|  |  |
| --- | --- |
| **PROJEKT** | **symulacja siatki opóźnień pociągów  w grafie spójnym stacji docelowych** |

*Krzysztof Wiłnicki ­­– opracowanie dokumentacji  
Kacper Wilgus – pomoc w opracowaniu dokumentacji*



**DDS – delay distribution simulation**

**Spis treści**

[INICJACJA 5](#_Toc169955623)

[Założenie projektu. 5](#_Toc169955624)

[1. Opóźnienie. 5](#_Toc169955625)

[1.1. Opóźnienia losowe awaryjne. 5](#_Toc169955626)

[1.1.1. Awaria części składu. 5](#_Toc169955627)

[1.1.2. Awaria urządzeń w trasie 5](#_Toc169955628)

[1.2. Opóźnienia losowe wypadkowe. 5](#_Toc169955629)

[1.2.1. Przypadek A. 5](#_Toc169955630)

[1.3. Omówienie zależności. 6](#_Toc169955631)

[Informacje dla punktu 1.4. 6](#_Toc169955632)

[1.4. Karta charakterystyki zależności czasowych. 8](#_Toc169955633)

[1.5. Opóźnienie w kontekście *pociągu*. 9](#_Toc169955634)

[2. Pociąg. 9](#_Toc169955635)

[2.1. Kierunek poruszania. 9](#_Toc169955636)

[2.2. Przepustowość. 9](#_Toc169955637)

[2.3. Pasażerowie. 10](#_Toc169955638)

[2.4. Oznaczenie. 10](#_Toc169955639)

[2.5. Aktualne opóźnienie. 10](#_Toc169955640)

[2.6. Dodatkowe informacje. 11](#_Toc169955641)

[3. Stacja. 12](#_Toc169955642)

[3.1. Parametryzacja obiektu typu *stacja*. 12](#_Toc169955643)

[3.2. Pozostałe pola. 13](#_Toc169955644)

[4. Szlak. 13](#_Toc169955645)

[5. Uwagi dodatkowe. 13](#_Toc169955646)

[IMPLEMENTACJA 14](#_Toc169955647)

[6. Diagramy sytuacyjne. 14](#_Toc169955648)

[6.1. Diagram przypadków użycia. 14](#_Toc169955649)

[6.2. Diagram obiektów. 16](#_Toc169955650)

[7. Przygotowanie środowiska. 17](#_Toc169955651)

[8. Hierarchia klas. 18](#_Toc169955652)

[8.1. *BaseDelay*. 19](#_Toc169955653)

[8.2. *TrainFailureDelay*. 19](#_Toc169955654)

[8.3. *LineFailureDelay*. 19](#_Toc169955655)

[8.4. *RandomSituationDelay*. 19](#_Toc169955656)

[8.5. *BaseTrain*. 19](#_Toc169955657)

[8.6. *Train*. 19](#_Toc169955658)

[8.7. *PassengerTrain*. 19](#_Toc169955659)

[8.8. *FreightTrain*. 19](#_Toc169955660)

[8.9. *Initialization*. 19](#_Toc169955661)

[8.10. *ApplicationWindow*. 19](#_Toc169955662)

[8.11. *MainSceneController*. 19](#_Toc169955663)

[8.12. *EntrySceneController*. 19](#_Toc169955664)

[9. Symulacja. 20](#_Toc169955665)

[20](#_Toc169955666)

[20](#_Toc169955667)

[20](#_Toc169955668)

[20](#_Toc169955669)

[20](#_Toc169955670)

[20](#_Toc169955671)

[20](#_Toc169955672)

[20](#_Toc169955673)

[20](#_Toc169955674)

[20](#_Toc169955675)

[20](#_Toc169955676)

[20](#_Toc169955677)

[21](#_Toc169955678)

[21](#_Toc169955679)

[21](#_Toc169955680)

[21](#_Toc169955681)

[21](#_Toc169955682)

[21](#_Toc169955683)

[21](#_Toc169955684)

[21](#_Toc169955685)

[21](#_Toc169955686)

[21](#_Toc169955687)

[21](#_Toc169955688)

[21](#_Toc169955689)

**Spis rysunków**

[Rysunek 1. Wizualizacja przedstawiająca bieg pociągów w czasie rzeczywistym naniesionych na mapę obiegów stacyjnych. 10](#_Toc168867721)

[Rysunek 2. Wizualizacja przedstawiająca bieg pociągów w czasie rzeczywistym naniesionych na niepoprawnie zaprojektowaną mapę. 11](#_Toc168867722)

[Rysunek 3. Diagram przypadków użycia dla zakwalifikowanych przypadków opóźnień – obraz zminimalizowany. 13](#_Toc168867723)

[Rysunek 4. Diagram przypadków użycia dla zakwalifikowanych przypadków opóźnień – część lewa. 14](#_Toc168867724)

[Rysunek 5. Diagram przypadków użycia dla zakwalifikowanych przypadków opóźnień – część prawa. 14](#_Toc168867725)

[Rysunek 6. Diagram obiektów dla przypadkowego układu symulacyjnego – przypadek losowy AL. 15](#_Toc168867726)

[Rysunek 7. Diagram obiektów dla przypadkowego układu symulacyjnego – przypadek losowy ZWA. 15](#_Toc168867727)

[Rysunek 8. Przytoczenie potrzebnych bibliotek do poprawnego zbudowania projektu. 16](#_Toc168867728)

[Rysunek 9. Diagram klas w projekcie z widocznym podziałem na podgrupy działania. 17](#_Toc168867729)

**Tym kolorem (CZERWONYM) OZNACZONE ZOSTAŁY elementy / założenia plano-wane, JEDNOCZEŚNIE MAJĄCE NISKI PRIORYTET WYKONANIA.**

**NIE ZOSTAŁY ONE ZAKWALIFIKOWANE DO ADAPTACJI W ostatecznej WERSJI PROJEKTU!**

**może ich nie być lub nie będą IMPLEMENTOWANE w ostatecznej wersji PROJEKTU!**

# INICJACJA

## **Założenie projektu.**

Założeniem projektu jest przedstawienie skutków niedokładnego (względnie losowego) planowania ruchu pociągów, a także pokazanie złożoności opóźnień wynikających z wypadków losowych   
i ich wpływu na całą sieć połączeń.

Symulacja zakłada bardzo uproszczony model siatki połączeń składający się ze stacji początkowych   
i końcowych, linii jedno- i dwutorowych o różnych parametrach oraz składów pociągów o różnych parametrach.

## **Opóźnienie.**

Kluczowym parametrem wpływającym na płynność przeprowadzania symulacji i różnicę między wynikami rzeczywistymi a oczekiwanymi będą rodzaje opóźnień, wszystkie 10 z nich zostało przybliżone poniżej. Opóźnienia stają się tym samym głównym agentem opisywanej symulacji.

