Studencka Pracownia Inżynierii Oprogramowania
Instytut Informatyki Uniwersytetu Wrocławskiego

Bartosz Bednarczyk, Łukasz Dzwoniarek

Inteligentny system planowania podróży pojazdami MPK

Koncepcja systemu

Wrocław, 21 listopada 2016
Wersja 0.7

Tabela 0. Historia zmian dokonanych w dokumencie

Data	Numer wersji	Opis	Autor
2016-11-13	0.1	Utworzenie dokumentu	Łukasz Dzwoniarek
2016-11-15	0.2	Dodanie opisu architektury	Bartosz Bednarczyk
2016-11-16	0.3	Dodanie zasad kodowania	Łukasz Dzwoniarek
2016-11-19	0.4	Dodanie przypadków użycia	Bartosz Bednarczyk
2016-11-20	0.5	Dodanie opisu bazy danych	Bartosz Bednarczyk
2016-11-21	0.6	Korekta formatowania	Łukasz Dzwoniarek
2016-11-25	0.7	Dodanie schematu bazy danych	Bartosz Bednarczyk

1. Scenariusze przypadków użycia

W tym rozdziale omówione zostaną trzy najważniejsze przypadki użycia systemu. Są one realizowane przez dwóch aktorów:

- użytkownik serwisu
- administrator serwisu

1. 1. Wyznaczanie trasy:

Aktor: użytkownik serwisu
Cel: Wyznaczenie optymalnej trasy pomiędzy podanymi przez użytkownika punktami: początkowym i końcowym, oraz wyznaczenie tras alternatywnych
Przebieg:

1. Podanie użytkownika punktu początkowego, końcowego oraz czasu odjazdu lub przyjazdu oraz sposobu poruszania się (pieszo, komunikacja miejska).
2. Wywołanie funkcji wyznaczania optymalnej trasy z podanymi przez użytkownika parametrami oraz zwrócenie jej wyniku.

1. 2. Wyznaczanie czasu podróży:

Aktor: użytkownik serwisu
Cel: Wyznaczenie czasu pokonania trasy przez użytkownika
Przebieg:

1. Wywołanie przez aplikację z podaniem optymalnej trasy oraz sposobu jej pokonania (pieszo, komunikacja miejska, samochód).
2. Analiza natężenia ruchu oraz rozkładów jazdy na podstawie których zwracana jest wartość przewidywanego czasu podróży.

1. 3. Informacja o opóźnieniach:

Aktor: administrator serwisu
Cel: Wykrywanie rozbieżności pomiędzy rozkładem jazdy, a czasami przejazdów pojazdów MPK
Przebieg:

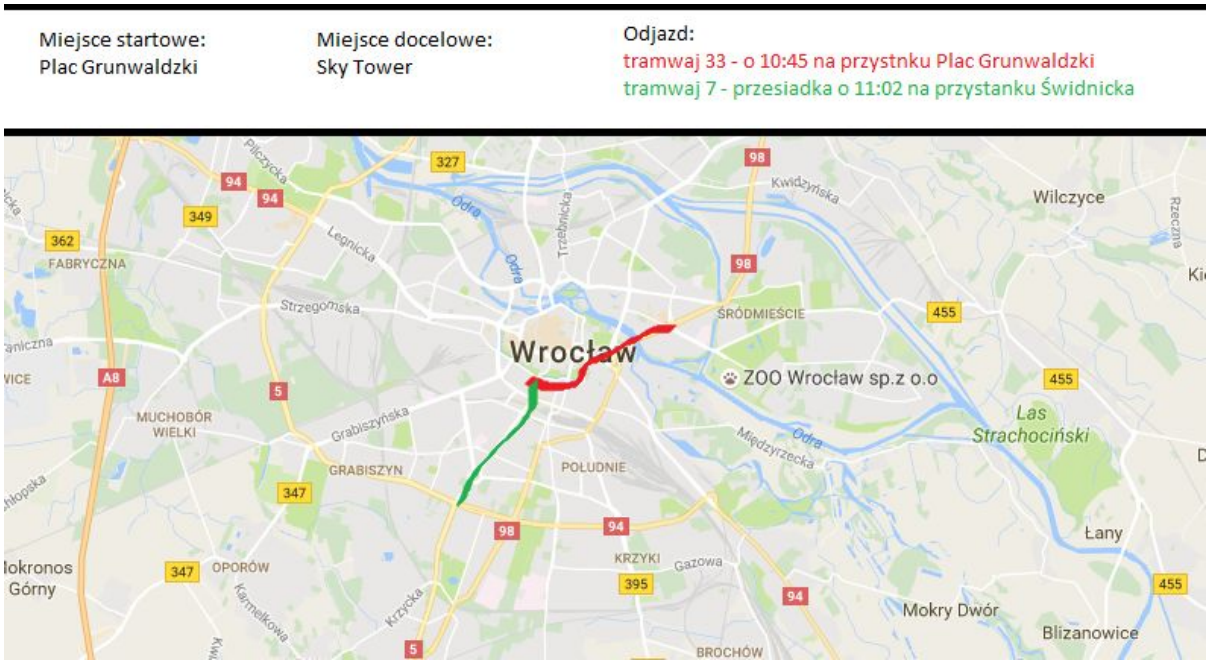
1. Inicjalizacja przez aplikację
2. Analiza położenia pojazdu i porównywanie wyników z rozkładem jazdy.
3. W przypadku wykrycia niezgodności pomiędzy położenie pojazdu, a rozkładem jazdy obliczana jest wartość opóźnienia, którą następnie zwracamy użytkownikowi.
4. W przypadku braku wykrycia niezgodności pomiędzy położeniem, a rozkładem wartość opóźnienia nie jest zwracana.

