

Wydział Matematyki i Nauk Informacyjnych Politechniki Warszawskiej



## Dokumentacja aplikacji

*Imię i nazwisko:* Sebastian Sudra i Łukasz Sznajder

*Kierunek studiów:* Informatyka

*Tytuł projektu:* Easy Train Tickets

Warszawa, dnia 21.12.2016

# Spis treści

<b>1</b>	<b>Specyfikacja</b>	<b>2</b>
1.1	Opis biznesowy . . . . .	2
1.2	Wymagania funkcjonalne . . . . .	2
1.3	Wymagania нефункционалне . . . . .	4
1.4	Harmonogram projektu . . . . .	4
1.5	Architektura rozwiązania . . . . .	5
<b>2</b>	<b>Dokumentacja końcowa</b>	<b>5</b>
2.1	Wymagania systemowe . . . . .	5
2.2	Biblioteki wraz z określeniem licencji . . . . .	6
2.3	Instrukcja instalacji . . . . .	6
2.4	Instrukcja uruchomienia . . . . .	6
2.5	Instrukcja użycia . . . . .	6
2.6	Raport odstępstw od specyfikacji wymagań . . . . .	13

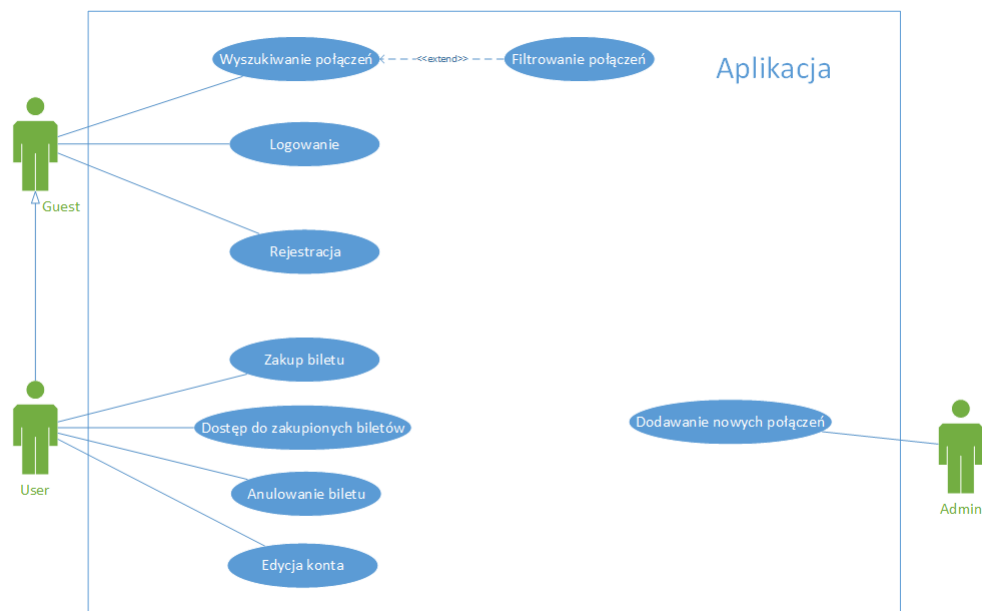
# 1 Specyfikacja

## 1.1 Opis biznesowy

Obecnie wszyscy żyjemy w globalnej wiosce. Rozwój środków transportu umożliwił szybsze przemieszczanie się po ziemskim globie. Zaś, rozwój telekomunikacji i nowoczesnych technologii sprawia, że przemieszczanie się może stać się łatwiejsze i wygodniejsze. Wychodząc naprzeciw oczekiwaniom naszym pasażerom oferujemy najnowszą aplikację umożliwiającą szybką rezerwację biletów na dany pociąg bez wychodzenia z domu. System zapewnia szybki i klarowny zakup biletu z określonym miejscem oraz ze stosowną ulgą. Aplikacja jest skierowana dla każdego użytkownika, któremu nie jest obojętny komfort podróży.

## 1.2 Wymagania funkcjonalne

Na poniższym rysunku przedstawiono w postaci diagramu UML zbiór przypadków użycia aplikacji.



Rysunek 1: Diagram przypadków użycia

W poniższej tabeli zebrano opisy poszczególnych przypadków użycia aplikacji.

