

# Dokumentacja funkcjonalna

Kinga Kawczyńska, Volodymyr Boiko, Swidziński Michał

1 kwietnia 2021

Metryka dokumentu				
Projekt	PZ_2021	Firma	Zespół Kingi	
Nazwa	Dokumentacja funkcjonalna			
Temat	Dokumentacja funkcjonalna projektu zespołowego			
Autorzy	Kinga Kawczyńska, Volodymyr Boiko, Swidziński Michał			
Plik	documentation_functional.pdf			
Nr. wersji	1.15	Status	finalny	Data sporządzenia 25.03.2021
Streszczenie	Ten dokument zawiera opis problemu, wymagania techniczne oraz opis użycia tworzonej aplikacji.			
Zatwierdził	Kinga Kawczyńska		Data ost. modyfikacji	31.03.2021

---

## Treść

<b>1</b>	<b>Opis problemu</b>	<b>3</b>
1.1	Sformułowanie słowne . . . . .	3
1.2	Sformułowanie formalne . . . . .	3
<b>2</b>	<b>Wymagania techniczne</b>	<b>4</b>
2.1	Wymagania funkcjonalne . . . . .	4
2.2	Wymagania niefunkcjonalne . . . . .	5
<b>3</b>	<b>Przypadki użycia oraz sytuacje błędne</b>	<b>5</b>
3.1	Przypadki użycia . . . . .	5
3.2	Sytuacje błędne . . . . .	6
<b>4</b>	<b>Zmiana harmonogramu</b>	<b>9</b>
<b>5</b>	<b>Historia zmian</b>	<b>10</b>

## Streszczenie

Ten dokument zawiera opis problemu, wymagani funkcjonalne i niefunkcjonalne oraz opis użycia tworzonej aplikacji wraz z opisem sytuacji błędnych.

Rozdział 1 zawiera opis problemu słowny, który jest potem przekształcony na język matematyczny.

Rozdział 2 uwzględnia opisany problem i formułuje wymagania do danych wejściowych oraz do samej aplikacji.

Rozdział 3 zawiera opis przypadków użycia wraz z *use case* diagramem. W podrozdziale drugim są opisane sytuacje błędne użycia aplikacji wraz z ilustrującymi przykładami.

Rozdział 4 uwzględnia zmiany w harmonogramie projektu oraz podziale pracy między członkami zespołu.

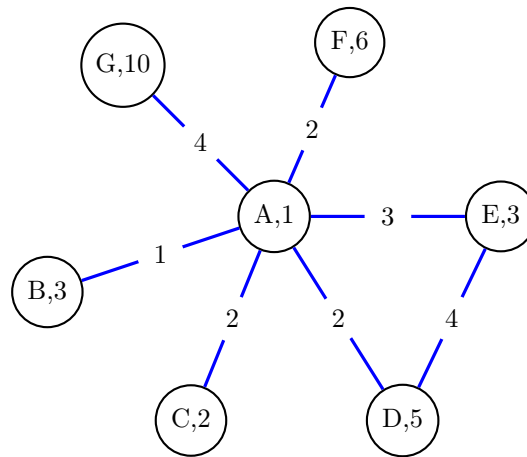
---

# 1 Opis problemu

## 1.1 Sformułowanie słowne

Wędrowny sprzedawca sprzedaje produkty w różnych miastach, pracę może zacząć i zakończyć w dowolnym mieście. Dla każdego miasta zna ilość produktów, które może tam sprzedać. Wie, jak długo trwa podróż dla każdej drogi. Sprzedawca ma ograniczoną ilość czasu.

Trzeba sprzedać jak najwięcej produktów przy ograniczonym czasie, to jest znaleźć trasę, na którą starczy czasu i ilość sprzedanych produktów będzie największa.



Rysunek 1: Dane wejściowe

**Przykład 1.** Sprzedawca ma 12 godzin. W najlepszym przypadku może sprzedać 25 produktów, idąc drogą (G, A, B, A, F, A, D).