### Opóźnienia losowe awaryjne.

#### Awaria części składu.

* + - * **AL** awaria lokomotywy

*uszkodzenie hamulca pomocniczego, uszkodzenie silników tyrystorowych, uszkodzenie pantografu etc.*

* + - * **AWP** awaria wagonu poważna

*uszkodzenie przewodu pneumatycznego, uszkodzenie sprzęgu podstawowego, uszkodzenie wózka etc.*

* + - * **AWS** awaria wagonu symboliczna

*uszkodzenie węzła AC, uszkodzenie szafy automatyki etc.*

* + 1. Awaria urządzeń w trasie.
       - **ASRK** awaria urządzeń sterowania ruchem kolejowym
       - **ASEM** awaria semafora dowolnego typu

*jazda na sygnał zastępczy bądź z pominięciem syngału S1 „Stój”*

### Opóźnienia losowe wypadkowe.

#### Przypadek A.

* + - * **WCZ** wypadek z udziałem człowieka / zwierzęcia
      * **WPO** wypadek z udziałem pojazdu
      * **ZWA** złe warunki atmosferyczne
      * **PRZ** przyczyny związane z działalnością przewoźnika kolejowego
      * **ZIN** przyczyny związane z działalnością zarządcy infrastruktury

### Omówienie zależności.

Każde z opóźnień posiada swoją charakterystykę uproszczoną do 4 podstawowych składowych:

* wypadkowej czasu

*wskaźnik WYP – WYPADKOWA dla pojazdu dotkniętego   
oraz innych (zależność opisana w przykładzie nr 2 poniżej)*

* prawdopodobieństwa wystąpienia

*wskaźnik PROB*

* spadku priorytetu trasowego

*wskaźnik SPT*

* dodatkowych uwag.

UWAGA! W czasie symulacji odwoływanie pociągów przewiduje się **wyłącznie** w sytuacji przekroczenia całkowitej ilości opóźnienia dla danego pociągu **powyżej 500** **minut**, **niezależnie od długości planowej trasy** – każdy inny skład wykonuje pracę przewozową na bazie swoich dwóch stacji krańcowych: początkowej i końcowej bez możliwości ich zmiany.

**Przykład nr 1**:

Awaria lokomotywy jest jedną z poważniejszych przyczyn, taka sytuacja wymaga bowiem zastąpienia lokomotywy pierwotnej, przy tym zapewnienia nowej drużyny trakcyjnej. Optymalne rozwiązanie nakłada tym samym na skład względnie duże opóźnienie. Dodatkowo pociąg taki blokuje przez pewien czas szlak główny, generując opóźnienie innych składów.

***Koniec przykładu nr 1.***

Wszystkie opóźnienia ze względu na mechanikę działania będą rejestrowane w danych wynikowych. Dokładna informacja dotycząca mechanizmu zliczania opóźnień znajduje się   
w punkcie 5. dokumentu.

### Informacje dla punktu 1.4.

W punkcie 1.4 przygotowana została karta charakterystyki zależności czasowych dla poszczególnych typów opóźnień. Przedstawia ona wpływ danego opóźnienia na pojazd dotknięty opóźnieniem, jak i pojazdy znajdujące się na tym samym szlaku za pojazdem dotkniętym w kierunku poruszania się tego pojazdu.

Przykładowe obliczenie wypadkowej czasu danego opóźnienia zostało przedstawione poniżej:

**Przykład nr 2**:

*Pociąg Axxxxx uległ* ***awarii lokomotywy*** *z* ***prawdopodobieństwem*** *równym* ***0.08****. Podstawowy całkowity czas przejazdu ze stacji A do stacji D wyniósł*

*.*

*Opóźnienie wydłuży czas przejazdu o wypadkową*

*.*

***Symulacja zatrzymuje*** *przejazd* pociągu *do momentu odliczenia* ***równowartości WYP****. Stąd wynika,   
że całkowity czas przejazdu pociągu dotkniętego opóźnieniem wyniesie końcowo*

*.*

*Zgodnie z tabelą charakterystyki, kolejne trzy* pociągi *w tej samej relacji* **oznaczone kolejnością**  *(jeśli takowe występują) także zostaną opóźnione –* ***symulacja NIE zatrzymuje*** *ich przejazdu w kierunku stacji docelowej – opóźnienie to wyniesie*

***.***

*Stąd wynika, że całkowity czas przejazdu trzech kolejnych pociągów (k 1-3) w tej samej relacji dotkniętego opóźnieniem wyniesie końcowo*

*Dla tego typu awarii obliczyć należy kolejne (ostatnie) opóźnienia – tym razem dla kolejnych trzech* pociągów *w tej samej relacji, lecz* ***oznaczonych kolejnością***  *(jeśli takowe występują). Tak samo jak powyżej,* ***symulacja NIE zatrzymuje*** *ich przejazdu w kierunku stacji docelowej, a opóźnienie wyniesie*

***.***

*Stąd wynika, że całkowity czas przejazdu trzech kolejnych pociągów (k 1-3) w tej samej relacji dotkniętego opóźnieniem wyniesie końcowo*

***Koniec przykładu nr 2.***

Skrótowe oznaczenia dla powyższego przykładu, jak i tabeli danych w punkcie 1.4 zostały sklasyfikowane poniżej:

\* oznaczenie poważnego opóźnienia  
 czas całkowity przejazdu pociągu [w minutach] [parametr pociągu]  
*WYP* wypadkowa czasu opóźnienia  
*PD* pojazd dotknięty  
*WYPPD* oznaczenie zastępcze dla wyrażenia «WYPADKOWA CZ. OPÓŹ. dla PD»  
n, m numeruje kolejne *pociągi* za PD  
*min max* minimalna i maksymalna wartość wskaźnika  
*mini* minimalna wartość priorytetu

Wszystkie ułamki widoczne w tabeli zaokrąglane są w górę sufitowo (tak jak wskazane w przykładzie nr 2) do najbliższej liczby całkowitej.

### Karta charakterystyki zależności czasowych.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **WYPADKOWA** | | | **PROB** | | **SPT** | *uwagi* |
| *WYPPD* | *inne (w tym samym kierunku)* | | *min* | *max* |
| **AL\*** |  | k 1-3 |  | 0.05 | 0.1 | ***mini*** | dla linii jednotorowych, wypadkową należy  przemnożyć przez **1,5** |
| k 4-6 |  |
| **AWP** |  | k 1-2 |  | 0.02 | 0.04 | **-1** | dla przepustowości  klasy poniżej **II**, wypadkową należy przemnożyć przez **1,1** |
| k 3-4 |  |
| **AWS** |  | k 1-2 |  | 0.15 | 0.25 |  |  |
| k 3 |  |
| **ASRK** |  | k 1 |  | 0.3 | 0.35 | **-1** |  |
| k 2-3 |  |
| **ASEM** |  | k 1 |  | 0.45 | 0.5 |  |  |
| k 2 |  |
| **WCZ\*** |  | k 1-4 |  | 0.01 | 0.05 | ***mini*** | dla linii jednotorowych, wypadkową należy  przemnożyć przez 2 |
| k 5-8 |  |
| **WPO\*** |  | k 1-3 |  | 0.005 | 0.01 | ***mini*** | dla przepustowości  klasy poniżej **II**, wypadkową należy przemnożyć przez **1,3** |
| k 4-6 |  |
| k 7-8 |  |
| **ZWA** |  | k 1-3 |  | 0.2 | 0.3 |  |  |
| **PRZ** |  | k 1-2 |  | 0.75 | 0.8 |  |  |
| k 3 |  |
| **ZIN** |  | k 1-2 |  | 0.75 | 0.8 |  |  |
| k 3 |  |

Tabela 1. Przedstawienie karty charakterystyki zależności czasowych.

