Bezpieczeństwo Systemów e-biznesu

System sprzedaży biletów na imprezy masowe

Adrian Warecki  
Przemysław Stroiwąs

Spis treści

[Założenia projektowe 3](#_Toc450491385)

[Wymagania funkcjonalne 3](#_Toc450491386)

[Wymagania niefunkcjonalne 3](#_Toc450491387)

[Harmonogram prac 4](#_Toc450491388)

[Baza danych 4](#_Toc450491389)

[Diagramy UML 6](#_Toc450491390)

[Diagram przypadków użycia 6](#_Toc450491391)

[Architektura serwera 6](#_Toc450491392)

[Konfiguracja sklepu 7](#_Toc450491393)

[Instalacja sklepu 7](#_Toc450491394)

[Język polski 7](#_Toc450491395)

[Waluta 8](#_Toc450491396)

[Podatki 9](#_Toc450491397)

[Ceny brutto 10](#_Toc450491398)

[Certyfikat SSL 10](#_Toc450491399)

[Bilety 11](#_Toc450491400)

[Użytkowanie sklepu 11](#_Toc450491401)

[Wyszukiwanie wydarzeń 11](#_Toc450491402)

[Wybór opcji wydarzenia 12](#_Toc450491403)

[Wyświetlanie koszyka 12](#_Toc450491404)

[Zakup biletów 13](#_Toc450491405)

[Pobieranie biletów 15](#_Toc450491406)

# Założenia projektowe

System pozwala na prowadzenie internetowej sprzedaży biletów na imprezy masowe, np. koncerty. Internauci mogą przeglądać planowane wydarzenia i dokonać zakupu biletu on-line przy pomocy płatności elektronicznych. Klienci otrzymują zakupiony bilet w postaci elektronicznej (pdf). Weryfikacja biletu jest przeprowadzana za pomocą elektronicznych czytników kodów zainstalowanych w bramkach wejściowych lub w wersji przenośnej obsługiwanej przez uprawniony personel.

System został oparty na darmowej aplikacji *OpenCart*, udostępnianej na licencji otwartego oprogramowania. Jej głównymi zaletami jest szybkość działania i intuicyjny panel administracyjny posiadający znaczne możliwości konfiguracji i przystosowania do sprzedaży każdego rodzaju produktów. Panel ten jest bardzo prosty w obsłudze nawet dla początkującego użytkownika. Aplikacja posiada budowę opartą na modelu MVC i została zaimplementowana w języku php.

Aplikacja *OpenCart* została wybrana, ponieważ jest to rozwiązanie dobrze przetestowane, z obszerną dokumentacją, dużą liczbą dodatków oraz z bardzo dobrym wsparciem społeczności.

# Wymagania funkcjonalne

Tabela . Wymagania funkcjonalne

|  |  |  |
| --- | --- | --- |
| Nr | Wymaganie | Aktor |
| 1 | Użytkownik może przeszukać zbiór wszystkich wydarzeń lub wybrać tylko ich podzbiór. | Użytkownik |
| 2 | System udostępnia odpowiednie narzędzia do pobrania biletu. | Użytkownik |
| 3 | System pozwala dokonać płatności za zamówienie. | Użytkownik |
| 4 | Użytkownik może założyć konto w systemie. | Użytkownik, Administrator |
| 5 | Każde zamówienie oznaczona jest unikatowym identyfikatorem, przypisanym do konta użytkownika. |  |
| 6 | System udostępnia możliwość pobrania faktury przez użytkownika. | Użytkownik |
| 7 | System pozwala na dodanie jednego lub wielu biletów do zamówienia. | Użytkownik |
| 8 | System udostępnia odpowiednie narzędzia do składania zamówienia. | Użytkownik |
| 9 | System udostępnia panel administracyjny do zarządzania sklepem. | Administrator |
| 10 | System pozwala na dodawanie nowych wydarzeń | Administrator, Pracownik |
| 11 | System pozwala na wyświetlenie statystyk sprzedaży | Administrator, Pracownik |
| 12 | System pozwala na modyfikacje uprawnień grup w systemie | Administrator |

# Wymagania niefunkcjonalne

Tabela . Wymagania niefunkcjonalne

|  |  |
| --- | --- |
| Nr | Wymaganie |
| 1 | System oparty o oprogramowanie OpenCart. |
| 2 | System wykorzystuje protokół SSL. |
| 3 | System działa w najnowszych wersjach przeglądarek Mozilla Firefox, Google Chrome, Opera, Microsoft Internet Explorer. |
| 4 | Sklep jest w języku polskim |
| 5 | Sklep obsługuje walutę polską - złotówki |

