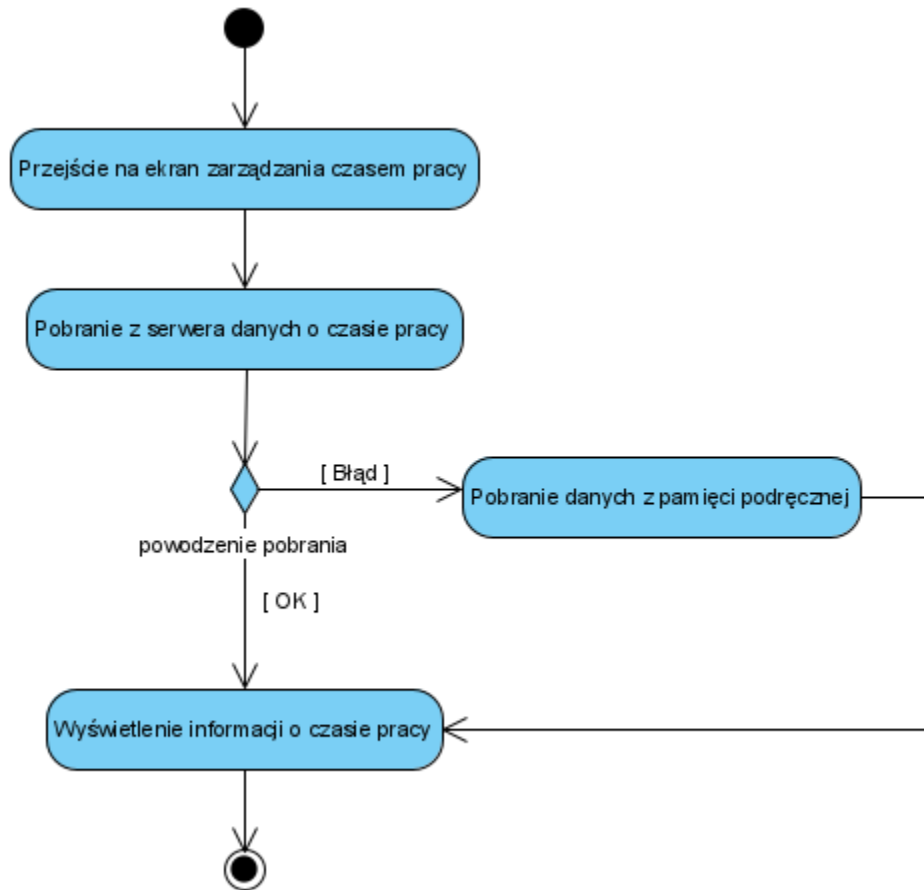


3. System powinien informować o opóźnieniach w kursowaniu.

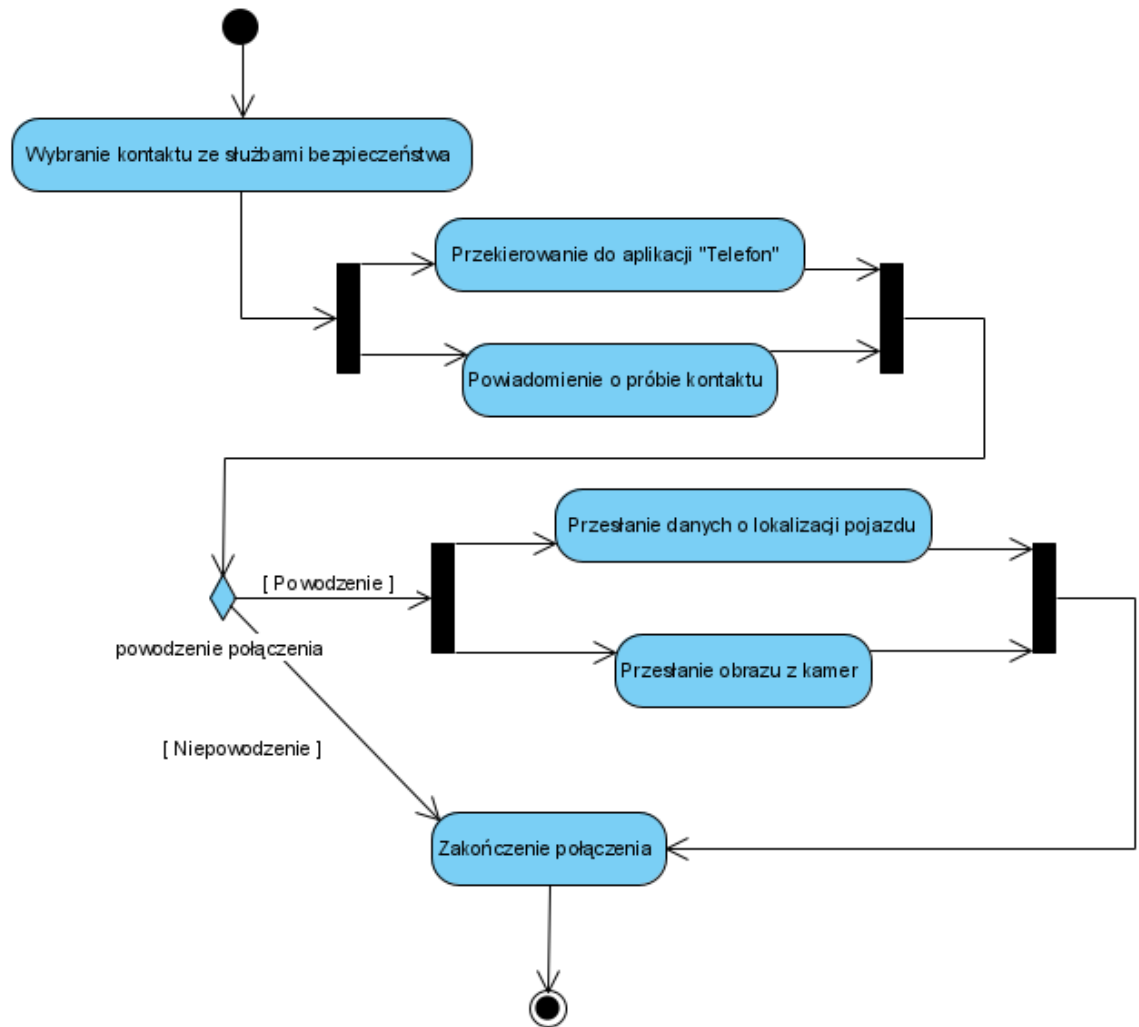




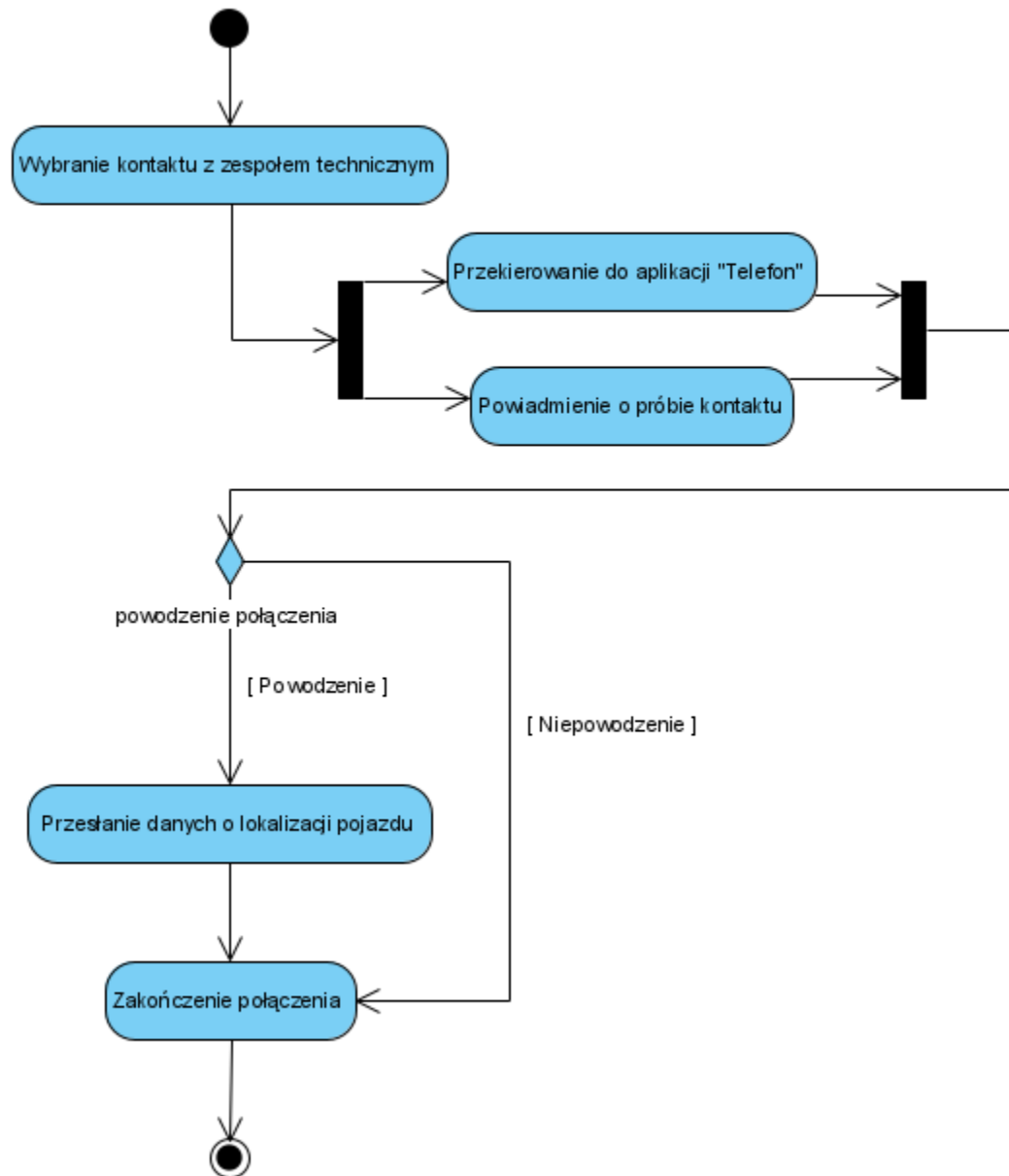