### Opóźnienie w kontekście *pociągu*.

Jako iż w kolejnym punkcie omówieniu ulega obiekt typu *pociąg,* należy przedstawić wszystkie możliwe sposoby zobrazowania zebranego opóźnienia przez wszystkie tego typu obiekty   
w toku trwania symulacji.

Opóźnienie w zamyśle obiektu typu *pociąg* dotyczy obliczania dodatkowego czasu przejazdu względem początkowo ustalonego całkowitego czasu przejazdu Na koniec symulacji opóźnienie dla każdego *pociągu* przedstawiane jest na cztery sposoby:

* liczbowa wartość opóźnienia w minutach (oraz jej stopień zaawansowania),
* całkowity czas przejazdu po uwzględnieniu opóźnienia,
* różnica względna wyrażana w procencie
* odchylenie od średniego opóźnienia wszystkich pociągów na koniec symulacji.

## **Pociąg.**

Obiektami nominalizowalnymi przez opóźnienia-agentów, poruszającymi się po wirtualnej mapie obiegów stacyjnych (z założenia początkowego) są *pociągi*. Każdy z tego typu obiektów posiada następujący zestaw parametrów wpływający na płynność jazdy, efektywność rozkładową oraz ostateczną moc opóźnienia:

### Kierunek poruszania.

Każdy *pociąg* bezwzględnie potrzebuje parametru kierunku poruszania ze względu   
na determinację opóźnień kolejnych jednostek powodowanych przez niego samego. Jeżeli   
dwa pociągi znajdują się w na tej samej linii w tej samej trasie, będą na siebie oddziaływać liniowo z każdym przeskokiem czasowym symulacji. Kierunek określany jest na zasadzie rachunku kodów stacji – jeśli

*gdzie*

to znaczy, że pociągi poruszają się **w tej samej relacji** – na tym samym szlaku w tym samym kierunku.

**WAŻNE! Graf wynikowy zostaje stworzony w taki sposób, aby żadna z relacji nie posiadała tej samej wartości identyfikacyjnej (aby działanie wartości bezwzględnej z różnicy dwóch kodów stacji zwracało taki sam wynik).**

### Przepustowość.

Każdy *pociąg* symbolizuje liczbowo ilość przystanków pośrednich (niewliczanych do danych symulacyjnych) oraz średni czas postoju na takowych. Dane te w pierwotnej wersji wprowadzane są losowo, a na ich podstawie wyliczana jest względna przepustowość końca szlaku (ilość kilometrów pozostająca za *pociągiem* na szlaku wg parametryzacji priorytetowej) wg poniższego wzoru:

*gdzie*

Ilość kilometrów konwertowana jest następnie na wektor odległościowy za pojazdem wstrzymujący kolejny pojazd w kolei od najechania nań. Przepustowość ostatecznie   
nie wpływa na zmianę czasu opóźnienia obiektu poszkodowanego, a raczej obiektów nieposzkodowanych.

### Pasażerowie.

Każdy *pociąg* posiada 3 parametry opisujące jego nabytą frekwencję. Są to:

* średnia ilość pasażerów – wyznaczana losowo dla każdego generowanego obiektu
* maksymalna pojemność (wytyczona) – jako parametr domyślny ładunku pasażerskiego
* frekwencja względna wyznaczana wg poniższego wzoru:

*gdzie*

Dodatkowo, w oknie dialogowym dodatkowej informacji (punkt 2.4, rysunek 1.) pokazywana będzie symboliczna *gwiazdkowana* ocena ruchu pociągu (wyliczana prostym algorytmem zmiany ze względu na nabyte opóźnienia – ilość i łączny czas niedogodności.