# Harmonogram prac

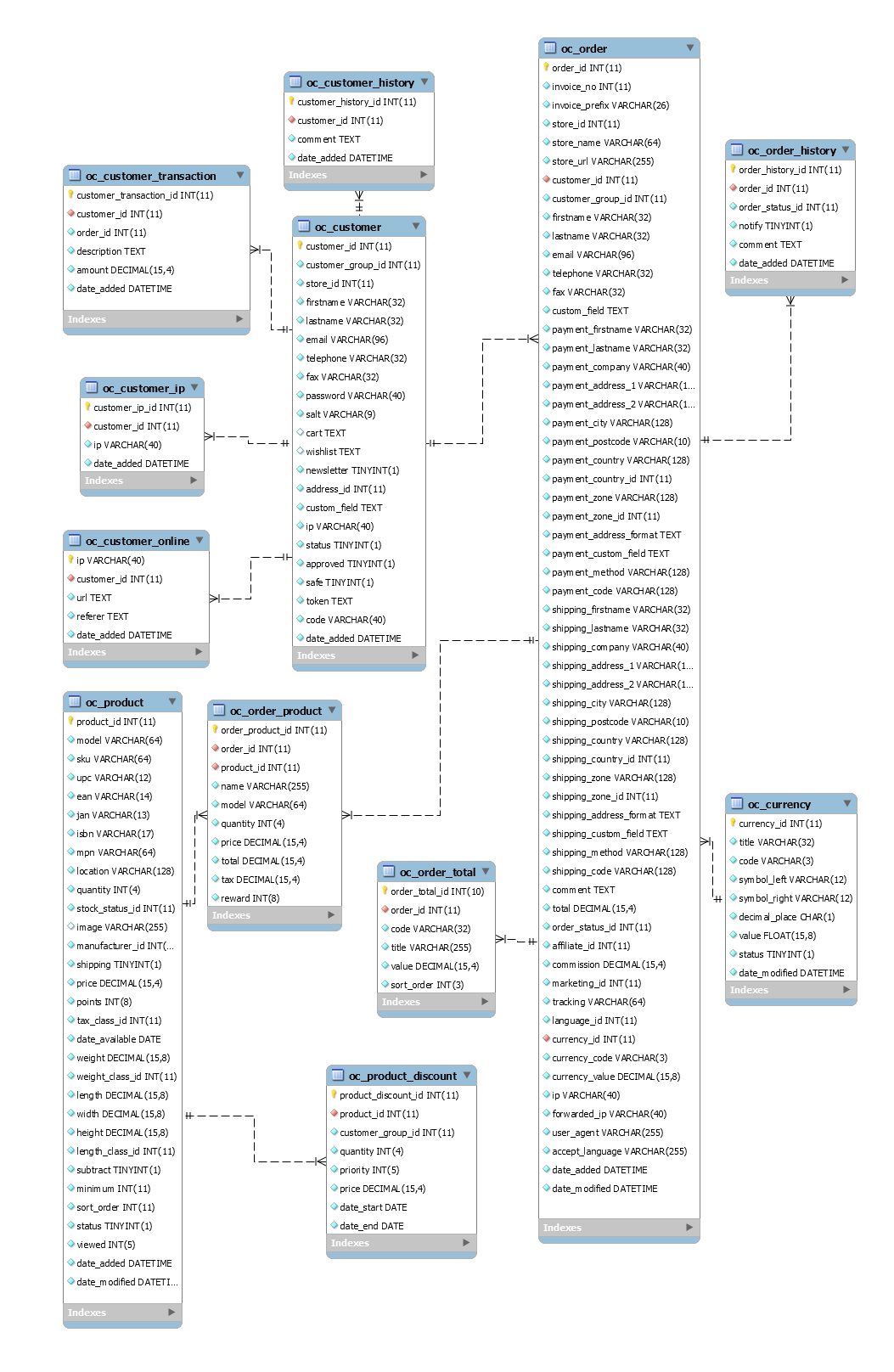
Tabela . Harmonogram prac

|  |  |
| --- | --- |
| Data | Opis |
| 15.03.16 | Przygotowanie założeń projektowych oraz listy wymagań funkcjonalnych i niefunkcjonalnych. |
| 29.03.16 | Wdrożenie podstawowej wersji aplikacji OpenCart. |
| 12.04.16 | Instalacja i konfiguracja wybranych modułów. |
| 26.04.16 | Wdrożenie wybranych zabezpieczeń. |
| 10.05.16 | Testy sklepu internetowego |
| 24.05.16 | Naprawa wykrytych luk w bezpieczeństwie programu. |
| 07.06.16 | Oddanie projektu. |

# Baza danych

Poniższy schemat bazy danych został wygenerowany programem MySQl Workbench na podstawie bazy danych aplikacji OpenCart. Baza danych posiada 126 tabel. Ze względu na złożoność bazy danych, poniżej przedstawiony został wyłącznie jej najważniejszy fragment.

Na rys. 1 przedstawiono fragment bazy danych, który zawiera tabele ważne z punktu widzenia sklepu internetowego oraz część ich relacji. Tabele te przechowują informacje na temat zamówień, produktów oraz klientów, a także innych danych powiązanych z nimi - historia zamówień, dane o rabatach czy dane o adresach IP klienta.

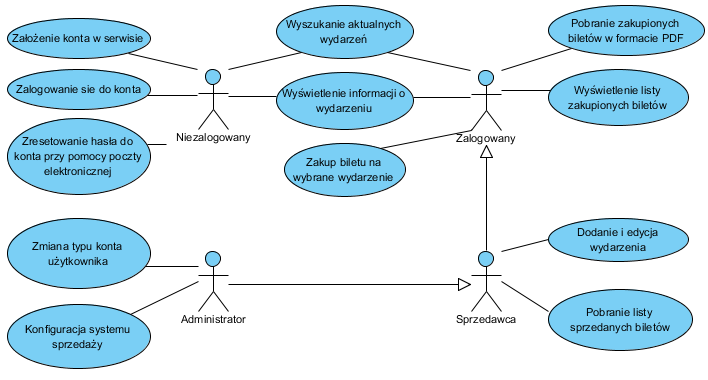


Rysunek . Schemat przedstawiający niektóre tabele bazy danych oraz ich relacje

# Diagramy UML

## Diagram przypadków użycia

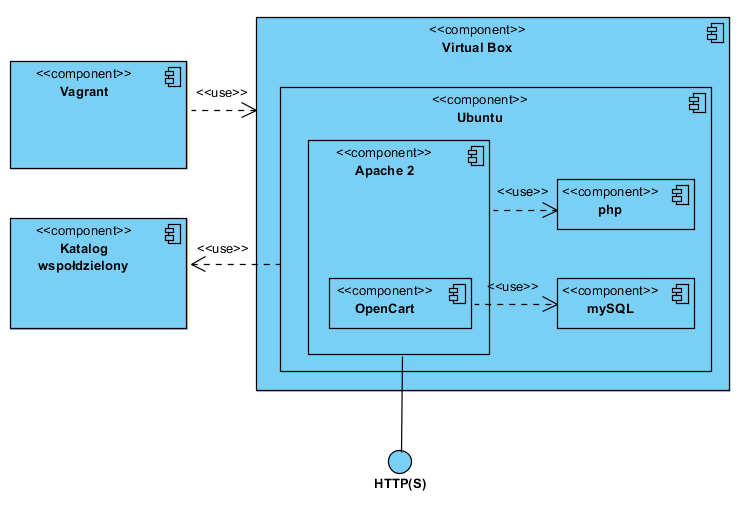
Rysunek 2 obrazuje, przy pomocy notacji UML, dane zawarte w wymaganiach funkcjonalnych opisanych w tabeli 1.