4. System powinien umożliwiać zarządzanie czasem pracy.



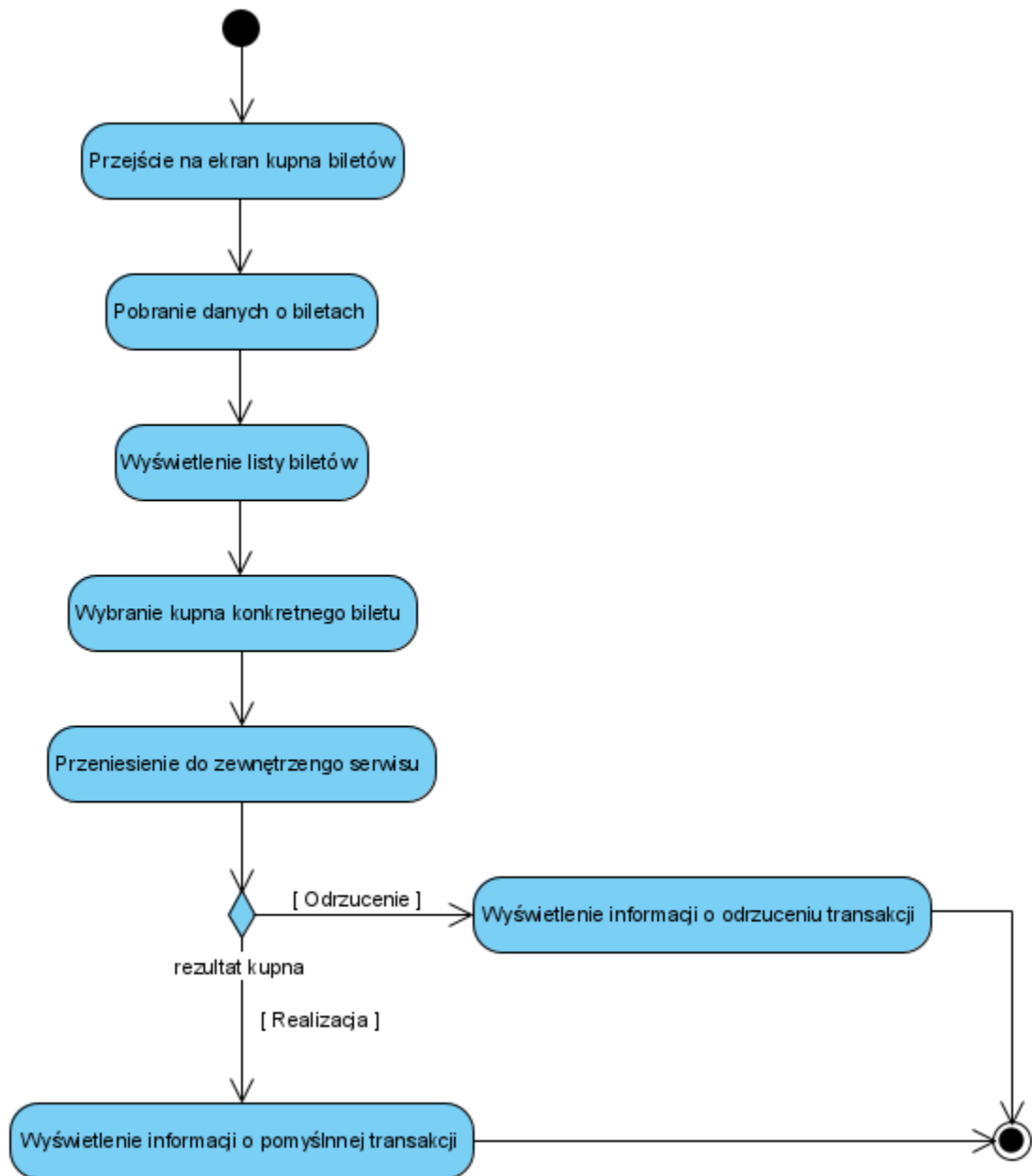
5. System powinien umożliwiać szybki kontakt ze służbami bezpieczeństwa.