### Oznaczenie.

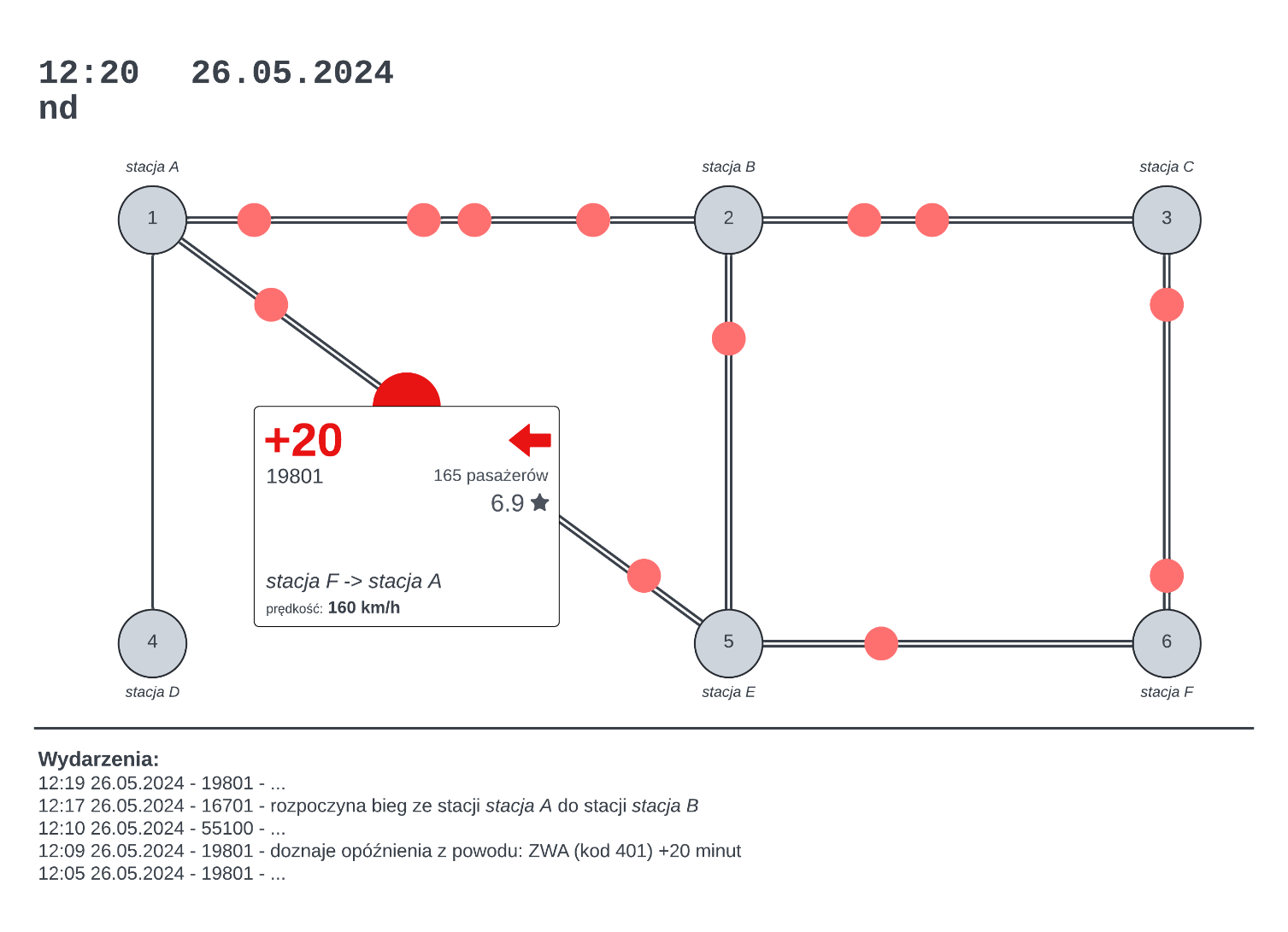
Oznaczenie *pociągu* opierać się będzie na symbolicznym przyporządkowaniu go do kategorii stanowiącej charakterystykę przewozową. Nazwy i skrótowce komunikacyjne nadawane będą w sposób określony realnymi wytycznymi planowania ruchu przewozowego wg *Instrukcji   
o prowadzeniu ruchu pociągowego* Ir-1 (PKP PLK SA). Nazwa rozkładowa pociągu składa się w takim wypadku z 2 członów alfanumerycznych. Pierwszy człon oznacza typ handlowy   
i przewozowy pociągu (przykładowo *pociąg towarowy* bądź *pociąg pasażerski* – zgodnie   
z wymagani). Drugi człon odpowiada 5-cyfrowemu numerowi handlowemu – dokładne złożenie numeru opisane jest wg wspomnianej *Instrukcji* Ir-1.

**Oznaczenia te nie będą wykorzystywane w docelowej logice symulacyjnej, a jedynie pomogą w odróżnieniu konkretnych obiektów!**

### Aktualne opóźnienie.

Opóźnienie zebrane w trakcie biegu *pociągu* zgodnie z zasadami określonymi w punkcie 1.4 dokumentu będzie wyświetlane w czasie rzeczywistym symulacji w sposób wizualnie zrozumiały.

*Proces identyfikacji obrazowej nie został dotychczas określony. W momencie aktualizacji symulacji znany będzie także powód opóźnienia wstrzymujący* pociąg *od ruchu.*

Prototyp wizualizacji obrazującej ww. zasady symulacji wygląda następująco:

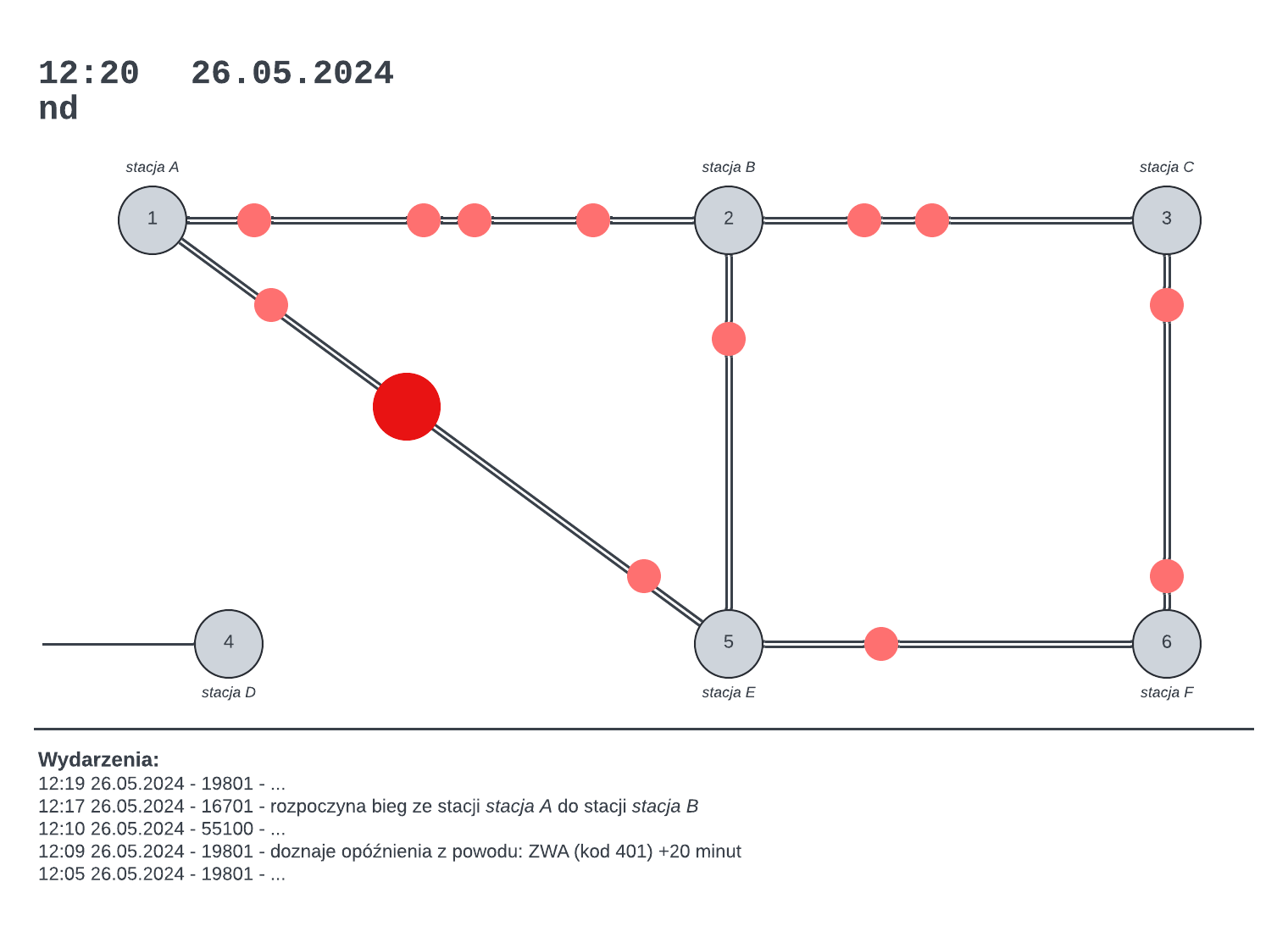
Rysunek 1. Wizualizacja przedstawiająca bieg pociągów w czasie rzeczywistym naniesionych na mapę obiegów stacyjnych.