Rysunek . Diagram przypadków użycia

## Architektura serwera

Rysunek 3 przedstawia diagram UML przedstawiający architekturę serwera aplikacji.



Rysunek . Architektura serwera

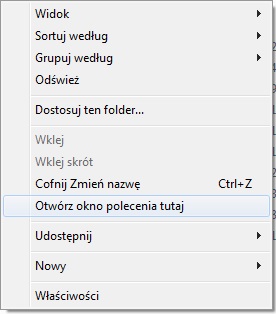
# Konfiguracja sklepu

## Instalacja sklepu

Sklep instalowany jest w systemie Ubuntu przy użyciu środowiska Vagrant, które pozwala na instalacje, aktualizacje i uruchamianie maszyn wirtualnych w sposób zautomatyzowany.

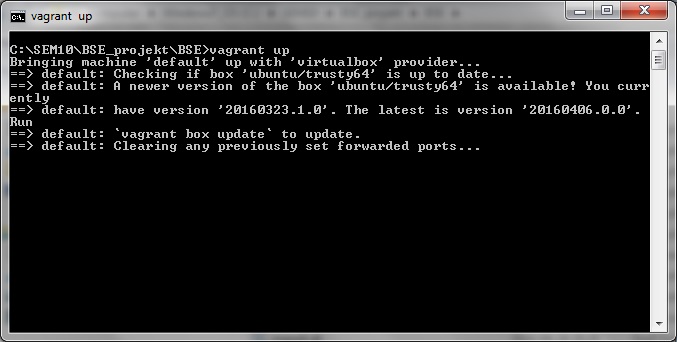
W pierwszej kolejności należy zainstalować środowisko Vagrant, a także program VirtualBox, oba w najnowszych wersjach. Użytkownik musi pobrać odpowiednie repozytorium z serwisu Github, a następnie wypakować je do dowolnego katalogu.

Po otwarciu katalogu, w którym zostało wypakowane repozytorium należy przytrzymać klawisz *shift*, a następnie kliknąć prawym przyciskiem myszy. Po otwarciu się menu kontekstowego należy wybrać opcję *Otwórz okno polecenia tutaj*, co zostało przedstawione na rysunku 4.



Rysunek . Otwieraniu menu kontekstowego

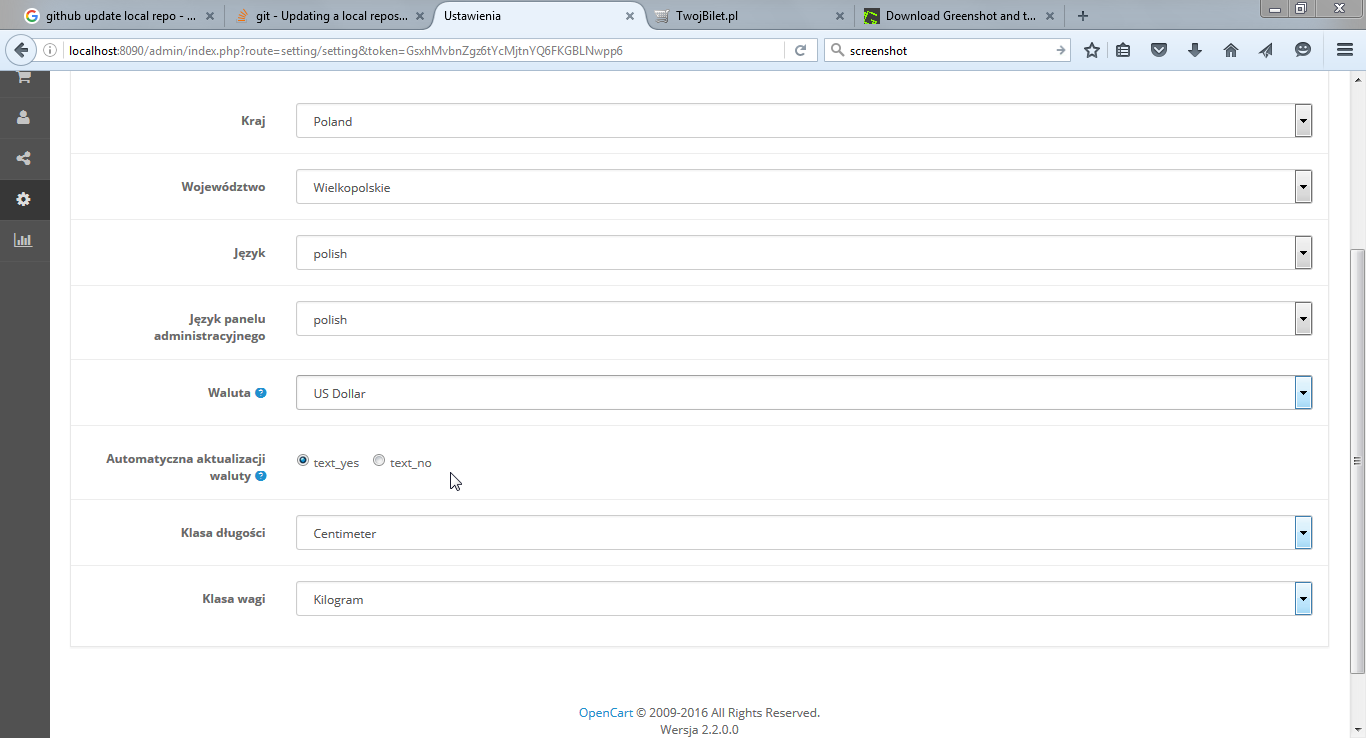
Następnie należy wpisać w oknie poleceń komendę *vagrant up* i poczekać, aż program zakończy instalację oraz uruchomi maszynę wirtualną wraz ze sklepem. Po zainstalowaniu sklepu każdorazowe uruchomienie maszyny wirtualnej musi zostać wykonane w sposób identyczny - przy użyciu komendy *vagrant up*. Przykładowe uruchomienie przy użyciu komendy zostało przedstawione na rysunku 5.