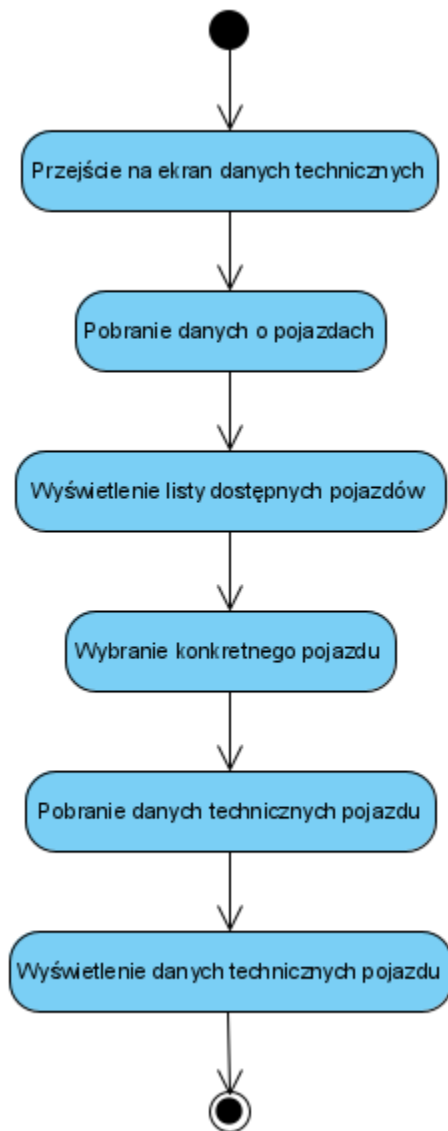
6. System powinien umożliwiać kontakt z zespołem technicznym w przypadku awarii lub usterki.



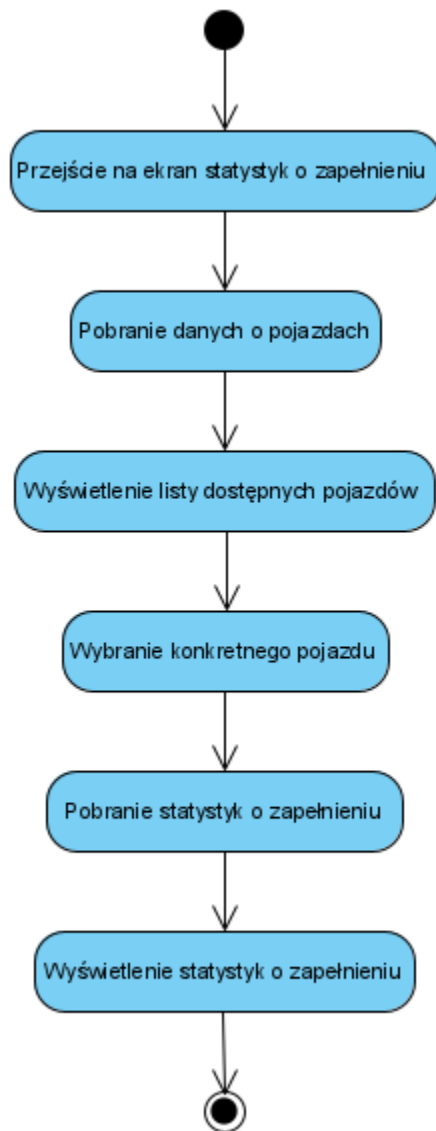
7. System powinien umożliwiać sprzedaż biletów.



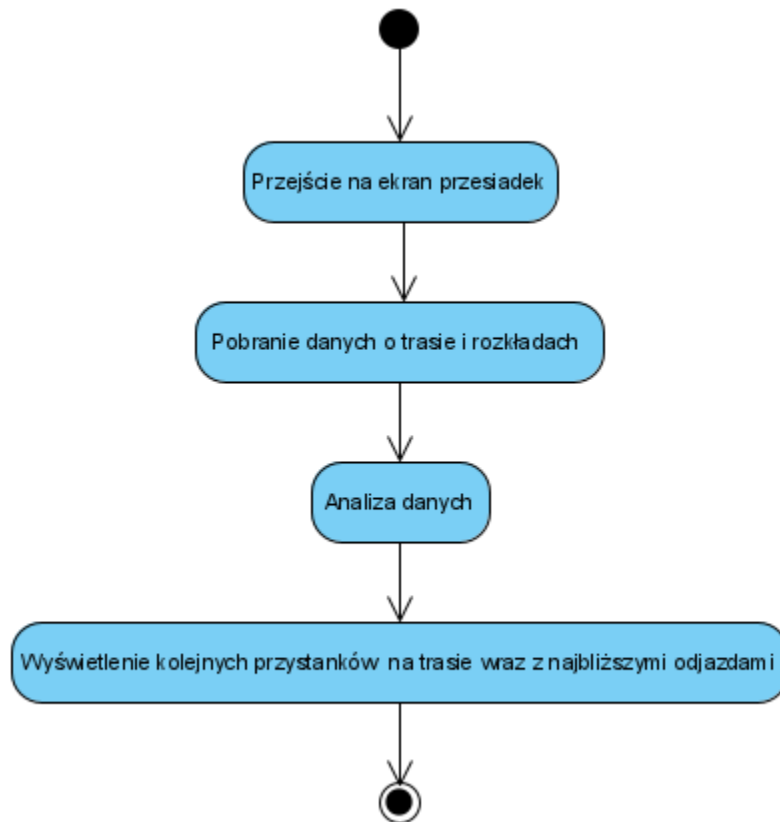
8. System powinien pokazywać informacje techniczne pojazdu



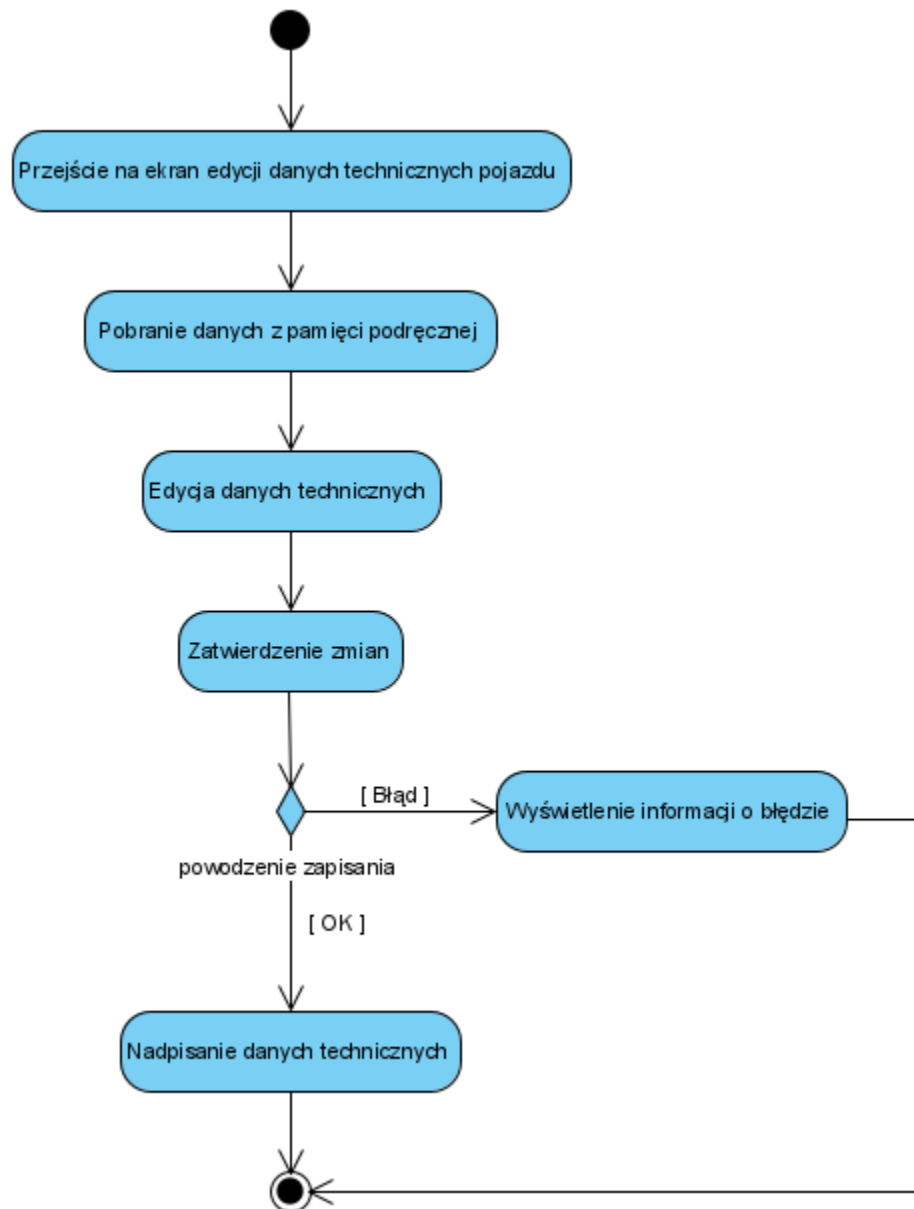
9. System powinien pokazywać statystyki o zapelnieniu pojazdu.