Na rysunku zauważamy także element charakterystyczny dla *pociągu* – okno dodatkowej informacji. Dedykowane dla każdego *pociągu*, generowane w czasie rzeczywistym.

### Dodatkowe informacje.

Dodatkowe informacje obejmują stację początkową biegu, stację końcową, aktualną prędkość akutalizowaną schematycznie z każdą sekundą symulacji, aktualny kierunek biegu pociągu oraz wskaźnik PRI – wskazanie względne priorytetu w momencie aktualizacji.

## **Stacja.**

Pojęcie grafu spójnego *stacji* docelowych oznacza jednoznacznie sieć, w której wierzchołkami są *stacje*, z żadnej z których nie wybiega półprosta do stacji niewidocznej na schemacie, a nieprawidłowym   
jest także pozostawienie *stacji* bez jakiegokolwiek połączenia (wierzchołek izolowany). Poprawna sytuacja przedstawiona została na rysunku 1. w punkcie 2.4. Poniżej zobrazowana jest przykładowa niepoprawna sytuacja: niepoprawne:

Rysunek 2. Wizualizacja przedstawiająca bieg pociągów w czasie rzeczywistym naniesionych na niepoprawnie zaprojektowaną mapę.

### Parametryzacja obiektu typu *stacja*.

Parametryzacja stacji opiera się o 4 główne składniki:

* + CODE kod stacji w zapisie liczbowym (),
  + S ilość pociągów obecnych w momencie aktualizacji

(z założeniem )

* + maksymalna pojemność stacji na godzinę jako ilość cykli wymiany

pasażerskiej przy założeniu pełnego obłożenia krawędzi peronowych oraz 2-minutowego czasu wymiany krawędziowej,

wyznaczana za pomocą wzoru

*gdzie*

Parametry te jednak nie są – na chwilę obecną – parametrami aktywnymi w sensie symulacji. Prawidłowa implementacja przewidziana jest w momencie utworzenia poprawnego algorytmu wzajemnego śledzenia między *pociągami*.

### Pozostałe pola.

Na chwilę obecną przewiduje się wprowadzenie pola nazwy stacji dla poprawnego odróżniania kolejnych obiektów w toku trwania symulacji w kontekście użytkownika.

## **Szlak.**

Wszystkie linie kolejowe zaznaczone na wizualizacji symulacyjnej są zbudowane w ten sam sposób   
z odróżnieniem linii jedno- i dwutorowych. Pierwsze zaznaczane są jedną linią odcinkową między kolejnymi stacjami – drugie kreślone zostają poprzez dwa równoległe odcinki międzystacyjne. Rozróżnienie widać to na rysunkach 1. do 3.

Ponadto, linie kolejowe posiadają własną klasyfikację przepustowości, której klasami interfejsu jest priorytetowość przejazdów:

|  |  |  |
| --- | --- | --- |
| priorytet | sytuacja użytkowania | opis szczegółowy |
| S | priorytet bezklasowy | priorytetyzuje się tylko pociągi o Vmax > 160 km/h (bez kategoryzacji opóźnień), reszta pojazdów jedzie zgodnie z kolejnością pozarozkładową |
| Y | priorytet klasowy zerowy | priorytetyzuje się tylko pociągi o Vmax > 160 km/h (bez kategoryzacji opóźnień), reszta pojazdów jedzie zgodnie z kolejnością pozarozkładową |
| YA | priorytet klasowy A | priorytetyzuje się pociągi o Vmax > 120 km/h i pociągi nieopóźnione, reszta pojazdów jedzie zgodnie z kolejnością pozarozkładową |
| YD | priorytet klasowy D | priorytetyzuje się pociągi o Vmax > 100 km/h i pociągi nieopóźnione, reszta pojazdów jedzie zgodnie z kolejnością pozarozkładową |
| WW | status nadzwyczajny | priorytetyzuje się tylko pociągi o statusie nadzwyczajnym i Vmax > 100 km/h, reszta pojazdów jedzie zgodnie z kolejnością pozarozkładową  (bez uwzględnienia opóźnień) |
| Z | priorytet bezklasowy awaryjny | używany w przypadku awarii urządzeń SRK |

Klasyfikacja priorytetowa nie będzie używana w pierwotnie zamówionej wersji symulacji.

## **Uwagi dodatkowe.**

Opóźnienia rejestrowane są w oparciu o ilość ich wystąpień – zasada doliczania do względnej reprezentacji danych wygląda następująco:

do końcowego zestawienia danych zliczane są jedynie opóźnienia powyżej 5 minut wynikowych (dodanych do całkowitego planowego czasu przejazdu pociągu). **Wynika to z oficjalnej klasyfikacji opóźnień – wg raportów m.in. Urzędu Transportu Kolejowego, większość przewoźników nie wlicza do swoich klasyfikacji obsunięć czasowych względem rozkładu jazdy mniejszych od 5 minut.**   
Tym samym powyższa teoria symulacji również takie zachowanie uwzględnia.

# IMPLEMENTACJA

Wybranym językiem programowania (platformą wykonawczą) jest język **Java (wersja stabilna jdk 19.0.1)** z wykorzystaniem biblioteki **JavaFX (wersja stabilna 22.0.1)**. Narzędziem umożliwiającym automatyczne budowanie oprogramowania i zarządzanie zależnościami w projekcie obrany został **Gradle (wersja stabilna 8.2)**. Wykaz dodatkowych zależności, importowanych bibliotek etc. znajduje się w dalszej części dokumentacji.

Przed przystąpieniem do właściwej implementacji w ww. wybranym języku programowania, przygotować należy dokumentację wykonawczą. W kolejnych punktach przedstawiona zostanie   
z wykorzystaniem realnego planu realizacji, wizualizacji planowego przebiegu symulacji i końcowych wyników przeprowadzonej symulacji.

Opisy rozwiązań zawarte będą w plikach źródłowych projektu w formie proponowanej przez typ komentarza **Javadoc**. Odnośniki do opisów będą zarówno interaktywnymi linkami do strony zewnętrznej, zrzutami ekranu ze środowiska programistycznego, jak i wycinkami kodu.