Aktor	Nazwa	Opis	Odpowiedź systemu
Gość	Wyszukiwanie połączeń	Przeszukanie katalogu danych w celu znalezienia optymalnych połączeń	Wyświetla znalezione połączenia
	Filtrowanie połączeń	Filtruje połączenia względem: przesiadek, rodzaju pociągów, stacji pośrednich	Wyświetla spełniające wymagania połączenia
	Logowanie	Próba dopasowania użytkownika wraz z podanym hasłem do obecnej puli użytkowników	Jeżeli logowanie zakończyło się sukcesem wyświetla się konto użytkownika, w przeciwnym wypadku wyświetlany jest komunikat o błędnych danych logowania
	Rejestracja	Sprawdza czy istnieje dany użytkownik, jeśli nie, dodaje go	Jeżeli rejestracja przebiegła pomyślnie wyświetla się konto użytkownika, w przeciwnym wypadku wyświetlany jest komunikat o istniejących danych logowania
Użytkownik	Zakup biletu	Możliwość losowego lub ręcznego wyboru miejsc	Wyświetla informację o udanym lub nieudanym kupnie biletu
		Możliwość wybrania stosownej ulgi	
	Anulowanie biletu	Usuwa bilet i zwalnia miejsce w pociągu określone przez bilet	Wyświetla się komunikat o usuniętym bilecie
	Dostęp do biletów	Wyszukuje wszystkie zakupione bilety przez danego użytkownika	Wyświetla bilety użytkownika
Admin	Edycja konta	Możliwość zmiany hasła	Pokazuje potwierdzenie
	Wstawianie nowych połączeń	Wstawia do bazy danych nowe połączenie	Jeżeli operacja przebiegła pomyślnie wyświetla potwierdzenie

Tabela 1: Opisy przypadków użycia aplikacji

#### User stories:

##### 1. Aplikacja publiczna

- Dowolny użytkownik (również ten niezalogowany) może wyszukać interesujące połączenie podając miejsce początkowe i końcowe podróży, a także datę. Użytkownik może zdecydować czy chciałby się przesiadać, czy też nie. Może również podać stacje pośrednie oraz wybrać akceptowane rodzaje pociągów. Użytkownikowi podaje się różne warianty spełniające jego kryteria.

- Po wyszukaniu połączeń użytkownik może sprawdzić szczegóły dotyczące wybranego połączenia i pociągu.
- Zalogowany użytkownik, gdy znajdzie interesujące go połączenie, może dokonać zakupu biletu na wybraną ilość miejsc i wybrane ulgi.
- Zalogowany użytkownik przeglądając zakupione bilety może je anulować.

## 2. Interfejs administracyjny

- Administrator może dodać nowe połączenie podając trasę pociągu wraz z przystankami pośrednimi oraz daty kursowania.

### 1.3 Wymagania niefunkcjonalne

Obszar wymagań	Nr wymagania	Opis
Użyteczność (Usability)	1	Wszystkie funkcjonalności aplikacji dostępne dla użytkownika muszą być intuicyjne i przejrzyste
Niezawodność (Reliability)	2	Aplikacja ma być aplikacją typu High Availability – dostępną w trybie 24x7 nie licząc przerwy serwisowej od 23:00 do 1:00
	3	Aplikacja zapewnia ochronę danych nawet w sytuacjach nieoczekiwanych
Wydajność (Performance)	4	Aplikacja wykorzystuje optymalne algorytmy w celu wyszukiwania połączeń w czasie nie dłuższym niż 3 sekundy
	5	Aplikacja powinna zapewniać wyszukiwanie optymalnych połączeń
Utrzymanie (Supportability)	6	Aplikacja przetestowana jest na różnych poziomach

Tabela 2: Lista wymagań niefunkcjonalnych

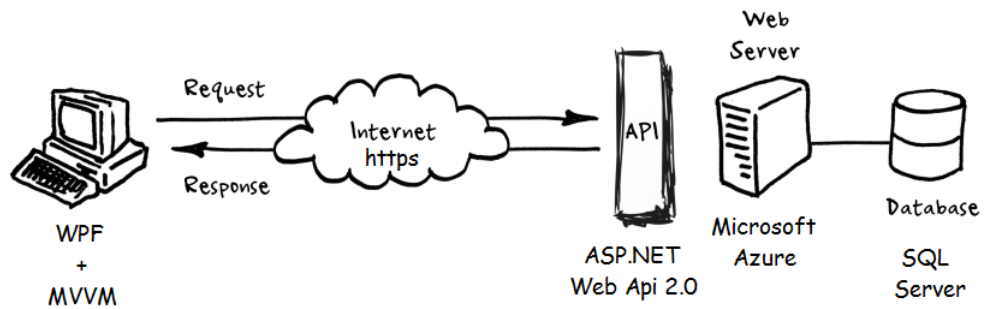
### 1.4 Harmonogram projektu

Planowana jest realizacja projektu zgodnie z następującym harmonogramem:

Temat	Dni	rozpoczęcie	zakończenie
Edycja bazy danych	4	19-11-2016	22-11-2016
Utworzenie Service Oriented Architecture	14	23-11-2016	06-12-2016
Utworzenie edycji konta	2	07-12-2016	09-12-2016
Filtrowanie połączeń	3	10-12-2016	12-12-2016
Zakup biletu przez użytkownika	3	13-12-2016	15-12-2016

Tabela 3: Harmonogram projektu

## 1.5 Architektura rozwiązania



Rysunek 2: Architektura aplikacji

## 2 Dokumentacja końcowa

### 2.1 Wymagania systemowe

Do uruchomienia aplikacji wymagany jest:

- System MS Windows 7 lub nowszy
- .NET Framework 4.5.2
- połączenie z internetem
- 512MB RAM

## 2.2 Biblioteki wraz z określeniem licencji

nr	Komponent i wersja	Opis	Licencja
1	Entity Framework 6.1.2	Entity Framework jest doskonałym narzędziem typu ORM	Apache 2.0
2	SQL Server 2014	Zapewnia tworzenie bazy danych	Server+CAL license model or a CPU Core license model
3	Caliburn.micro 3.0.1	Framework wspomaga tworzenie aplikacji opartej o wzorzec MVVM	MIT
4	OptimizedPriorityQueue 3.0.0	Wysoce zoptymalizowana kolejka priorytetowa	MIT
5	Extended Wpf Toolkit	Dodatkowe kontrolki	Ms-PL
6	Moq 4.5.23	Tworzenie atrap w .NET	The BSD 2-Clause License
7	Microsoft.AspNet.Mvc 5.2.3	platforma aplikacyjna do budowy aplikacji internetowych opartych na wzorcu Model-View-Controller	Apache 2.0
8	Microsoft.AspNet.WebApi 5.2.3	platforma aplikacyjna do budowy API Services	Apache 2.0
9	AutoMapper 5.2.0	Mapowanie obiektów	MIT
10	Ninject.Web.WebApi 3.2.4	udana implementacja wzorca Dependency Injection	Apache 2.0

Tabela 4: Lista komponentów

## 2.3 Instrukcja instalacji

Aplikacja nie wymaga instalacji

## 2.4 Instrukcja uruchomienia

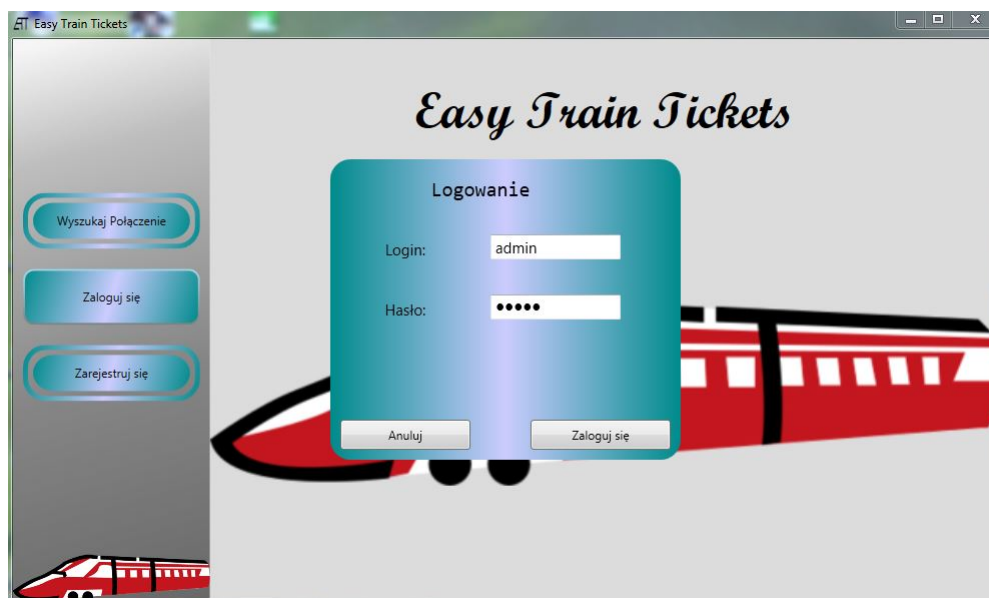
Należy dwukrotnie wcisnąć lewy przycisk myszy na EasyTrainTickets.DesktopClient.exe