2. Przykład wyglądu interfejsu

Poniżej przedstawiono prototyp przykładowego graficznego interfejsu użytkowników systemu.

2. 1. Interfejs po stronie użytkownika:

Możliwość wyznaczenia trasy pomiędzy dwoma zadanymi miejscami wraz z podanymi godzinami odjazdów.



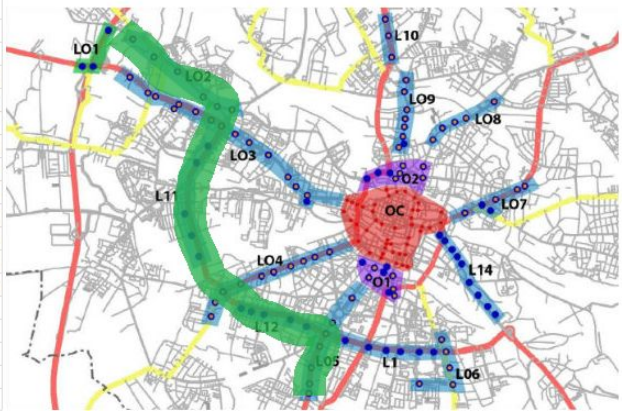
2. 2. Interfejs po stronie administratora:

Możliwość wyświetlania przewidywanych godzin odjazdów, wraz z możliwością graficznego wyświetlania trasy, dla wybranego pojazdu MPK.

127: Jaworowa Słupek: 11376

PARK POŁUDNIOWY - KOZANÓW

W dni robocze							Sobota							Niedziela						
05	24	51					05	28	58					05	58					
06	08	23	40	55			06	28	58					06	28	58				
07	05	15	25	35	45	55	07	28	58					07	28	58				
08	05	17	27	42	57		08	28	48					08	28	48				
09	07Z	12	27	42	57		09	01	20	31	46			09	01	16	31	46		
10	12	27	42	57			10	01	16	31	46			10	01	20	31	46		
11	12	27	42	57			11	01	16	31	46			11	01	16	31	46		
12	12	27	42	57			12	01	16	31	46			12	01	16	31	46		
13	12	24	39	55			13	01	16	31	46			13	01	16	31	46		
14	08	20	35	50			14	01	16	31	46			14	01	16	31	46		
15	05	20	33	48			15	01	16	31	46			15	01	16	31	46		
16	05	21	38	50			16	01	16	31	46			16	01	16	31	46		
17	17	29	43	59			17	01	16	31	46			17	01	16	31	46		
18	14	29	44	59			18	01	16	31	46			18	01	16	31	46		
19	14	29Z	44	51Z			19	01	16	46				19	01	16	46			
20	06	36					20	06	36					20	06	36				
21	06	16Z	36				21	06	16Z	36				21	06	16Z	36			
22	06	36					22	06	36					22	06	36				
23	06Z						23	06Z						23	06Z					



3. Architektura

3. 1. Specyfikacja sprzętu

Oprogramowanie serwisu będzie udostępniać wszystkie funkcje użytkownikom za pomocą strony internetowej. Z tego powodu architektura sprzętowa opiera się o model klient-serwer.