## **Diagramy sytuacyjne.**

### Diagram przypadków użycia.

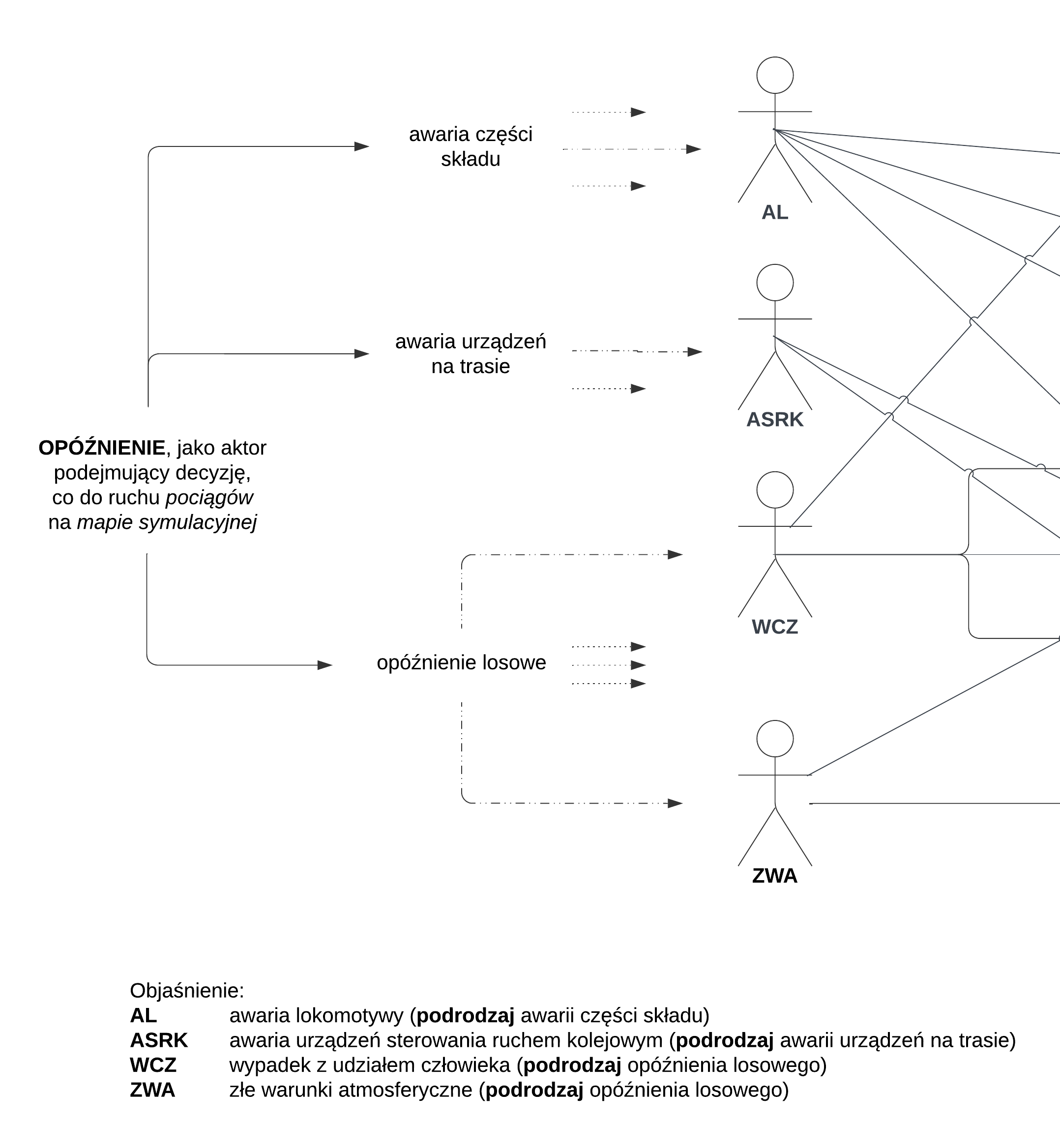
Diagram ten przedstawia funkcjonalność systemu zaprezentowanego teoretycznie w fazie **INICJACJI** wraz z jego otoczeniem. Pozwala on na graficzne zaprezentowanie własności systemu tak, jak są one widziane po stronie użytkownika aplikacji symulacyjnej. Poniżej pokazujemy widok ogólny:

Obraz zawierający tekst, diagram, linia, Plan

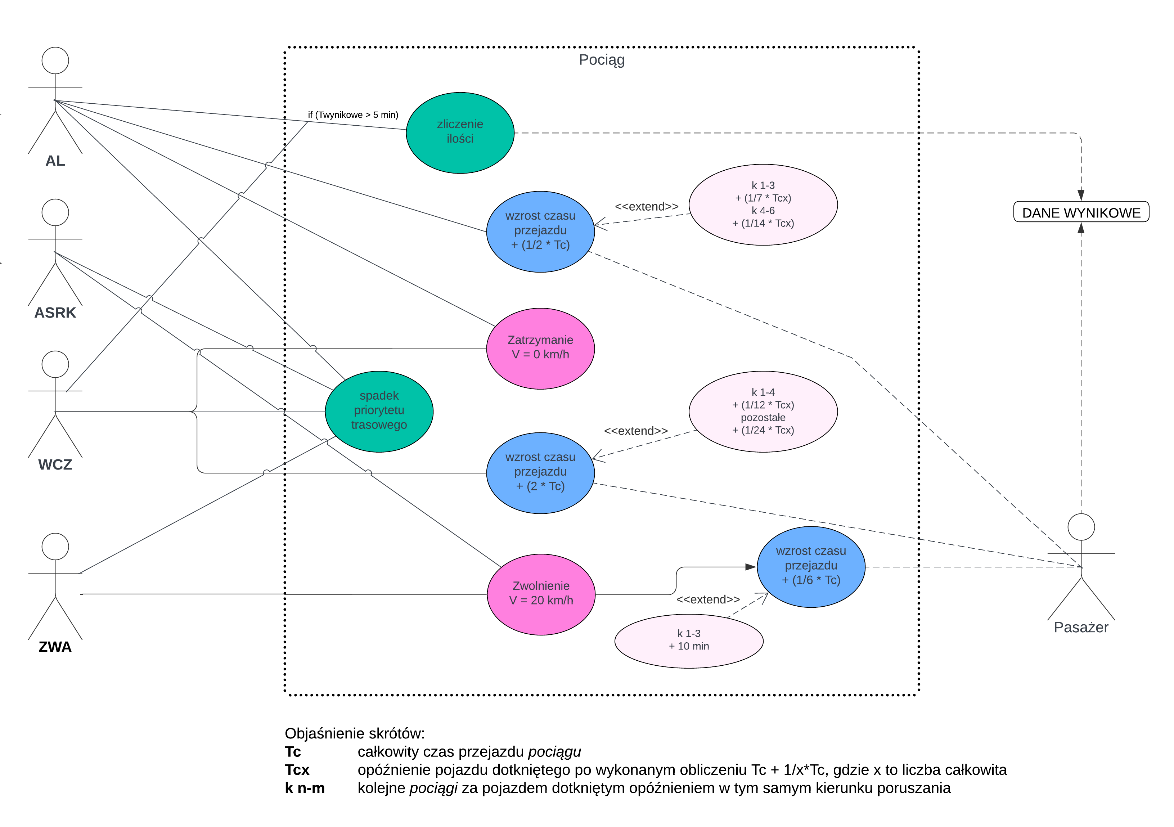
Opis wygenerowany automatycznie

Rysunek 3. Diagram przypadków użycia dla zakwalifikowanych przypadków opóźnień – obraz zminimalizowany.