Rysunek . Uruchamianie lub instalacja maszyny wirtualne przy użyciu środowiska Vagrant

## Język polski

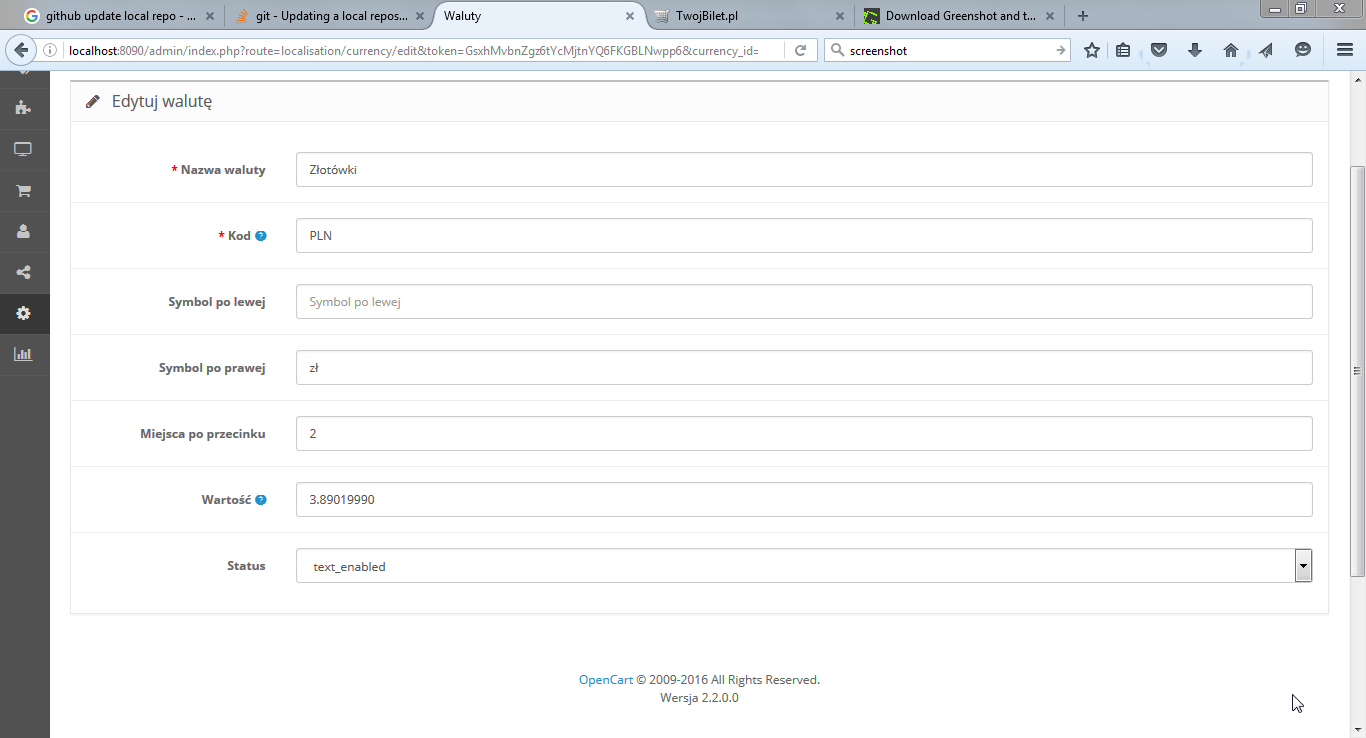
W sklepie został zainstalowany język polski, który nie jest dostępny w wersji podstawowej aplikacji.

1. Do katalogów aplikacji *admin/language/* należy skopiować pliki z tłumaczeniem do katalogu *admin/language/pl-PL*.
2. Do katalogów aplikacji *catalog/language/* należy skopiować pliki z tłumaczeniem do katalogu *catalog/language/pl-PL*.
3. W ustawieniach sklepu *Settings->Store->Local*, w polach *Language*, *Administrator language* należy wybrać nowododany język polski - *polish* co zostało pokazane na rysunku 4.