10. System powinien pokazywać możliwość przesiadki na inny autobus, tramwaj lub pociąg.



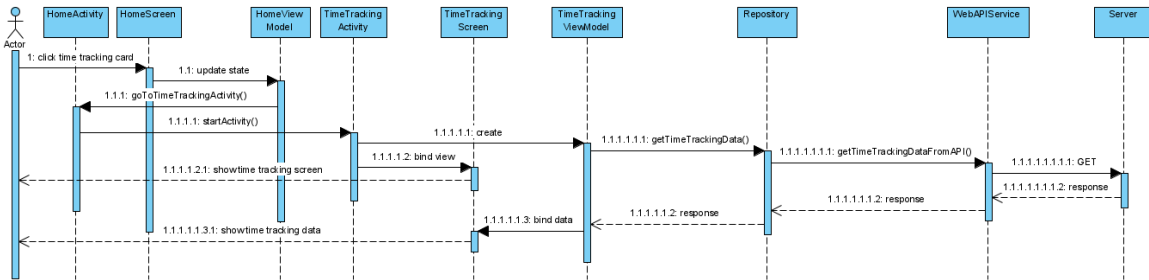
11. System powinien umożliwiać modyfikowanie danych technicznych pojazdu.



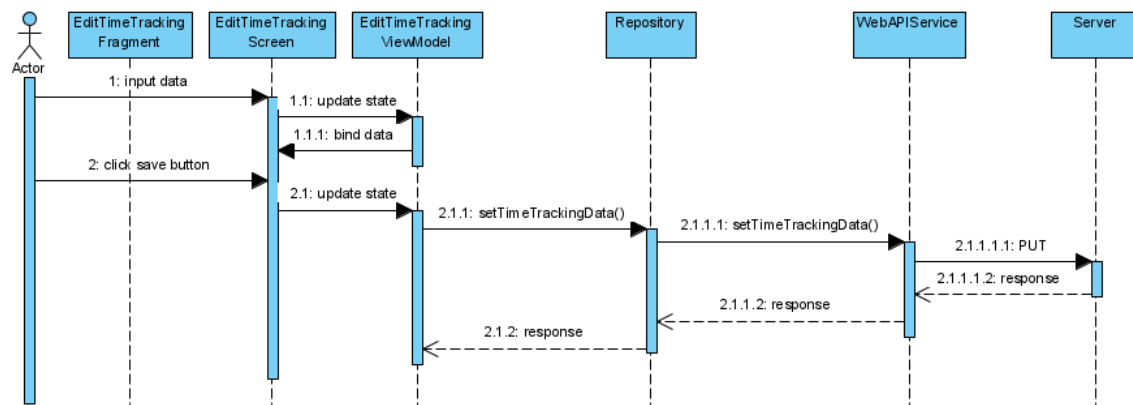


### 9.3.3 Diagramy sekwencji

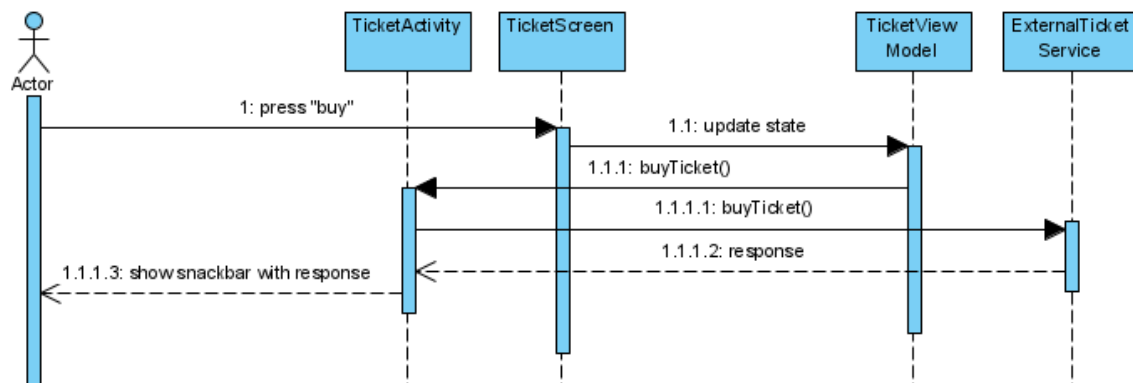
#### 1. Wyświetlanie czasu pracy



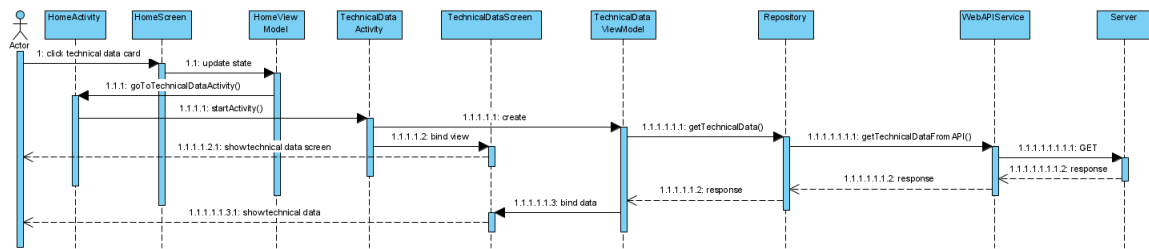
#### 2. Edycja czasu pracy



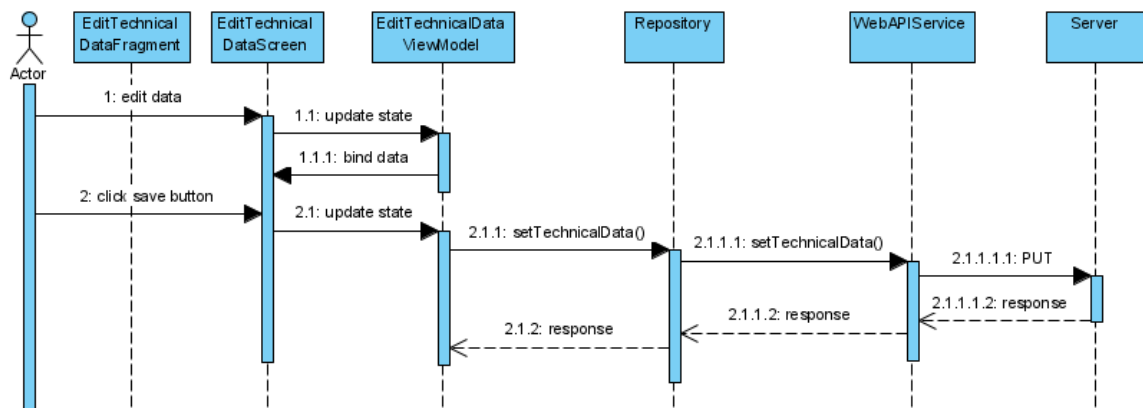
#### 3. Sprzedaż biletów



#### 4. Wyświetlanie danych technicznych pojazdu

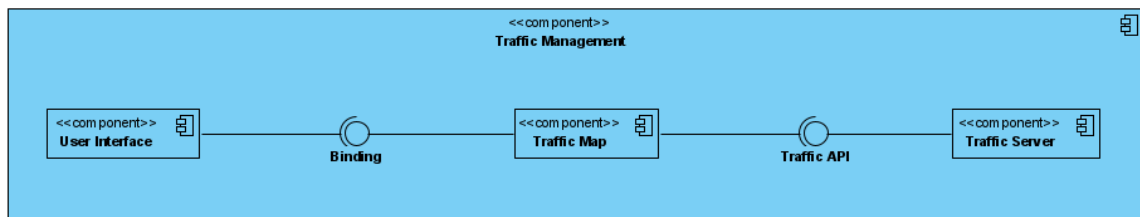


#### 5. Modyfikowanie danych technicznych pojazdu

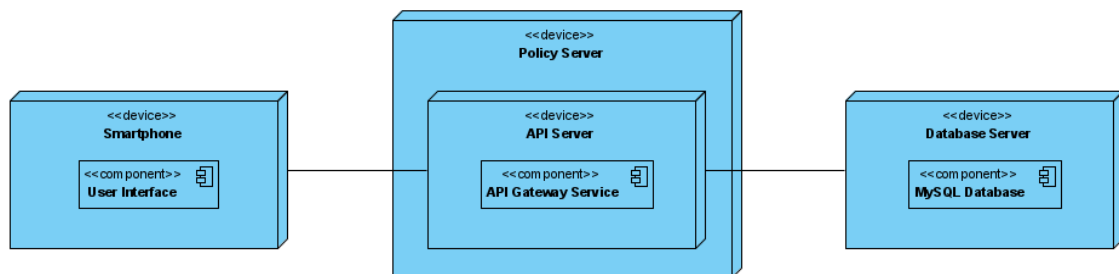


### 9.3.4 Inne diagramy

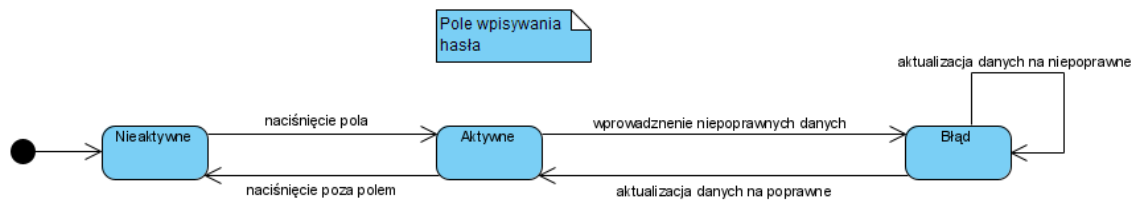
#### 9.3.4.1 Diagram komponentów



#### 9.3.4.2 Diagram rozmieszczenia



### 9.3.4.3 Diagram maszyny stanowej

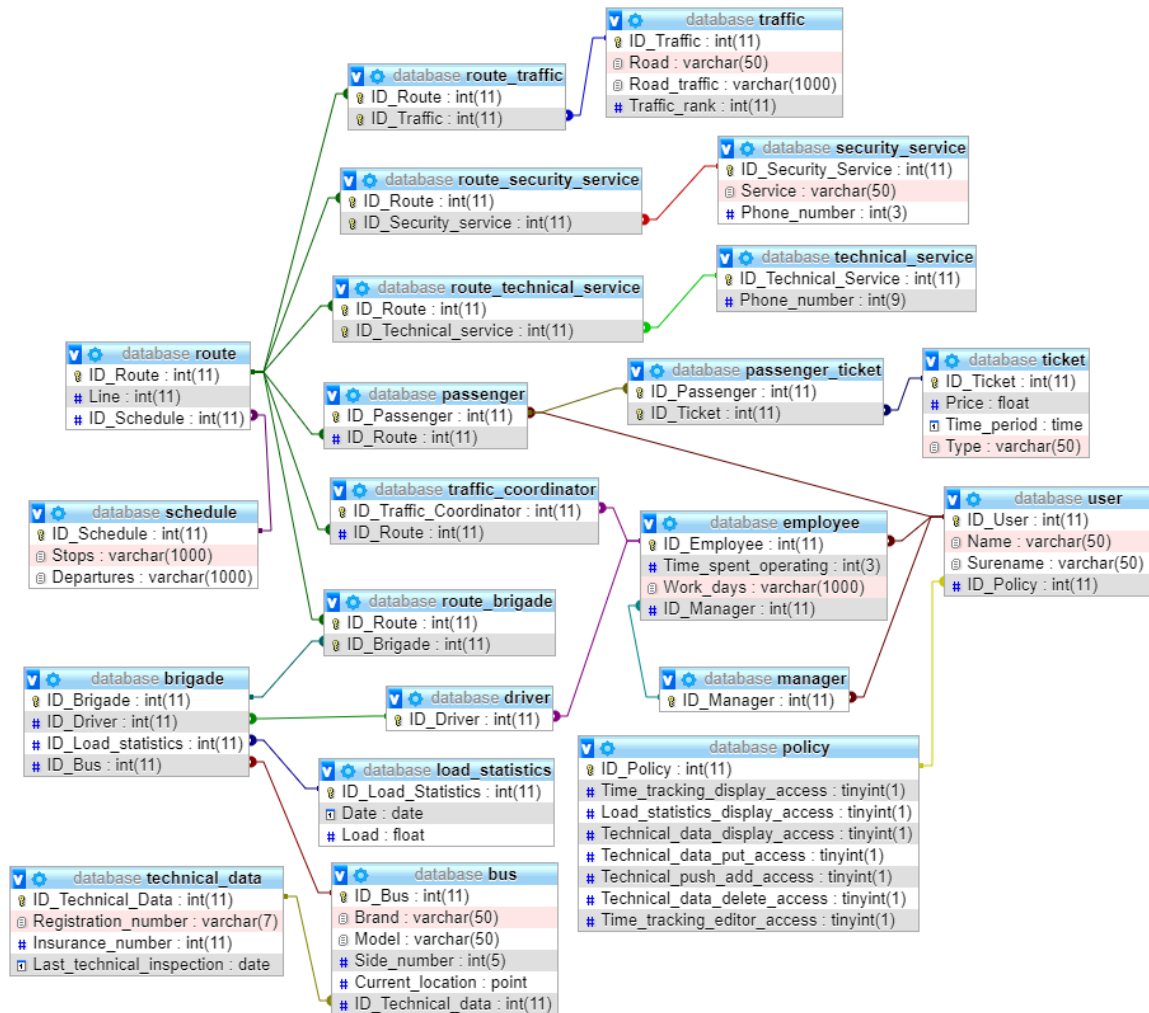


## 9.4 Charakterystyka zastosowanych wzorców projektowych

W projekcie technicznym aplikacji zostało wykorzystanych kilka wzorców projektowych ułatwiających utrzymywanie oprogramowania. Na przykładzie [pierwszego diagramu klas](#) można przedstawić wykorzystanie wzorca Singleton na klasie Repository oraz wzorca Obserwator wykorzystanego przy ViewModelu. Oprócz tego w aplikacji zostanie zastosowany wzorec Wstrzykiwanie zależności oraz zasady SOLID.