Na kolejnej stronie znajdują się przybliżone części powyższego diagramu przypadków użycia – kolejno strony: lewa i prawa.



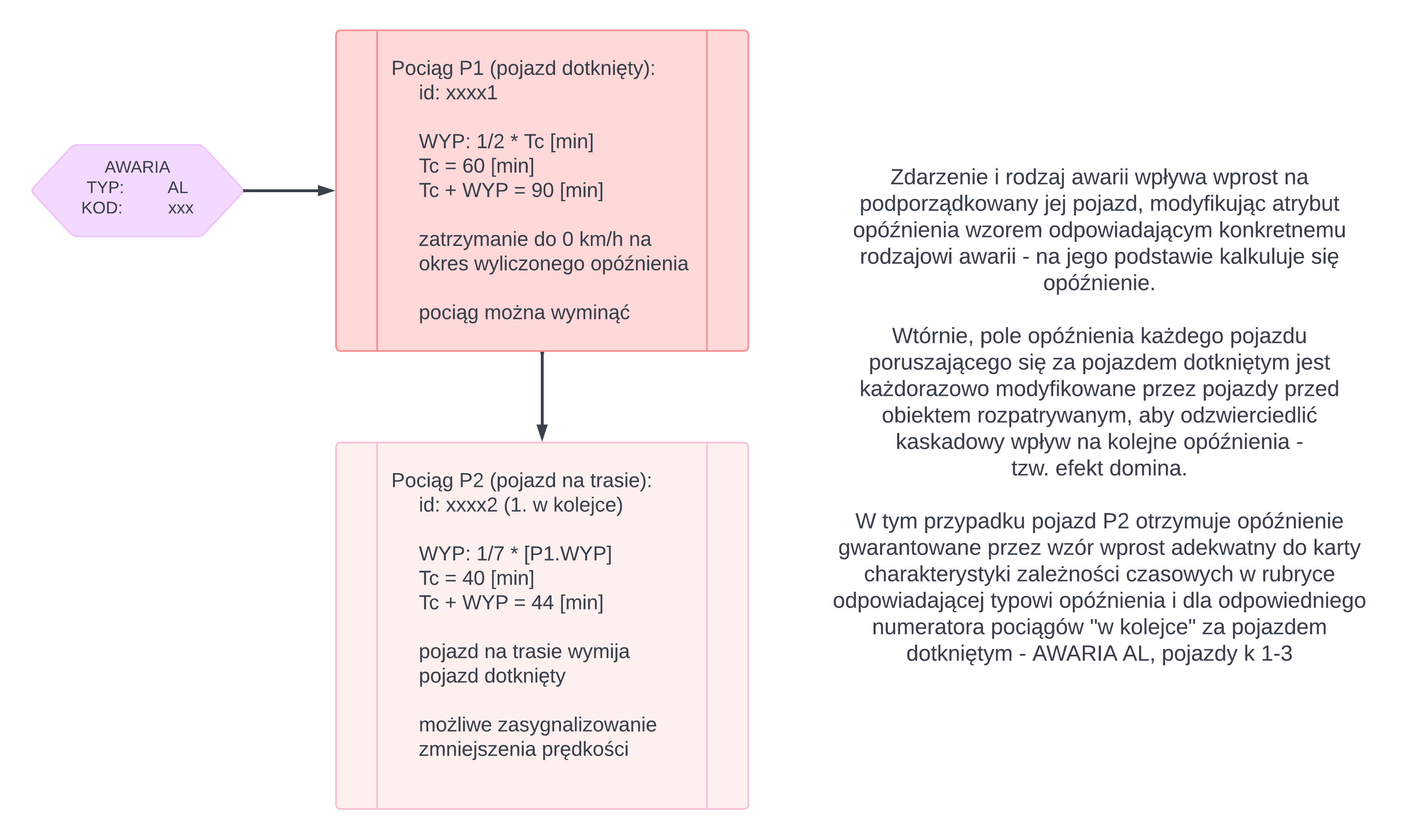
Rysunek 4. Diagram przypadków użycia dla zakwalifikowanych przypadków opóźnień – część lewa.



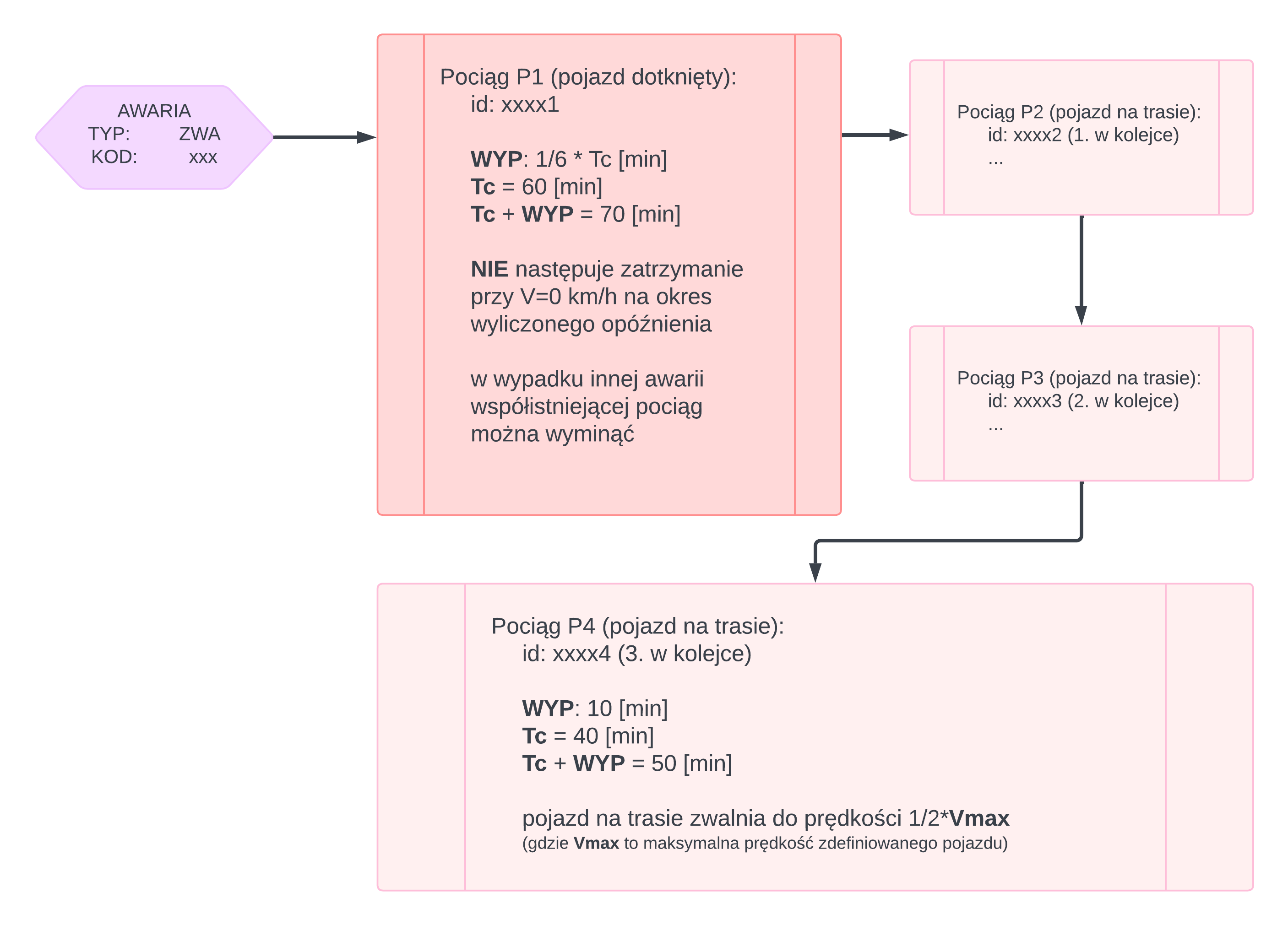
Rysunek 5. Diagram przypadków użycia dla zakwalifikowanych przypadków opóźnień – część prawa.