Ponadto wiemy, że drogi nie mają skrzyżowań poza miastami oraz że dane wejściowe muszą być zgodne z rzeczywistością.

## 1.2 Sformułowanie formalne

**Definicja 1.** Mapą nazywamy parę  $\{V, E\}$ , złożoną ze zbioru miast  $V$  oraz zbioru dróg między miastami  $E$ . Każdemu miastu  $v_i$  jest przyporządkowana liczba produktów do sprzedaży  $p_i$ , oraz każdej drodze  $v_i v_j$  jest przyporządkowany czas podróży  $t_{ij}$ .

**Definicja 2.** Drogą nazywamy skończony ciąg miast  $(v_{i_1}, v_{i_2}, \dots, v_{i_n})$  taki, że dla dowolnych miast  $v_{i_k}$  oraz  $v_{i_{k+1}}$  istnieje droga, łącząca ich. Zbiór wszystkich dróg dla podanej mapy  $(V, E)$  będziemy oznaczali przez  $\mathcal{G}(V, E)$ .

---

**Definicja 3.** Czasem drogi  $g = (v_{i_1}, v_{i_2}, \dots, v_{i_n})$  nazywamy liczbę

$$T(g) = \sum_{1 \leq i \leq n-1} t_{i(i+1)}.$$

**Definicja 4.** Zyskiem drogi  $g = (v_{i_1}, v_{i_2}, \dots, v_{i_n})$  nazywamy liczbę

$$P(g) = \sum_{1 \leq k \leq n} p_{i_k} [v_{i_k} \notin (v_{i_1}, \dots, v_{i_{k-1}})].$$

**Uwaga 5.** Problem wędrującego sprzedawcy dla podanej mapy oraz czasie  $t_0$  sprowadza się do znalezienia drogi  $g \in \mathcal{G}(V, E)$  dla podanej mapy, spełniającej założenia:

$$T(g) \leq t_0$$

$$\forall h \in \mathcal{G}(V, E) [T(h) \leq t_0 \Rightarrow P(h) \leq P(g)]$$

Taką drogę nazywamy *rozwiązaniem* mapy  $(V, E)$ .

## 2 Wymagania techniczne

Poniższy rozdział opisuje wymagania funkcjonalne oraz niefunkcjonalne do tworzonej aplikacji. Sformułowanie tych wymagań pozwoli na efektywne tworzenie modelu projektu oraz testowanie.

### 2.1 Wymagania funkcjonalne

1. Aplikacja może wczytywać listę miast i listę dróg bez rozwiązania w formacie *.csv* i wyświetlać je jako grafy ważone.
2. Aplikacja może wczytywać listę miast i listę dróg wraz z rozwiązaniem w formacie *.csv* i wyświetlać je jako graf ważony wraz z rozwiązaniem z pliku.
3. Aplikacja może wczytywać listę miast i listę dróg w formacie *.csv* i wyświetlać je jako grafy ważone wraz z drogą optymalną. Droga optymalna jest zapisana do pliku *.csv*.
4. Aplikacja odrzuca mapy i rozwiązania niepoprawne, w tym:
  - mapy, w których istnieją drogi przecinające się
  - mapy puste
  - mapy, w których istnieją miasta o tych samych współrzędnych
  - mapy, w których istnieją drogi o niedodatnim czasie podróży
  - mapy, w których istnieją miasta o ujemnej ilości produktów możliwych do sprzedania

- 
- mapy, dla których sprzedawca dysponuje ujemnym czasem
  - mapy, dla których istnieją drogi między nieistniejącymi miastami
  - rozwiązania, które zawierają drogi nie uwzględnione na mapie
  - rozwiązania, dla których czas przejścia sprzedawcy po mapie jest większy niż czas, którym sprzedawca dysponuje
5. Aplikacja pozwala użytkownikowi na wczytywanie czasu, którym dysponuje sprzedawca
  6. Aplikacja zapisuje rozwiązanie optymalne do pliku *.txt* w postaci ciągu miast.
  7. Aplikacja odrzuca pliki *.csv*, nie odpowiadające ustalonej postaci danych wejściowych.
  8. Aplikacja pozwala na sprzedanie towarów w dowolnym mieście jeden raz.