3. 1. 1. Sprzęt po stronie użytkownika

Główną zaletą naszej aplikacji jest jej dostępność w każdym miejscu i o każdej porze. Jedynym warunkiem jest obsługa przeglądarki stron internetowych (komputery, tablety, telefony komórkowe), oraz dostęp do Internetu. Dla starszych urządzeń, wyposażonych w małą ilość pamięci RAM, przewiduje się utworzenie specjalnej wersji strony pozbawionej graficznej mapy przejazdu.

3. 1. 2. Sprzęt po stronie aplikacji

W początkowym okresie działania aplikacji, jej wymagania sprzętowe nie są zbyt wygórowane. Wystarczą dobre serwery dedykowane dostępne na rynku. Wraz z rozwojem aplikacji istnieje możliwość inwestycji we własne serwery oraz dzierżawę łącza internetowego.

3. 2. Oprogramowanie systemowe

Prawidłowe działanie aplikacji ISPP wymaga od użytkownika korzystania ze sprawnego sprzętu komputerowego i oprogramowania zapewniającego dostęp do sieci Internet.

3. 2. 1. Po stronie użytkownika

W celu zapewnienia dostępności szerokiej grupie odbiorców nasza aplikacja będzie obsługiwać użytkowników korzystających z różnych systemach operacyjnych (Linux, Windows, MacOS, Android, Windows Phone) oraz na przeglądarkach stron internetowych (Mozilla Firefox, Google Chrome, Internet Explorer, Safari, Opera).

3. 2. 2. Po stronie serwera aplikacji

Aplikacja będzie korzystać z systemu operacyjnego Linux z serwera WWW Apache oraz z serwera baz danych Oracle MySQL. W projekcie zastosowany zostanie język programowania Python (IDE PyCharm) oraz framework Django.

3. 3. Struktura logiczna oprogramowania

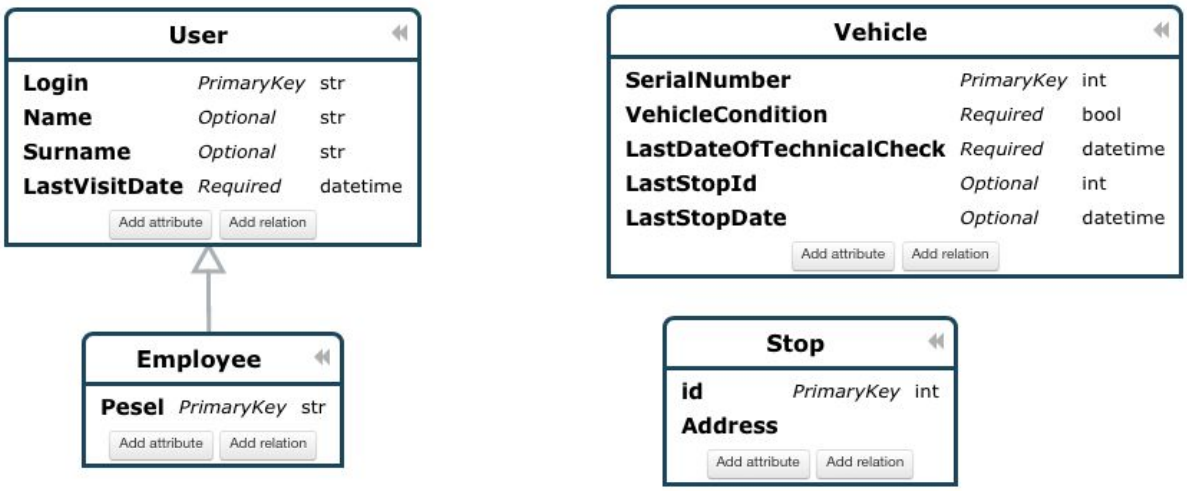
Oprogramowanie powinno być napisane zgodnie z paradygmatem programowania obiektowego wykorzystując wzorce projektowe, a w szczególności wzorca dependency injection. Kod powinien być podzielony na klasy niewielkich rozmiarów, tak by jak najbardziej ułatwić testowanie programu. Do każdej klasy powinny zostać zaprojektowane i zaimplementowane testy jednostkowe, integracyjne oraz funkcjonalne.

4. Schemat bazy danych systemu

4. 1. Schemat tabeli

Baza danych została podzielona na cztery tabele:

- User (Użytkownicy)
- Employee (Pracownicy)
- Vehicle (Pojazdy)
- Stop (Przystanki)



4. 2. Baza użytkowników systemu

Podział na użytkowników oraz pracowników jest wymagany przez aplikację ze względu na prawa dostępu do niektórych funkcji. Przykładowo nie chcemy, aby zwykły użytkownik miał dostęp do wszystkich danych technicznych pojazdów MPK. Tego typu funkcje dostępne będą tylko dla pracowników, którzy w naszym serwisie będą musieli autoryzować się numerem pesel. Użytkownicy nie posiadający takiego numeru (np. obcokrajowcy) dostaną osobne czterocyfrowe numery identyfikacyjne, zaczynając od 0000, na potrzeby systemu.

