**Inżynieria Oprogramowania**Raport

Grzegorz Puścizna, Michał Kot

1. **Inżynieria wymagań**
   1. **Macierz umiejętności**

|  |  |  |
| --- | --- | --- |
| **Kompetencje** | **Grzegorz Puścizna** | **Michał Kot** |
| Znajomość języka Java | Tak | Tak |
| Użytkowanie GITa | Tak | Nie |
| Tworzenie UML | Nie | Tak |
| Komentowanie kodu | Tak | Tak |
| Planowanie pracy | Tak | Tak |
| Testowanie programu | Nie | Nie |

* 1. **Pytania szczegółowe**

|  |  |  |
| --- | --- | --- |
| **Pytanie** | **Odpowiedź** | **Uwagi** |
| Ile paczek w trasie ? | Maksymalnie 5. | Może być mniej. |
| Jak ma wyglądać mapa ? | Mapa będzie macierzą liczb całkowitych. Dane będą wagi poziome i pionowe. | Musimy odpowiednio wczytać wagi. |
| Co jest kryterium przy wyborze trasy ? | Czas (koszt) przejścia między miastami (węzłami). | Rozpatrujemy w algorytmie wagi. |
| Co ma być plikiem wyjściowym ? | Raport jako plik txt. | Wyniki działań muszą być zapisywane do pliku. |
| Do czego służy epooch ? | Będzie służył do testów w naszym programie. | Nie ma znaczenia dla funkcjonalności programu. |

* 1. **Format danych wejściowych**

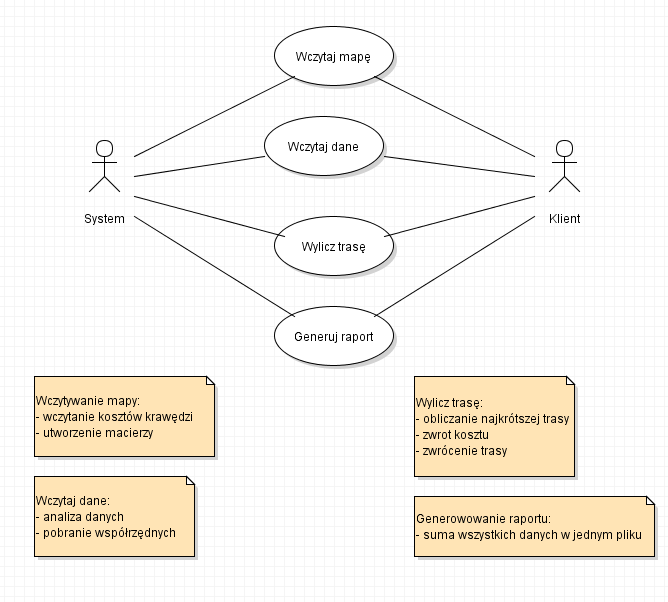
Dane: 2535135567,Jan,(1,10),(2,20),(2,9),(3,30),(31,22);

Mapa: 1234567891  
 1234567891  
 1234567891  
 1234567891  
 1234567891  
 1234567891  
 1234567891  
 1234567891  
 1234567891  
 1234567891

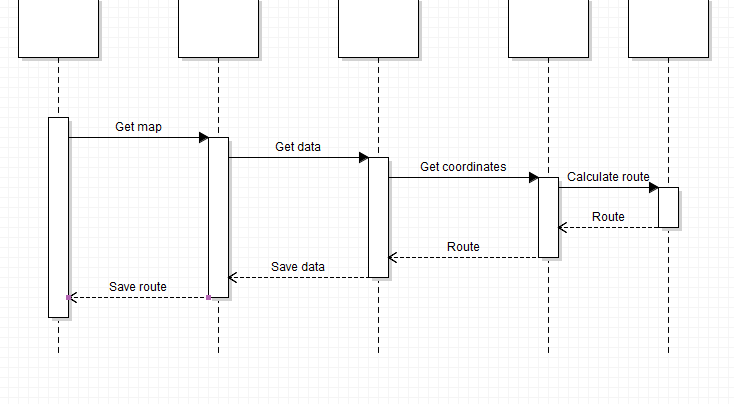
1. **Model i Architektura systemu**
   1. **Tabela UML**

|  |  |
| --- | --- |
| Aktorzy | System, klient |
| Opis | Obliczanie najbardziej optymalnej trasy dla kierowców |
| Dane | Mapa, czas przejazdów ulic; Czas otrzymania paczki, kierowca, współrzędne paczek |
| Wyzwalacz | Kolejna paczka |
| Odpowiedź | Optymalna trasa rozwozu i jej czas |
| Uwagi | Musimy przetworzyć dane mapy, aby stworzyć macierz |

* 1. **Diagram UML**



* 1. **Diagram Sekwencyjny UML**



1. **Testowanie oprogramowania**
   1. **Znalezione błędy**

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Opis błędu** | **Rozwiązanie** |
| 1 | Błędny zapis tras we Floydzie | Poprawa wczytywania grafu |
| 2 | Błędne trasy w Dijkstrze | Poprawa porównywania wag krawędzi |
| 3 | Błędy przy wczytywaniu danych | Poprawa wykrywania błędów |
| 4 | Błędy przy czyszczeniu arraylisty | Użycie pętli while |
| 5 | Nieodpowiednie wagi do mapy | Przystosowanie kodu do błędnych danych |
| 6 | Algorytmy wykonują się zbyt długo | Optymalizacja kodu |

* 1. **Przetestowane funkcjonalności**

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Funkcjonalność** | **Uwagi** |
| 1 | Nieprawidłowy czas otrzymania zlecenia | Informacja o błędzie w danych |
| 2 | Błąd w danych wejściowych | Informacja o błędzie |
| 3 | Nadmiarowe wagi w mapie | Nadmiarowe wagi są usuwane |

**Poprawić funkcjonalności wczytywanie mapy itp**