## 2.2 Wymagania niefunkcjonalne

1. Aplikacja znajduje rozwiązanie optymalne za 30 sekund.
2. Jeśli rozwiązanie optymalne nie zostało znalezione za 30 sekund, aplikacja zwraca rozwiązanie przybliżone.
3. Aplikacja może być uruchomiona na systemie Windows 7 lub nowszym.
4. Interfejs aplikacji jest w języku polskim.
5. Minimalne wymagania sprzętowe:
  - procesor: co najmniej 1GHz; zalecane 2GHz lub więcej
  - połączenie z internetem: nie jest wymagane
  - dysk twardy: co najmniej 2GB
  - pamięć (RAM): co najmniej 1GB; zalecane 4GB lub więcej
  - monitor
  - mysz komputerowa
  - klawiatura

## 3 Przypadki użycia oraz sytuacje błędne

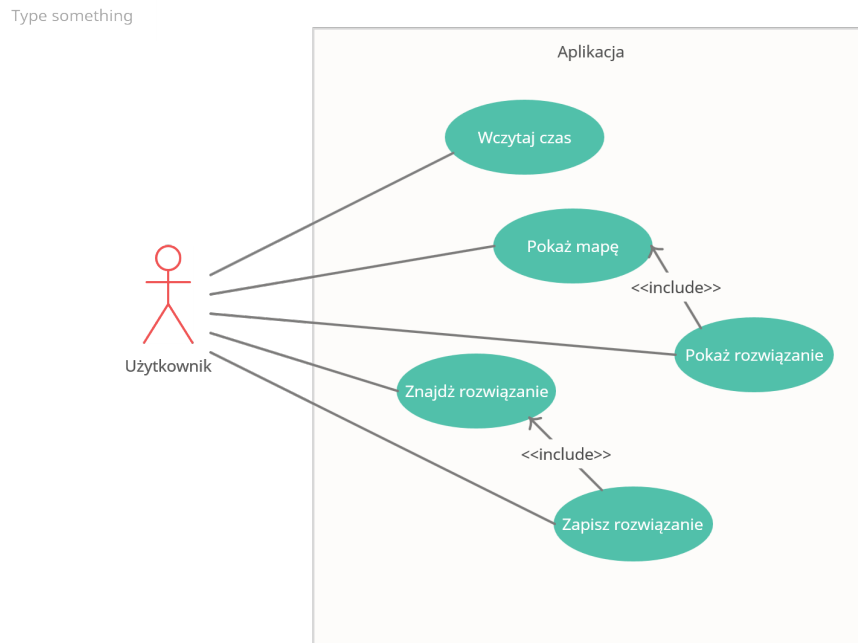
### 3.1 Przypadki użycia

Aplikacja pozwala użytkownikowi:

- pokazać mapę złożoną z miast i dróg
- pokazać rozwiązanie istniejące dla podanej mapy

- 
- znaleźć rozwiązanie dla podanej mapy
  - zapisać rozwiązanie dla podanej mapy
  - wczytać czas, którym dysponuje sprzedawca

Odpowiednie związki między przypadkami użycia można zobaczyć na diagramie przypadków użycia poniżej.

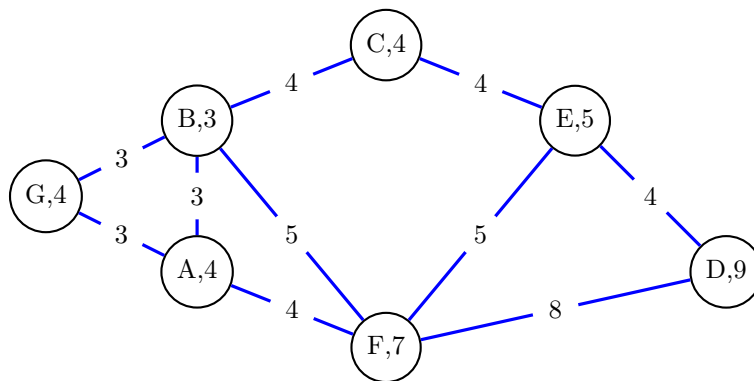


Rysunek 2: Diagram przypadków użycia.