### Diagram obiektów.

Ten typ diagramu jest z kolei wizualizacją hipotetycznego stanu systemu podczas jego działania. Służy do tworzenia przykładów pomagających zrozumieć diagram klas a przede wszystkim powiązań w nim występujących. Poniższe przedstawiają sytuację z **zatrzymania klatki w toku symulacji dla dwóch (lub więcej) *pociągów*** – jednego typu PD i innych w tej samej relacji znajdującego się za nim.



Rysunek 6. Diagram obiektów dla przypadkowego układu symulacyjnego – przypadek losowy AL.



Rysunek 7. Diagram obiektów dla przypadkowego układu symulacyjnego – przypadek losowy ZWA.

## **Przygotowanie środowiska.**

Projekt – tak jak zostało to wspomniane w początku rozdziału ***IMPLEMENTACJA*** *–* zbudowany został przy wykorzystaniu narzędzia Gradle. Poza plikami i katalogami potrzebnymi do prawidłowego budowania i uruchamiania projektu z poziomu narzędzi wbudowanych, w folderze ***root***projektu (katalogu, który znajduje się na szczycie drzewka katalogów) znajdują się ponadto **inne foldery i pliki**. Ważniejsze z nich zostały wymienione poniżej:

* (folder) **docs** = tutaj umiejscowiona jest dokumentacja, jej poszczególne elementy graficzne oraz podstawa przygotowania emdletu README.md;
* (folder) **src** – jest topodstawa drzewa katalogów z kodem źródłowym i zależnościami (dodatkowymi plikami obrazów / pliki z rozszerzeniem FXML) potrzebnymi do zbudowania   
  i uruchomienia końcowej aplikacji;
* (plik) **build.gradle** – plik ten przechowuje informacje związane z dołączaniem paczek zależności, komunikacją między serwerem wymiany plików **Maven,** a także informacje (m.in. o głównej klasie projektu) potrzebne do zainicjalizowania projektu.

Ze względu na specyfikę pracy w środowisku Gradle, ważne było **odpowiednie skonfigurowanie biblioteki JavaFX**. Dzięki temu każdy następujący po sobie etapie nie jest obarczony ryzykiem krytycznych błędów inicjalizowania projektu, czy też możliwe jest skutecznie efektywne budowanie osobnej gałęzi konfiguracji – pliku .jar.

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

Rysunek 8. Przytoczenie potrzebnych bibliotek do poprawnego zbudowania projektu.

Na powyższym rysunku zauważyć możemy dwa wyróżniające się typy bibliotek: *junit-jupiter* oraz *openjfx-javafx*. Aby skrótowo opisać obie biblioteki posłużę się odnośnikami do już istniejących, dokładnych opisów:

* *openjfx-javafx* – biblioteka JavaFX v19.0.2 – [link](https://openjfx.io/javadoc/19/)
* *junit-jupiter* – biblioteka potrzebna do wykonywania testów jednostkowych – [link](https://junit.org/junit5/docs/current/api/)

Obraz zawierający zrzut ekranu, tekst, Czcionka, linia

Opis wygenerowany automatycznie

Rysunek 9. Diagram modułów zależności potrzebnych do poprawnego zbudowania projektu.

## **Hierarchia klas.**

Projekt dzieli się abstrakcyjnie na dwie podgrupy klas – klasy odpowiadające za działanie części logicznej oraz klasy odpowiedzialne za UI (*user interface* – prezentację wizualną). Na poniższym diagramie zostały one oddzielone czerwoną przerywaną linią, przy czym **klasa *Initialization* stanowi pewnego rodzaju *middleware[[1]](#footnote-1)* dla obu podgrup klas**.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Rysunek 10. Diagram klas w projekcie[[2]](#footnote-2) z widocznym podziałem na podgrupy działania.

Działanie każdej z klas omówione zostało dzięki wygenerowaniu bazy wiedzy wg protokołu Javadoc. Poniżej znajdują się przykładowe opisy dla kilku wybranych klas.

## **Symulacja.**

Poniżej zostaną przedstawione wyniki symulacji przeprowadzone w formie testów szczegółowych konfiguracji. Każde dane przytoczone w punkcie 9. opisane będą w następujący sposób:

gdzie k – numer porządkowy przeprowadzonej symulacji,  
m – współczynnik wariacji generowanych wartości losowych dla opóźnień ,  
v – ilość pociągów w symulacji.

***UWAGA!*** *Współczynnik m symbolizuje grupę czynników niezależnych takich, jak*

***dostępność drużyn trakcyjnych****,* ***dostępność zapasowych pojazdów trakcyjnych******efektywność pracy przewozowej*** *czy* ***likwidowania skutków awarii****.*

## 

abc

## 

abc

## 

abc

## 

abc

## 

abc

## 

abc

## 

abc

## 

abc

## 

abc

## 

abc

## 

abc

## 

1. middleware (ang. oprogramowanie pośredniczące) – rodzaj oprogramowania (w naszym przypadku zestaw metod jednej klasy) umożliwiający komunikację pomiędzy różnymi systemami (podgrupami klas) [↑](#footnote-ref-1)
2. zgodny ze stanem z dnia 22 czerwca 2024 r. [↑](#footnote-ref-2)