Rysunek . Wybór języka

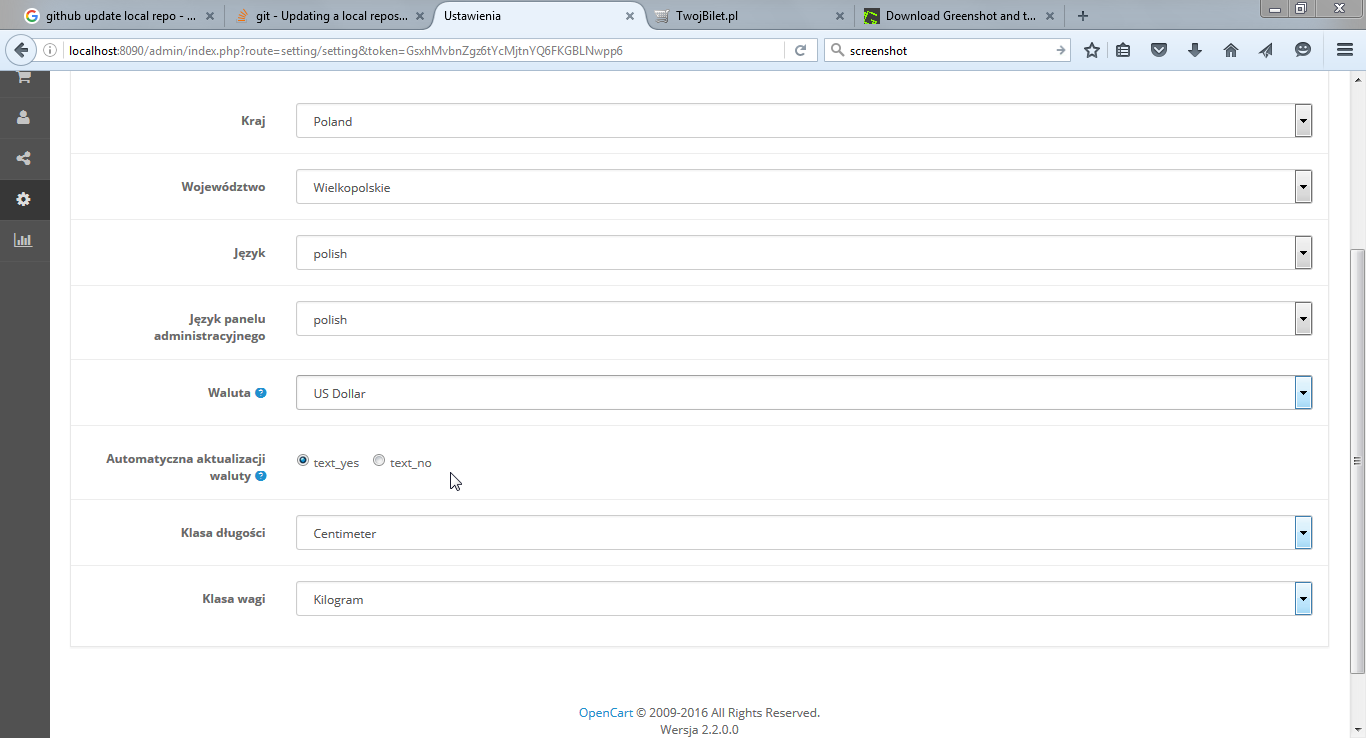
## Waluta

W podstawowej konfiguracji sklepu dostępne są trzy waluty: euro, dollary oraz funty. Aby zmienić walutę na inną niż standardowa należy najpierw skonfigurować nową walutę w zakładce *Ustawienia->Lokalizacja->Waluta*, co widoczne jest na rysunku 5.



Rysunek . Konfiguracja nowej waluty

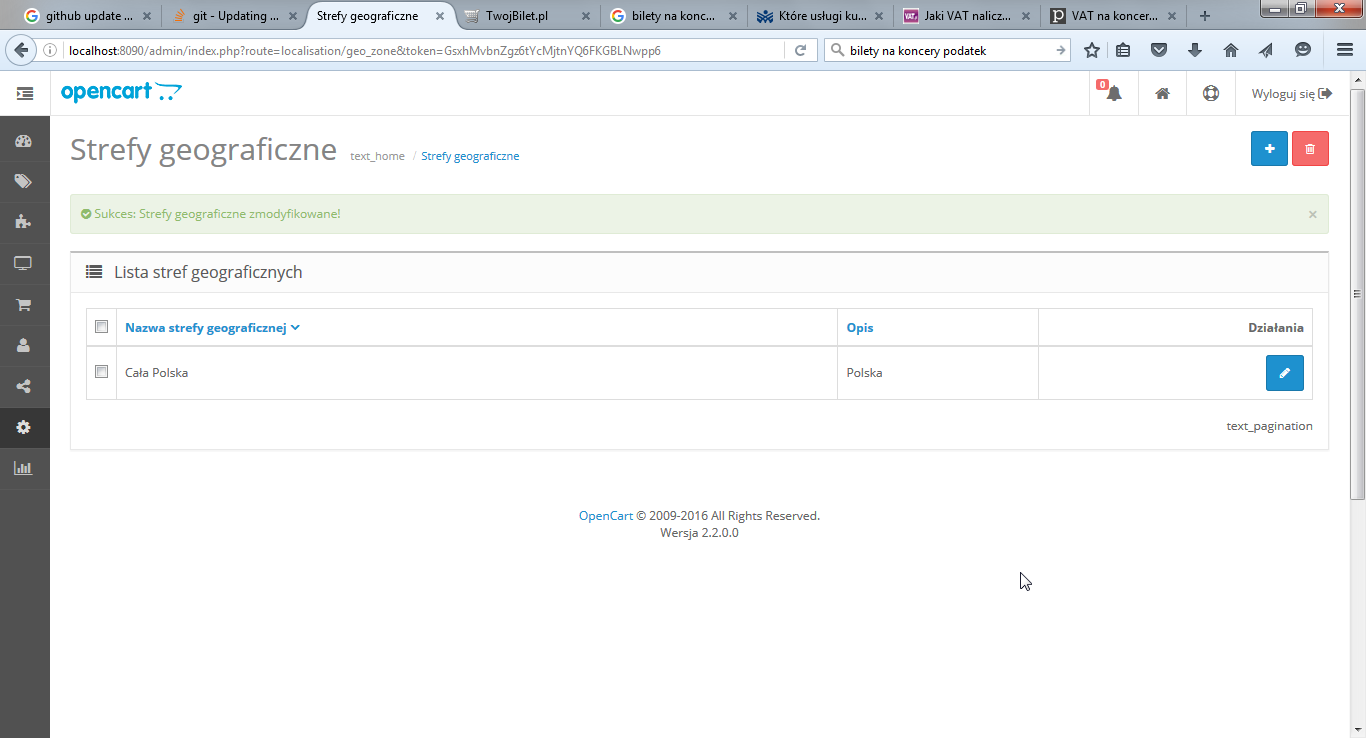
Następnie w celu zmiany waluty używanej w sklepie należy przejść do zakładki *Ustawienia->Sklep->Lokalne* i w polu waluta wybrać odpowiednią opcje - złotówki, co zostało przedstawione na rysunku 6.