## 2.5 Instrukcja użycia

Każda osoba (gość) może dokonać rejestracji w aplikacji podając nazwę użytkownika (login) oraz hasło. Zalogowany użytkownik istniejący już w bazie może zmienić hasło wielokrotnie zachowując przy tym swój login. Jeżeli chcemy się zalogować klikamy na przycisk "Zaloguj się" znajdujący się w menu przy lewej stronie okna aplikacji. Logując się na konto admina wpisujemy:

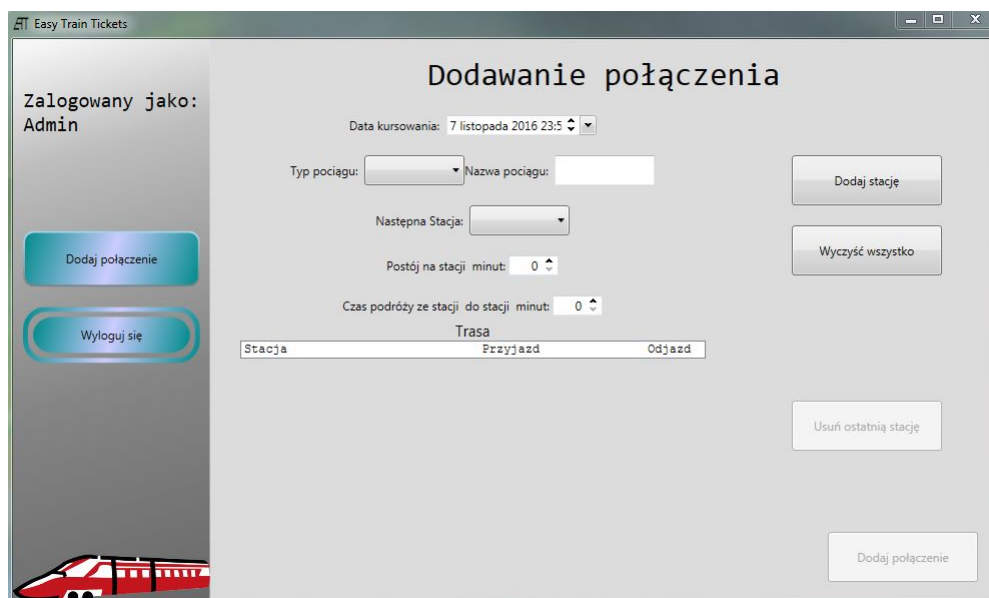
Login: admin

Hasło: admin



Rysunek 3: Logowanie do aplikacji

Admin może dodawać nowe połączenia korzystając z odpowiedniego okna, które zostaje uruchomione przez wciśnięcie przycisku "Dodaj połączenie".



Rysunek 4: Okno dodawania połączeń



W celu dodania nowego połączenia trzeba:

- wybrać datę odjazdu
- wybrać typ pociągu
- wpisać nazwę pociągu
- wybrać trasę pociągu poprzez dodawanie kolejnych stacji wybranych w comboboxie i kliknięcia przycisku "Dodaj stację"
- wybrać czas postoju i czas podróży z jednej stacji do drugiej (czas jest liczony w minutach)

Na dole okna widoczna jest aktualna trasa pociągu, tak jak na rysunku 5. Gdy skończymy definiować trasę dodajemy połączenie klikając przycisk "Dodaj połączenie".