## 9.5 Projekt bazy danych

### 9.5.1 Schemat



### 9.5.2 Projekty szczegółowe tabel

Route		
ID_Route	Line	ID_Schedule
1	75	1

Schedule		
ID_Schedule	Stops	Departures
1	[ Dworzec Główny, Dworcowa, Wyszyńskiego, ... ]	[ [ 5:05, 5:20, ... ], [ 5:06, 5:21, ... ], [ 5:08, 5:23, ... ], ... ]

Load_Statistics		
ID_Load_Statistics	Date	Load
1	2021-05-18	[13.47, 28.05, 56.42, ...]

Technical_Data			
ID_Technical_Data	Registration_number	Insurance_number	Last_technical_insp ection
1	ZS 246KE	123456789...	2020-07-18

Bus					
ID_Bus	Brand	Model	Side_number	Current_location	ID_Technical_ Data
1	Solaris	Urbino 18	1850	53.418982130090704, 14.551978022038151	1

Traffic			
ID_Traffic	Road	Road_traffic	Traffic_rank
1	Kolumba	[ ..., 0, 0, 1, 2, 3, 3, 3, 3, 3, ... ]	accident

Security_Service		
ID_Security_Service	Service	Phone_number
1	Police	997

Technical_Service	
ID_Technical_Service	Phone_number
1	91 444 55 66



## 9.6 Projekt interfejsu użytkownika

### 9.6.1 Lista głównych elementów interfejsu

Główna funkcjonalność - natężenie ruchu i propozycja zmiany trasy:

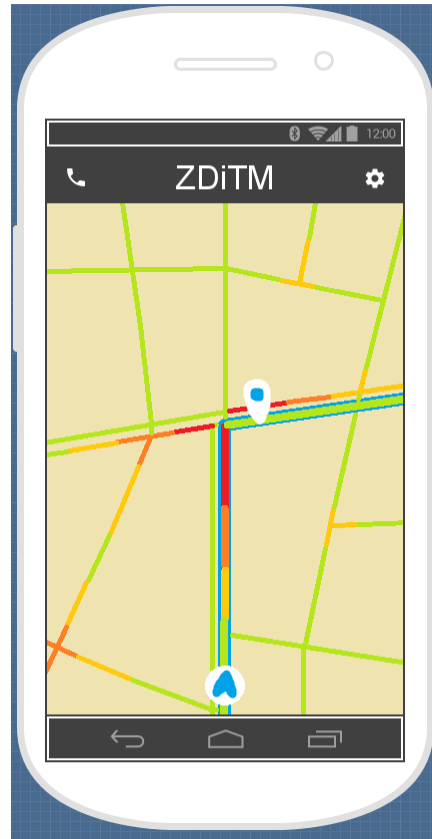
1. Splash screen
2. Ekran logowania
3. Ekran główny
4. Ekran natężenia ruchu
5. Dialog z propozycją zmiany trasy