4. 3. Baza pojazdów i informacji o przejazdach

W tabeli pojazdy przechowywać informacje na temat numeru seryjnego pojazdu i jego stanie technicznym. Dodatkowo będziemy przechowywać również datę jego ostatniego przeglądu, aby nasz serwis mógł informować pracowników na temat zbliżającej się daty okresowego przeglądu pojazdu. Następnie na potrzeby mierzenia czasu przyjazdu oraz zliczania ewentualnych opóźnień na trasie. Baza danych przechowujemy również numer identyfikacyjny przystanku, na którym znajdował się pojazd oraz datę i godzinę przyjazdu na ten przystanek.

Każdy przystanek opisany jest jako pewien unikatowy numer oraz adres, będący położeniem geograficznym przystanku na mapie. Zakłada się zaimportowanie istniejącego już rozkładu jazdy ze stron urzędu miasta Wrocławia. Będzie to podstawą do działania funkcji wyznaczania czasów przejazdów zanim system zgromadzi wystarczającą ilość danych statystycznych.

5. Główne zasady kodowania

5. 1. Strona Internetowa

Interfejs powinien zatem spełniać standardy organizacji W3C. Kod strony powinien być zgodny ze standardem XHTML 1.0, a arkusz stylów ze standardem CSS 3.0. Interfejs użytkownika powinien zaś być prosty i intuicyjny. Strona powinna implementować zasady Responsive Web Design, płynnie dostosowując wygląd do ekranów o rozmiarach od 4 do 20 cali. Powinna poprawnie wyświetlać się w poniższych przeglądarkach internetowych:

- Chrome pow Windows 10 od wersji 52
- Firefox pod Windows 10 od wersji 47
- Safari pod iOS od wersji 9.0
- Android Browser od wersji 5.0

5. 2. Backend serwisu

Strona internetowa ma opierać się o WWW Apache w wersji 6 lub nowszej. Serwer hostujący stronę internetową musi działać pod kontrolą Oracle MySQL. Oprogramowanie działając na czterech procesorach Intel Xeon E5 o taktowaniu zegara 3 GHz powinien być w stanie obsłużyć 1000 zapytań na godzinę ze średnim opóźnieniem 1 sekundy.

5. 3. Baza danych

Baza danych systemu powinna opierać się o Oracle MySQL działającym pod kontrolą systemu Linux. Aplikacja nie musi przechowywać wszystkich informacji w tym zewnętrznym serwerze danych. Wymagane jest przechowywanie w ten sposób tylko aktualnie obliczony rozkład jazdy dla wszystkich pojazdów MPK.

5. 4. Formatowanie kodu

- wcięcia w kodzie za pomocą tabulatora
- pętle nawet jedno liniowe muszą być otoczone nawiasami klamrowymi w poniższy sposób:
  if(a<b)
  {
  a++;
  }
- nazwy funkcji i zmiennych powinny być utworzone w języku angielskim i być adekwatne do pełnionych ról
- komentarze w kodzie powinny być zwięzłe i zrozumiałe

6. Ocena zgodności wykonanych prac z wizją systemu i specyfikacją wymagań

Po przeanalizowaniu wykonanych prac projektowych systemu uznano, że spełniają one wcześniej postawione przed nimi zadania. Zaplanowane struktury danych pozwalają na przechowywanie wszystkich niezbędnych dla użytkowników danych. Wybrana architektura sprzętowa i narzędzia programistyczne pozwalają na zaimplementowanie rozwiązań spełniających wszystkie przypadki użycia.

8. Słownik

- **Aplikacja** – program komputerowy.
- **FAQ** – często zadawane (przez użytkowników) pytania oraz odpowiedzi na nie.
- **ISPP** - Inteligentny system planowania podróży
- **Inteligentny** - potrafiący dostosować swoje działanie do aktualnej sytuacji, nie polegający na z góry zdefiniowanych schematach
- **Online** – status osoby, serwera, przedmiotu - podłączony do Internetu.
- **Serwer** – komputer lub program przeznaczony do obsługi użytkowników przez udostępnianie ich komputerom swoich zasobów i udostępnianie pewnych usług.
- **SZBD** – system zarządzania baza danych (np. MySQL, Oracle, PostgreSQL).