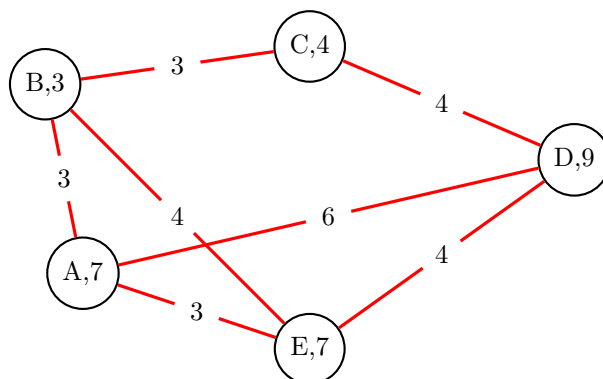
### 3.2 Sytuacje błędne

Wszystkie przypadki błędnych danych zostały opisane w podpunkcie czwartym wymagań funkcjonalnych. W przypadku danych błędnych aplikacja odrzuca ich z odpowiednim komunikatem. Omówmy sytuacji błędne aplikacji używając do tego graficznego przedstawienia na przykładowych grafach. Wierzchołki grafów oznaczone są w sposób [nazwa\_miasta, liczba\_produktów\_do\_sprzedania]. Położenie wierzchołka odpowiada położeniu miasta na mapie. Krawędzie między wierzchołkami odpowiadają drogom między miastami, które wierzchołki reprezentują. Wartości na krawędziach odpowiadają liczbie godzin potrzebnych do przejścia drogi między miastami. Rysunek 1 przedstawia poprawne dane (spełniające założenia projektu). Następne rysunki pokazują przykłady błędnych danych. W ich podpisach opisane jest dlaczego program nie przyjmie danych. W

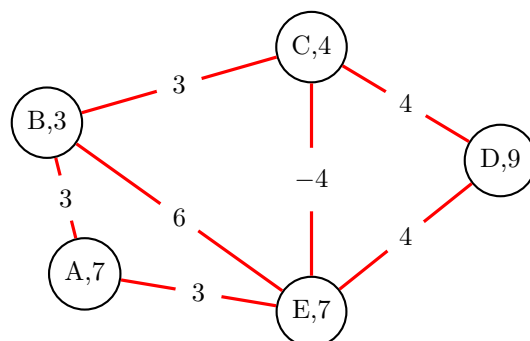
tych przypadkach aplikacja odrzuca dane wejściowe z odpowiednim komunikatem.



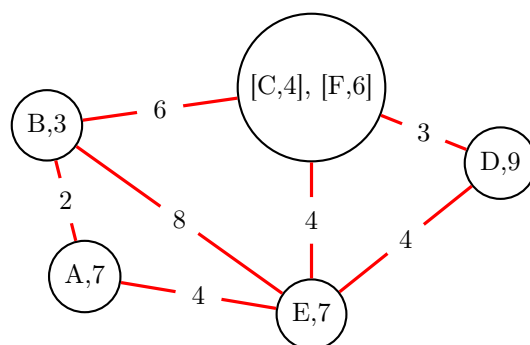
Rysunek 3: Dane poprawne



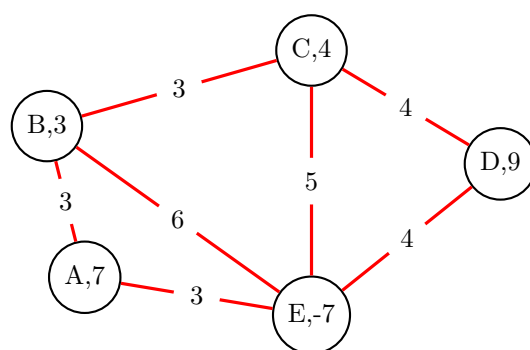
Rysunek 4: Błąd przecinania się dróg. Droga A-D przecina się z drogą B-E i w przecięciu ich nie znajduje się inne miasto.