### 9.6.2 Przejścia między głównymi elementami





→ 3. →



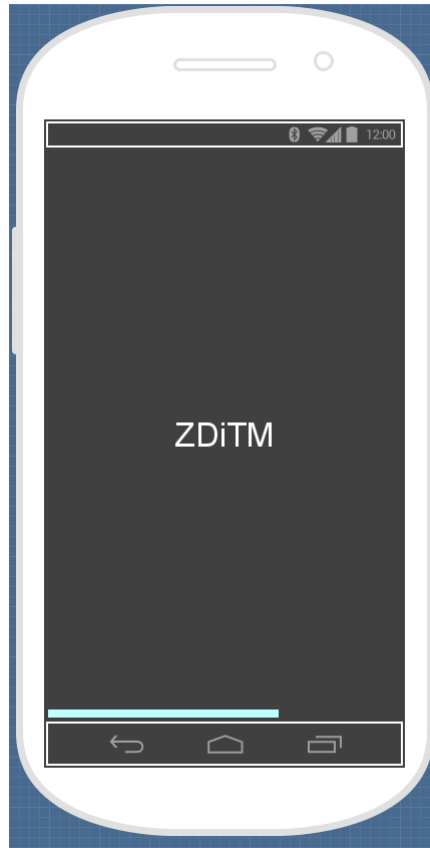
→ 4. →





### 9.6.3 Projekty szczegółowe poszczególnych elementów

#### 1. Splash screen



- Projekt graficzny:
- Opis:  
Jest to ekran powitalny widziany podczas uruchamiania aplikacji.
- Wykorzystane dane:  
Połączenie z serwerem, pobranie podstawowych danych o użytkowniku oraz lokalne ustawienia aplikacji.
- Opis działania:

Brak interakcji	Ekran widnieje do momentu pobrania potrzebnych danych do uruchomienia aplikacji, co symbolizuje progress bar na dole ekranu.
-----------------	--

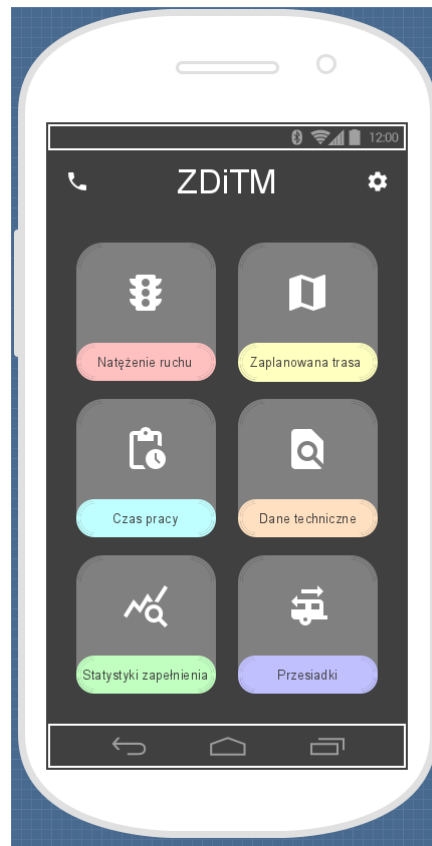
## 2. Ekran logowania



- Projekt graficzny:
- Opis:  
Ekran umożliwiający zalogowanie się do aplikacji.
- Wykorzystane dane:  
Dane o użytkowniku i jego uprawnieniach.
- Opis działania:

Interakcja z polem do wpisania loginu	Wpisany tekst oraz informacje o niepoprawnym loginie lub przekroczeniu ilości dozwolonych znaków.
Interakcja z polem do wpisania hasła	Wpisany tekst oraz informacje o niepoprawnym hasle lub przekroczeniu ilości dozwolonych znaków.
Naciśnięcie przycisku "ZALOGUJ"	Wywołanie sprawdzenia danych, jeśli niepoprawne to wyświetlenie informacji w polach tekstowych, a jeśli poprawne to przekierowanie do ekranu głównego.