Rysunek . Wybór waluty

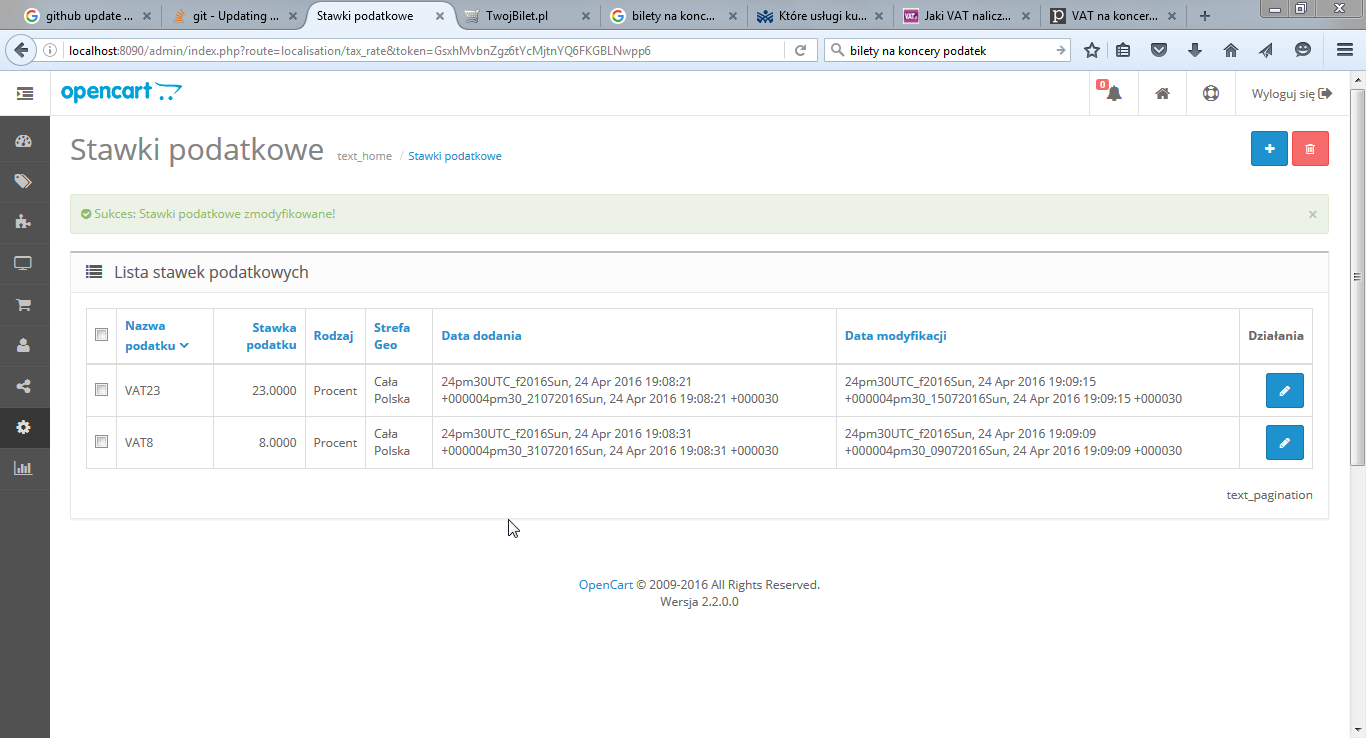
## Podatki

W celu skonfigurowanie odpowiednich podatków najpierw należy ustawić strefy geograficzne, których dotyczyć będą wprowadzane podatki. W celu dodania nowej strefy należy przejść do zakładki *Ustawienia->Lokalizacja->Strefy geograficzne*. Lista przykładowych stref widoczna jest na rysunku 7.



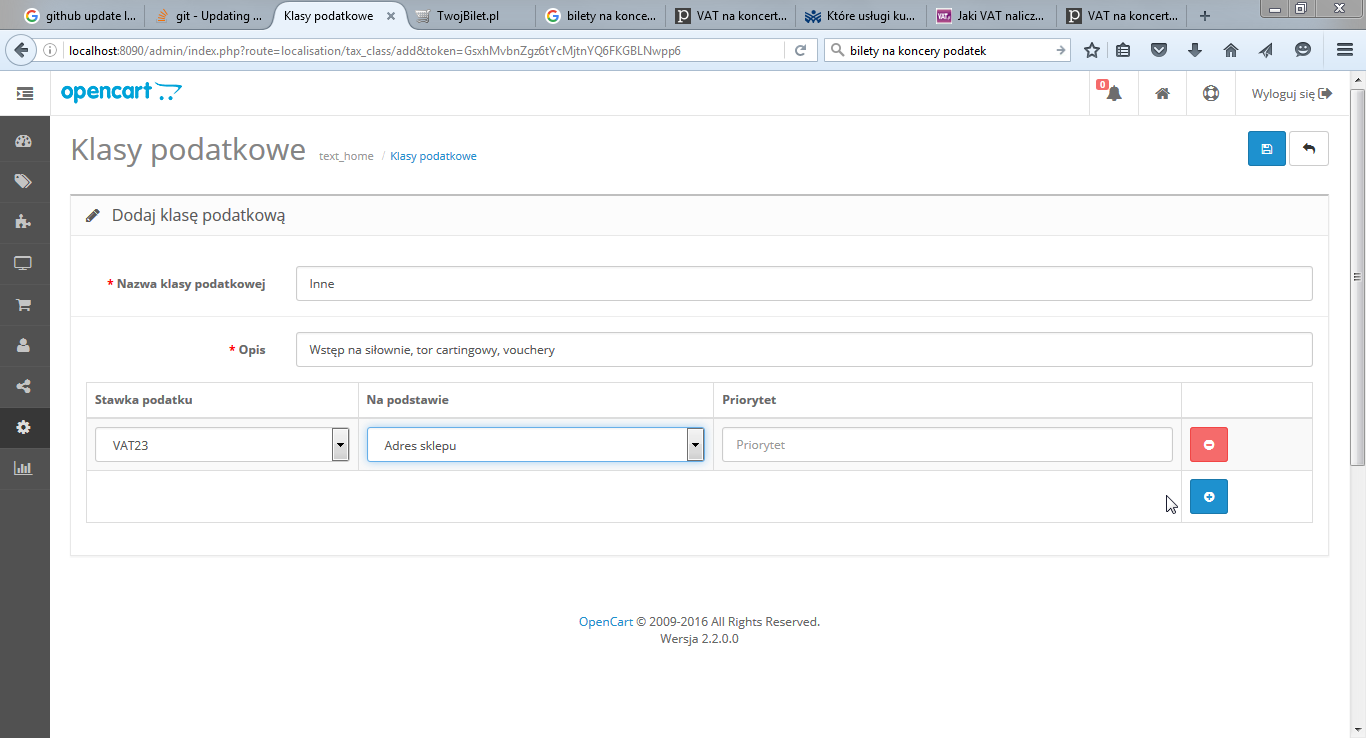
Rysunek . Lista skonfigurowanych stref geograficznych