Easy Train Tickets

Zalogowany jako: Admin

**Dodawanie połączenia**

Data kursowania: 8 listopada 2016 09:00

Typ pociągu: Ekspres    Nazwa pociągu: Pendolino

Następna Stacja:

Postój na stacji Kraków minut: 10

Czas podróży ze stacji Kraków do stacji minut: 0

Trasa		
Stacja	Przyjazd	Odjazd
Gdańsk		08:00
Warszawa	11:40	11:50
Kraków	14:15	

Dodaj stację

Wyczyść wszystko

Usuń ostatnią stację

Dodaj połączenie

Rysunek 5: Wprowadzanie danych do nowego połączenia

Osoba zalogowana, jak i osoba niezalogowana może wyszukać połączenie poprzez kliknięcie przycisku "Wyszukaj połączenie". W celu wyszukania interesującego nas połączenia zaznaczamy datę odjazdu, a także z jakiego miejsca i dokąd chcemy się wybrać. Do dyspozycji mamy także trzy filtry:

1. jednym z nich jest możliwość dodania stacji pośredniej, czyli stacji przez którą chcemy odbyć naszą podróż
2. drugi z nich sprawdza czy interesują nas trasy z przesiadkami czy bezpośrednie
3. trzeci dopuszcza do naszej trasy ekspresy

Po wciśnięciu przycisku "Szukaj" wyświetlają nam się wyszukane połączenia. W celu uzyskania więcej informacji o danym połączeniu klikamy na nie i pokazują się dodatkowe informacje, tak jak na rysunku nr 6.

**Easy Train Tickets**

**Wyszukiwarka Połączeń**

Zalogowany jako: user

Skąd: Białystok      Stacja Pośrednia: Warszawa      Dokąd: Poznań

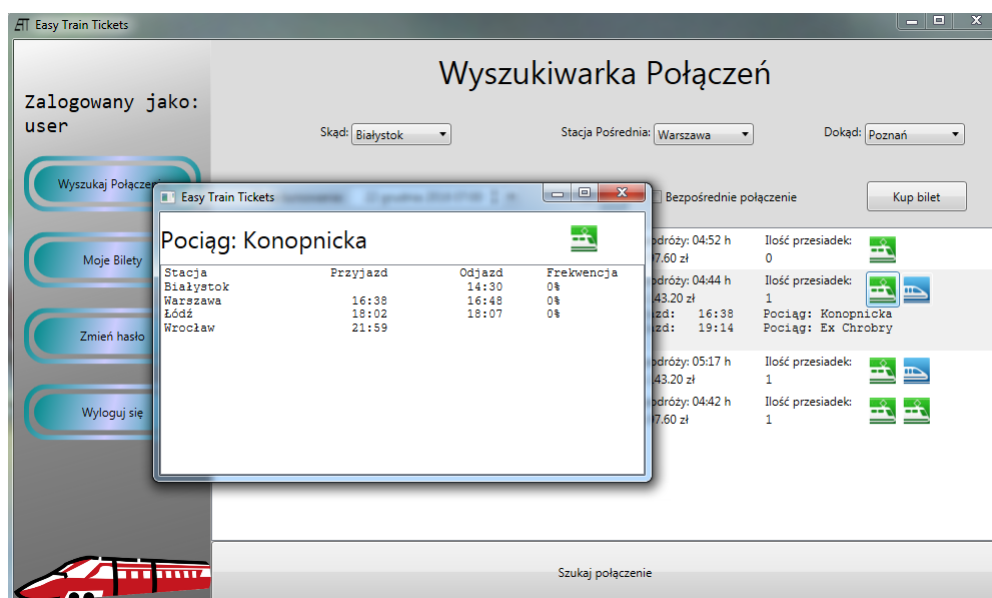
Data kursowania: 22 grudnia 2016 07:00      ☒      ☐ Bezpośrednie połączenie      [Kup bilet](#)

Z: Białystok Do: Poznań	22.12.2016	Odjazd: 09:03 Przyjazd: 13:55	Czas podróży: 04:52 h Cena: 97.60 zł	Ilość przesiadek: 0	
Z: Białystok Do: Poznań	22.12.2016	Odjazd: 14:30 Przyjazd: 19:14	Czas podróży: 04:44 h Cena: 143.20 zł	Ilość przesiadek: 1	
<div>Białystok =&gt; Warszawa</div> <div>Warszawa =&gt; Poznań</div>		<div>Odjazd: 14:30</div> <div>Przyjazd: 16:38</div> <div>Odjazd: 17:00</div> <div>Przyjazd: 19:14</div>			<div>Pociąg: Konopnicka</div> <div>Pociąg: Ex Chrobry</div>
Z: Białystok Do: Poznań	22.12.2016	Odjazd: 16:00 Przyjazd: 21:17	Czas podróży: 05:17 h Cena: 143.20 zł	Ilość przesiadek: 1	
Z: Białystok Do: Poznań	22.12.2016	Odjazd: 17:46 Przyjazd: 22:28	Czas podróży: 04:42 h Cena: 97.60 zł	Ilość przesiadek: 1	

[Szukaj połączenie](#)

Rysunek 6: Wyszukiwarka połączeń

Istnieje również możliwość sprawdzenia całej trasy pociągu, którym będziemy jechać, a także zobaczenia frekwencji na danej trasie. W celu uzyskania takiej informacji klikamy na ikonkę pociągu umieszczoną z prawej strony, co przedstawia rysunek nr 7.