Rysunek 5: Błąd podania niedodatniego czasu drogi. Czas drogi C-E wynosi -4.

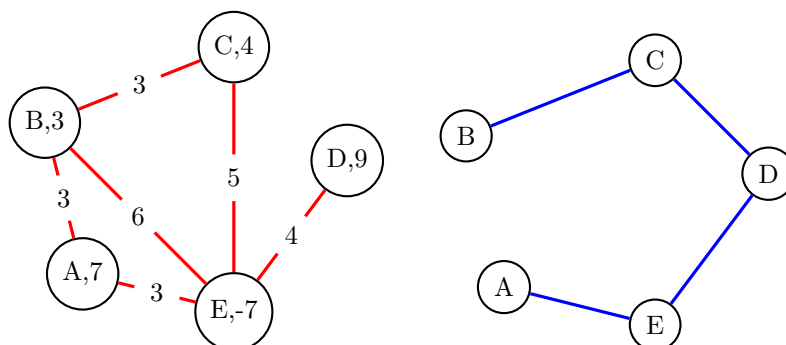


Rysunek 6: Błąd kilku miast w jednym miejscu. Miasta C i F znajdują się w jednym punkcie.



Rysunek 7: Błąd podania ujemnej liczby produktów do sprzedaży. Dla miasta E wynosi ona -7.





Rysunek 8: Lewy graf przedstawia mapę dróg, a prawy rozwiązanie. Zauważmy, że droga C-D nie istnieje, ale jest podana w rozwiązaniu. Jest to błąd.

## 4 Zmiana harmonogramu

Nie planowana jest zmiana terminów w harmonogramie projektu. Jednak z powodu rozmiarów dokumentu Michał Swidziński napisał w dokumentacji funkcjonalnej słowny opis problemu. Dodatkowo podczas wspólnego spotkania przez platformę MS Teams wszyscy członkowie zespołu uczestniczyli w ogólnej edycji dokumentu.

---

## 5 Historia zmian

Historia zmian			
Wersja	Data	Kto	Opis
0.10	25.03.2021	Kinga Kawczyńska	Utworzony dokument
0.15	25.03.2021	Volodymyr Boiko	Dodanie metryki oraz historii zmian. Tworzenie bazowej struktury dokumentu.
0.45	28.03.2021	Volodymyr Boiko	Dodanie wymagań funkcjonalnych. Dodanie wymagań нефункциональных.
0.50	28.03.2021	Kinga Kawczyńska	Dodanie sekcji sytuacji błędnych. Stworzenie 4 przykładów grafów.
0.55	29.03.2021	Volodymyr Boiko	Przypadki użycia aplikacji. Diagram przypadków użycia.
0.60	29.03.2021	Swidziński Michał	Opis problemu słowny.
0.70	29.03.2021	Volodymyr Boiko	Opis problemu formalny. Literówki.
0.8	29.03.2021	Kinga Kawczyńska Volodymyr Boiko Swidziński Michał	Ogólna edycja treści dokumentu.
0.9	29.03.2021	Kinga Kawczyńska Volodymyr Boiko	Przerobienie grafu z rozdziału 1. Dodanie sytuacji błędnych wraz z ilustracjami. Edycja wymagań funkcjonalnych.
1.0	29.03.2021	Kinga Kawczyńska	Edycja wymagań funkcjonalnych i нефункциональных. Uwzględnienie zmian harmonogramu.
1.1	29.03.2021	Volodymyr Boiko	Edycja przypadków użycia Edycja diagramu przypadków użycia.
1.15	31.03.2021	Volodymyr Boiko	Poprawiono błąd we wzorze na zysk drogi. Dodano streszczenie dokumentu.