Po wprowadzeniu odpowiednich stref geograficznych należy dodać stawki podatków. W przypadku sprzedaży biletów należy wprowadzić dwie stawi podatku VAT - 8% oraz 23%. Można to zrobić wybierając z menu *Ustawienia->Lokalizacja->Podatki->Stawki podatkowe.* Lista stawek po skonfigurowaniu widoczna jest na rysunku 8.



Rysunek . Lista stawek podatkowych po skonfigurowaniu

Ostatnim krokiem przy konfiguracji podatków jest wprowadzenie klas podatkowych. W przypadku sprzedaży biletów na koncerty, występy kabaretowe oraz widowiska artystyczne podatek VAT wynosi 8%. W przypadku sprzedaży voucherów, biletów do siłowni czy na tor kartingowy podatek ten wynosi 23%. Konfiguracja klasy podatkowej została przedstawiona na rysunku 9.



Rysunek . Konfigurowanie klasy podatkowej

## Ceny brutto

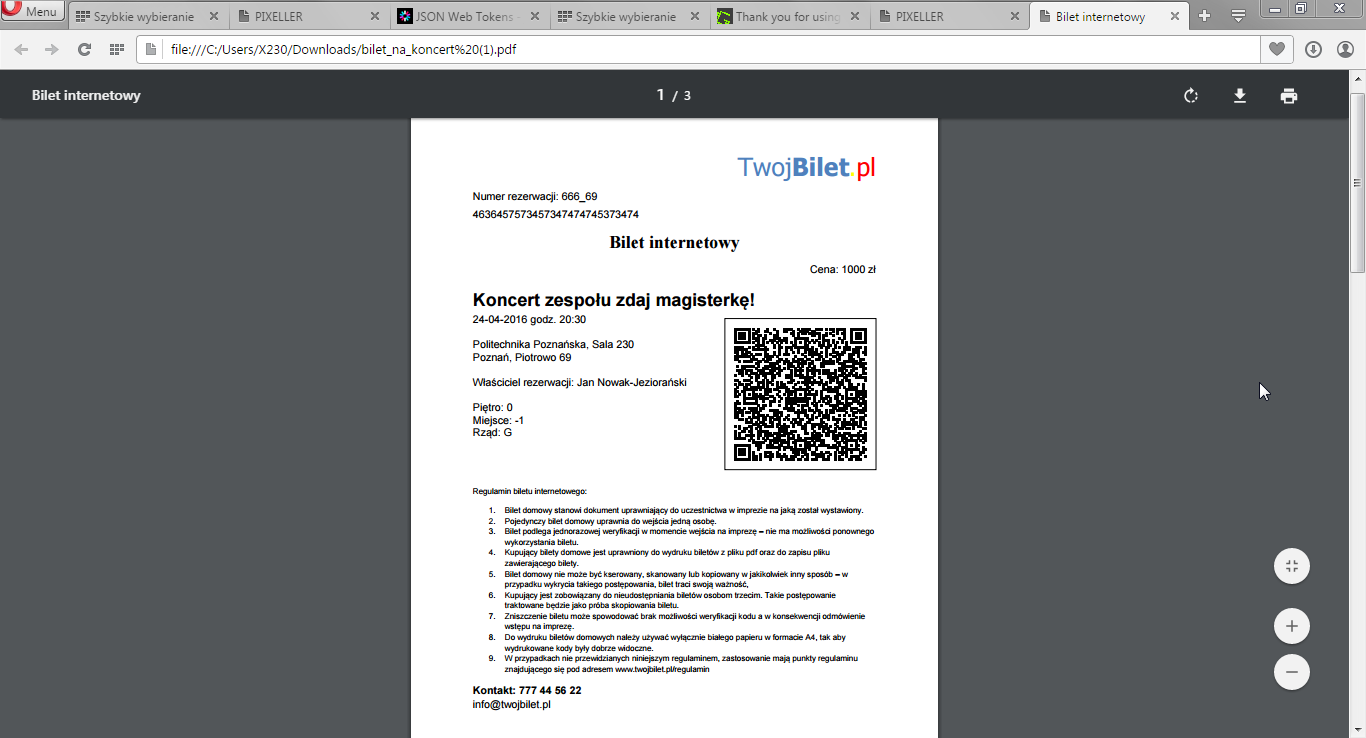
Domyślnie wprowadzając produkt do sklepu internetowego zbudowanego na aplikacji *OpenCart* należy podać cenę netto. Jest to dość problematyczne przy cenach biletów, gdyż organizatorzy podają ceny biletów jako ceny brutto. Aby móc wprowadzać towary w cenach brutto należy zmodyfikować kod aplikacji OpenCart.

## Certyfikat SSL

Dla sklepu został wygenerowany odpowiedni certyfikat SSL. W celu skorzystania z tego certyfikatu został on dodany do konfiguracji serwera *Apache*. Po skonfigurowaniu serwera WWW, w sklepie włączona została opcja korzystania z połączenia bezpiecznego wykorzystującego wygenerowany certyfikat SSL.

# Bilety

Do sklepu internetowego został napisany dodatek pozwalający na generowanie biletów w formacie PDF. Przykładowy bilet wygenerowany za pomocą tego narzędzia został przedstawiony na rysunku 10.