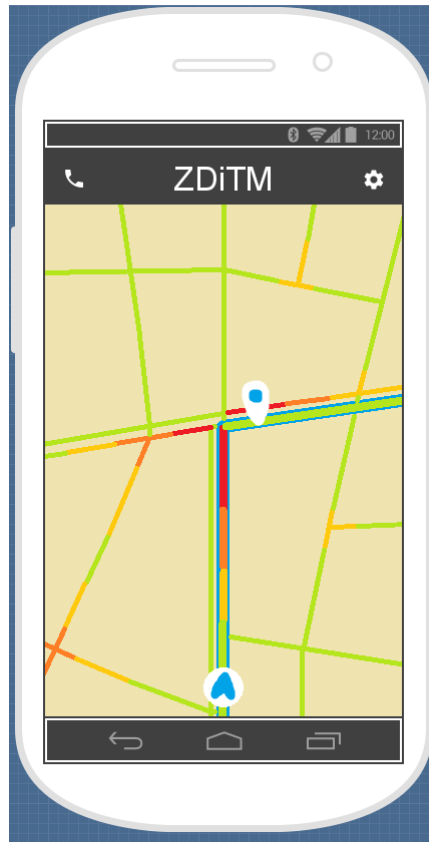
### 3. Ekran główny



- Projekt graficzny:
- Opis:  
Ekran pozwalający na przemieszczanie się pomiędzy najważniejszymi funkcjonalnościami aplikacji.
- Wykorzystane dane:  
Dane o uprawnieniach użytkownika w celu wyświetlenia odpowiednich kafelków i opcji na ekranie.
- Opis działania:

Naciśnięcie przycisku szybkiego kontaktu (ikona telefonu na app barze)	Wyświetlenie listy dostępnych służb do wyboru.
Naciśnięcie przycisku ustawień (ikona koła zębatego na app barze)	Przeniesienie do ekranu ustawień aplikacji.
Naciśnięcie wybranego kafelka	Przeniesienia do ekranu odpowiedzialnego za wybraną funkcjonalność.

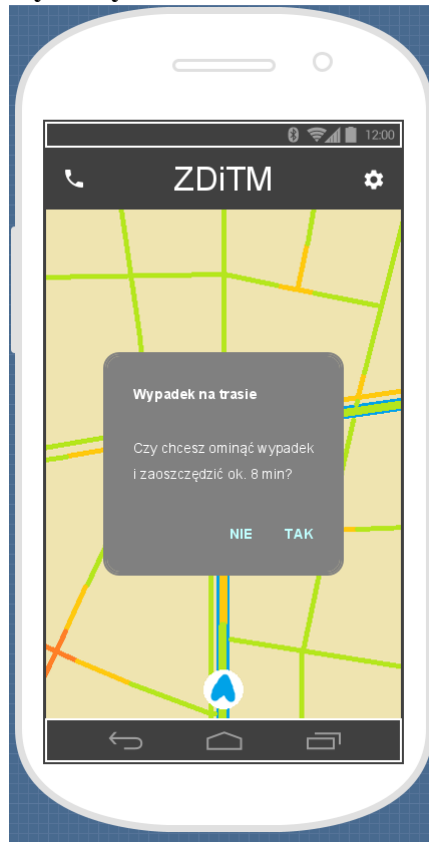
#### 4. Ekran natężenia ruchu



- Projekt graficzny:
- Opis:  
Ekran przedstawiający natężenie ruchu na trasie oraz wskazujący drogę.
- Wykorzystane dane:  
Dane o natężeniu ruchu, trasie oraz brygadzie.
- Opis działania:

Brak interakcji	Wyświetlanie natężenia ruchu oraz wskazywanie drogi.
Naciśnięcie przycisku szybkiego kontaktu (ikona telefonu na app barze)	Wyświetlenie listy dostępnych służb do wyboru.
Naciśnięcie przycisku ustawień (ikona koła zębatego na app barze)	Przeniesienie do ekranu ustawień aplikacji.

## 5. Dialog z propozycją zmiany trasy



- Projekt graficzny:
- Opis:  
Dialog wyświetlany na pierwszym planie z propozycją zmiany trasy.
- Wykorzystane dane:  
Dane o natężeniu ruchu, trasie oraz brygadzie.
- Opis działania:

Naciśnięcie przycisku “TAK”	Akceptacja propozycji zmiany trasy, zamknięcie dialogu oraz aktualizacja trasy na mapie na nowo wybraną.
Naciśnięcie przycisku “NIE” lub poza obszar dialogu	Odrzucenie propozycji zmiany trasy, zamknięcie dialogu oraz brak wpływu na aktualną trasę.

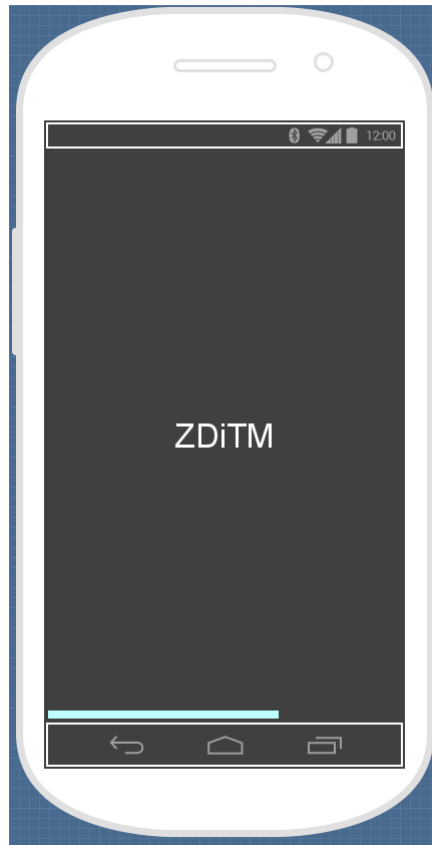
### **9.7 *Procedura wdrożenia***

Wdrożenie systemu będzie oparte na metodzie Agile. Główne etapy projektu tj. rozwój konkretnej wersji oprogramowania będą składać się z siedmiu dwutygodniowych sprintów, w trakcie których będzie miało miejsce rozwijanie oprogramowania przez deweloperów oraz testowanie przez testerów. W międzyczasie prowadzone będą rozmowy z klientem przez Product Ownera oraz UX Designera w celu ustalenia dalszej drogi rozwoju systemu. Okres ten zakończy wydanie nowej wersji oprogramowania, a następnie zbierane będą informacje zwrotne od użytkowników na przykład po wydaniu wersji beta.