Rysunek 7: Szczegółowy opis pociągu

Jeżeli chcemy dokonać zakupu danego biletu wystarczy zaznaczyć dane połączenie i kliknąć przycisk "Kup bilet". Do tej operacji wymagane jest logowanie, jeżeli użytkownik tego nie zrobił wyświetli się okno logowania, w przeciwnym wypadku aplikacja zabierze użytkownika do okienka zakupu biletu wyglądającego jak na rysunku nr 8.

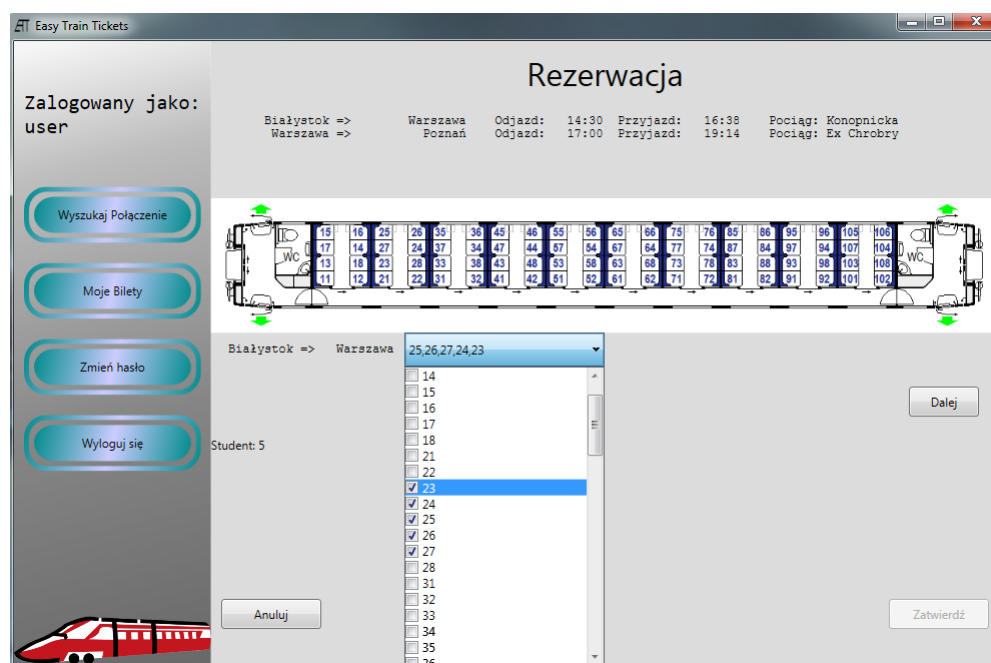


Rysunek 8: Wybieranie ulg

W tym oknie decydujemy o rodzaju ulgi oraz liczbie biletów, jakie chcemy kupić. Do wyboru mamy bilety:

- normalny
- student
- uczeń
- senior

Możemy zakupić maksymalnie 5 biletów z każdej ulgi. Aplikacja oferuje nam także losowy wybór miejsc w pociągu, w przypadku gdy sami nie chcemy ich wskazywać. Po wybraniu ilości biletów klikamy przycisk "Przejdź do rezerwacji". Aplikacja przenosi nas do okna rezerwacji miejsc, w którym mamy możliwość wybrania sobie danych miejsc w pociągu, którym jedziemy poprzez wybranie konkretnych numerów w comoboboxie i zatwierdzenie ich przyciskiem "Dalej". Przycisk ten zostanie aktywny tylko wtedy, gdy zaznaczymy odpowiednią ilość miejsc. Przedstawione okno zostało ukazane na rysunku nr 9.