## 10 Dokumentacja dla użytkownika

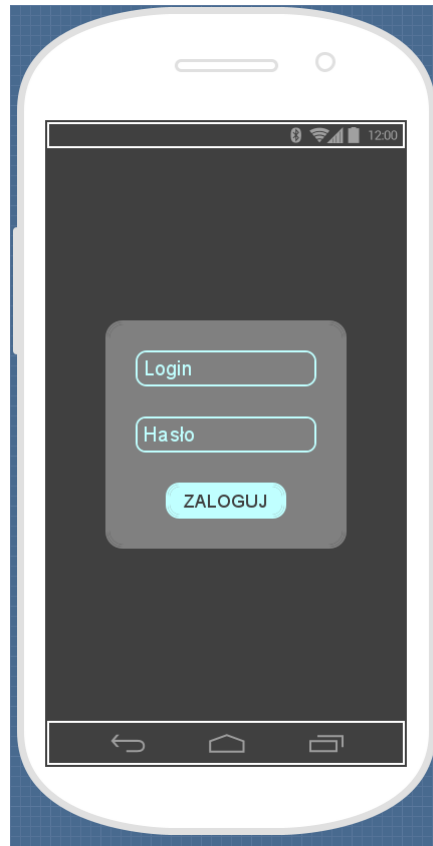
### Natężenie ruchu krok po kroku - instrukcja dla kierowcy autobusu

1. Splash screen - jest to ekran powitalny pojawiający się bezpośrednio po uruchomieniu aplikacji. Poczekaj aż dane się załadują, a następnie zostaniesz automatycznie przeniesiony do ekranu logowania. Postęp ładowania jest symbolizowany przez pasek postępu u dołu ekranu.



Podgląd ekranu:

2. Ekran logowania - jest to ekran, poprzez który możesz zalogować się na swoje własne konto założone przez administratora. Wpisz login i hasło, a następnie naciśnij przycisk "ZALOGUJ", jeśli wprowadzone zostały poprawne dane zostaniesz przeniesiony do ekranu głównego aplikacji. W przeciwnym razie zostaniesz poinformowany o błędnie wprowadzonych danych.



Podgląd ekranu:

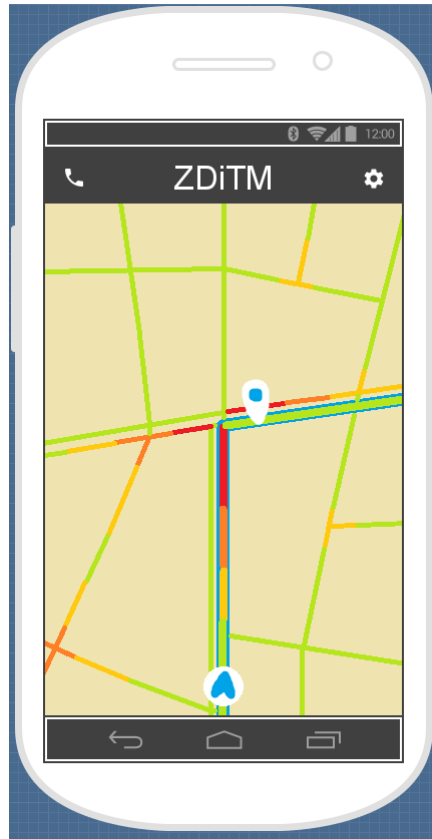


3. Ekran główny - jest to najważniejszy ekran aplikacji, to dzięki niemu możesz poruszać się pomiędzy wszystkimi funkcjonalnościami. W górnej części znajduje się pasek z dwoma ikonami. Ikona telefonu służy do szybkiego kontaktu, natomiast ikona koła zębatego przenosi do ustawień aplikacji. W głównej części znajdują się kafelki przenoszące do najważniejszych części aplikacji. Naciśnij "Natężenie ruchu", aby przenieść się do ekranu odpowiedzialnego za funkcjonalności związane z natężeniem ruchu.



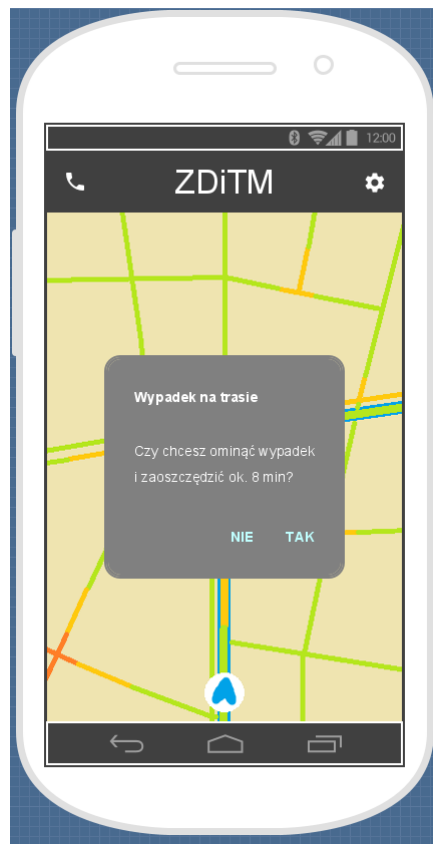
Podgląd ekranu:

4. Ekran natężenia ruchu - jest to ekran na którym widoczne jest natężenie ruchu drogowego w okolicy aktualnej lokalizacji. Oprócz tego na mapie wyznaczona jest trasa linii, a w górnej części ekranu znajduje się identyczny pasek jak na ekranie głównym.



Podgląd ekranu:

5. Dialog z propozycją zmiany trasy - jest to propozycja objazdu wyświetlająca się nad pozostałymi elementami systemu. Pojawia się samoistnie bez interakcji użytkownika w momencie, gdy system przeanalizuje trasę i znajdzie szybszy dojazd do celu. Dzieje się to na przykład w momencie wystąpienia wypadku lub dużego korku. Naciskając przycisk "TAK" zaakceptujesz zaproponowaną przez algorytm trasę, natomiast naciskając "NIE" lub obszar poza oknem dialogowym anulujesz propozycję.



Podgląd ekranu:

6. Podsumowanie:  
To wszystko co musisz wiedzieć o funkcjonalności jaką jest proponowanie nowej trasy na podstawie analizy natężenia ruchu. W razie jakichkolwiek pytań lub wątpliwości nie wahaj się zwracać do swojego administratora lub pisać bezpośrednio na [support@firma.com](mailto:support@firma.com).

## 11 Podsumowanie

### 11.1 Szczegółowe nakłady projektowe członków zespołu

	Michał Kucznerowicz	Michał Niedzielski
Wprowadzenie	80% ~ 2 h	20% ~ 1.5 h
Specyfikacja wymagań	90% ~ 8 h	10% ~ 6 h
Zarządzanie projektem	100% ~ 2 h	0% ~ 0 h
Zarządzanie ryzykiem	100% ~ 2 h	0% ~ 0 h
Zarządzanie jakością	100 % ~ 4 h	0% ~ 0 h
Projekt techniczny	100% ~ 28 h	0% ~ 0 h
Dokumentacja dla użytkownika	100% ~ 2 h	0% ~ 0 h
<b>PODSUMOWANIE</b>	<b>95%</b> 9 h - czas na zajęciach 39 h - czas poza zajęciami	<b>5%</b> 7.5 h - tylko czas na zajęciach

## **12 Inne informacje**