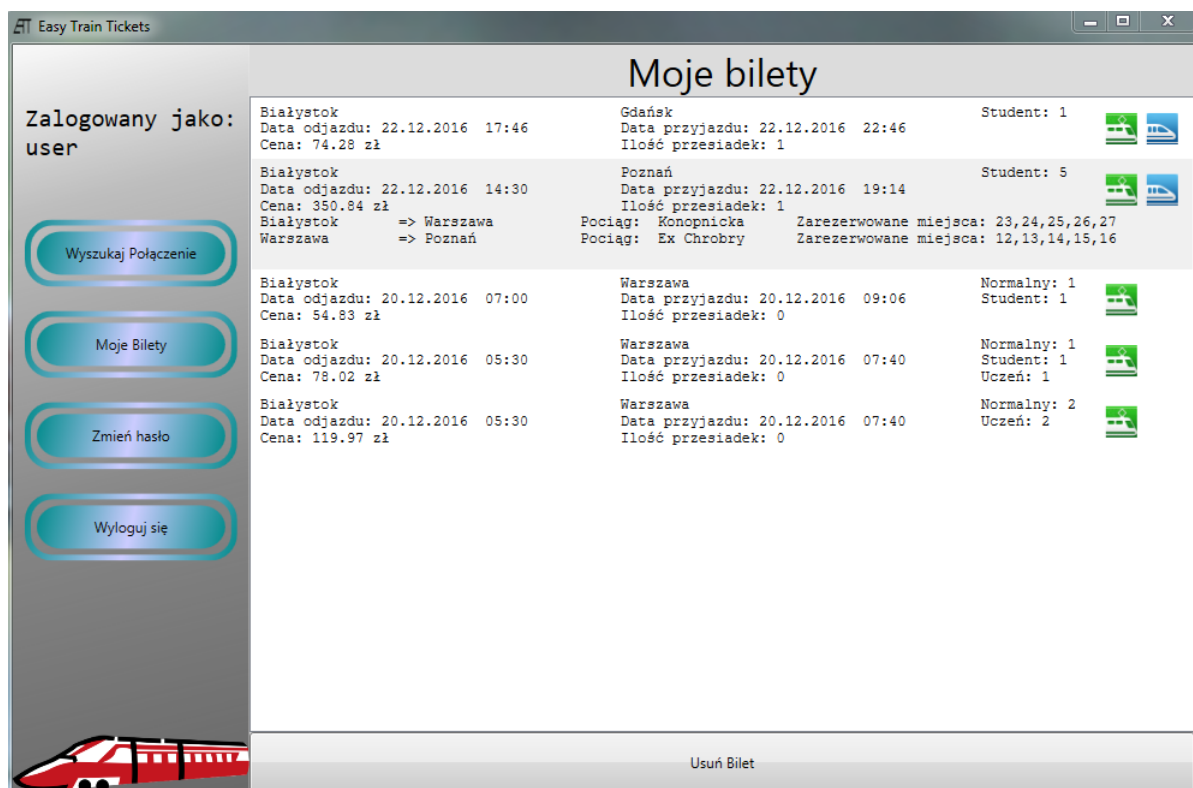
Rysunek 9: Wybranie danych miejsc

Gdy wybierzemy już wszystkie miejsca, aplikacja umożliwi nam zatwierdzenie biletu, co jest równoznaczne z jego zakupieniem (zob. rys. nr 10).



Rysunek 10: Zatwierdzenie biletu

Po zatwierdzeniu, czyli kliknięciu na przycisk "Zatwierdź" wrócimy do okna wyszukiwania. Jeżeli operacja zakończyła się sukcesem to zostanie wyświetlona informacja na zielono. Gdy jesteśmy zalogowani i chcemy zobaczyć zakupione wcześniej bilety wystarczy wybrać opcję "Moje bilety" z lewego menu. Zobaczymy wtedy nasze bilety posortowane według daty odjazdu. W celu uzyskania dodatkowych informacji o bilecie wystarczy na niego kliknąć tak jak na rysunku nr 11. Gdy data odjazdu pociągu jeszcze nie minęła wówczas możemy bilet usunąć klikając na przycisk "Usuń bilet".



Rysunek 11: Wypisywanie biletów

## 2.6 Raport odstępstw od specyfikacji wymagań

### Zmiana:

Zmiana filtru wyszukiwania. Zamiana stacji pośrednich na jedną stację pośrednią.

### Uzasadnienie:

Ten pierwotny filtr mijał się z celem aplikacji.