Rysunek . Przykładowy bilet wygenerowany ze sklepu internetowego

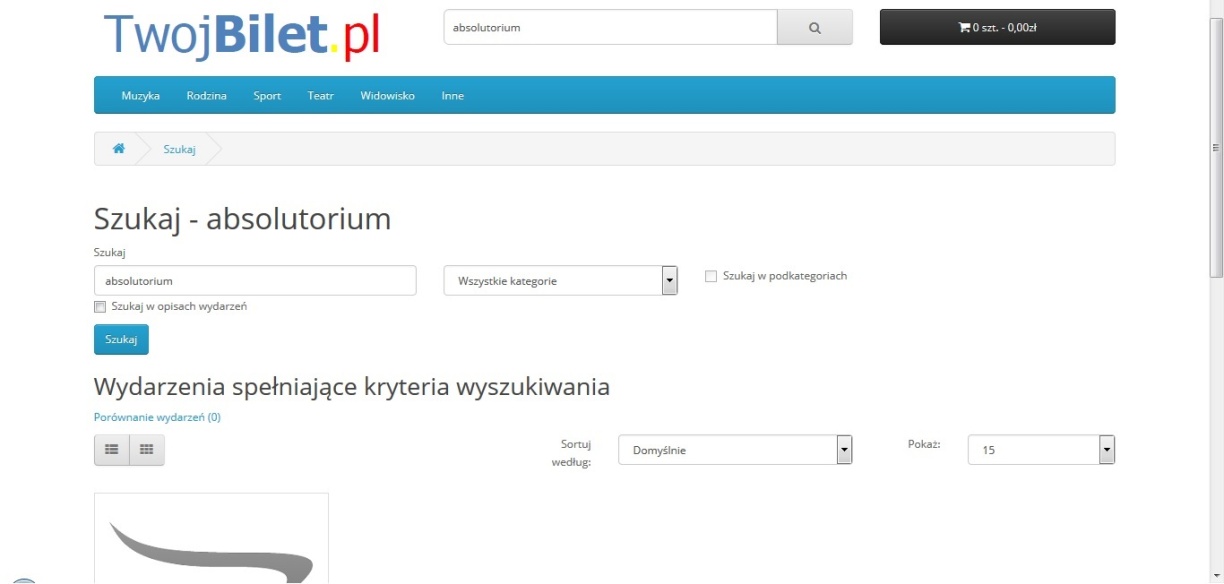
Na bilecie zostały umieszczone takie informacje jak:

* numer rezerwacji,
* numer biletu,
* kod QR,
* nazwa, adres i data wydarzenia,
* numer zamówienia,
* wykupione miejsca,
* regulamin korzystania z biletu,
* informacje kontaktowe do sklepu.

# Użytkowanie sklepu

## Wyszukiwanie wydarzeń

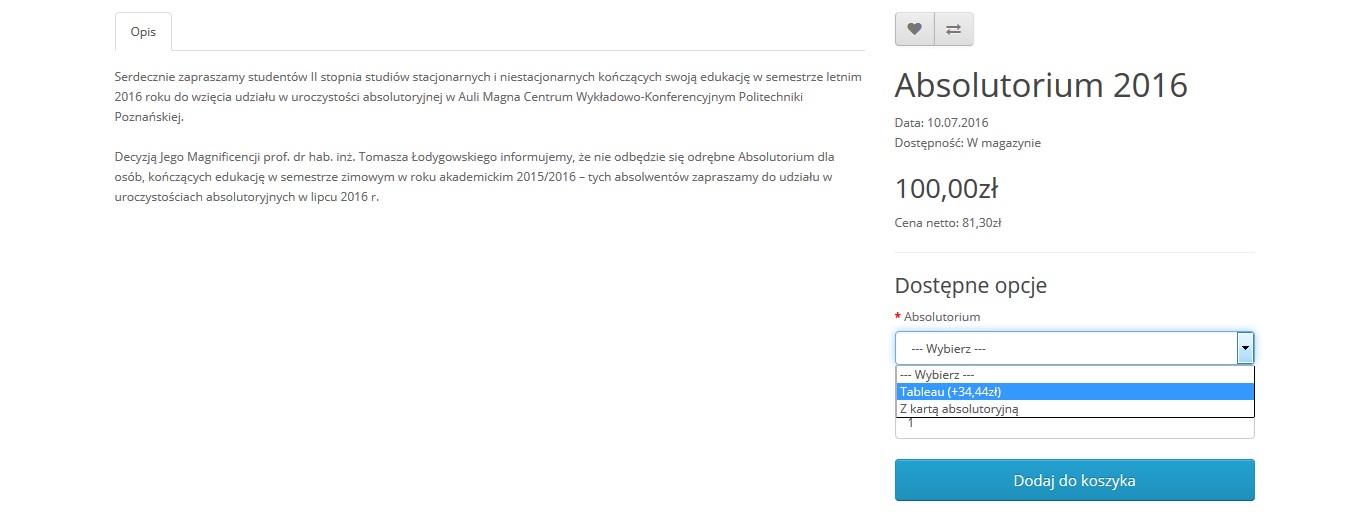
W celu wyszukania interesującego użytkownika wydarzenia, może on skorzystać z zakładki kategorii wybierając jeden z kilku rodzajów imprez, bądź skorzystać z wyszukiwarki. Wyszukiwarka pozwala na odnalezienie wydarzeń w sposób bardziej szczegółowy. Przykład wyszukiwania został przedstawiony na rysunku 13.



Rysunek . Wyszukiwanie wydarzenia

## Wybór opcji wydarzenia

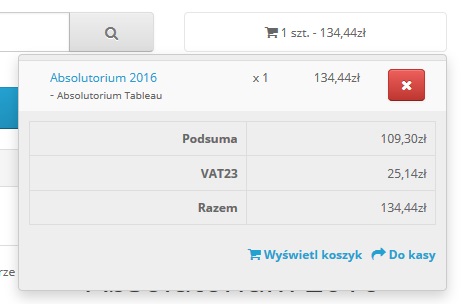
Przy zakupie wydarzenia dostępne są różne opcje pozwalające na wybranie np. sektora w którym użytkownik chciałby zająć miejsce czy dodatkowych gadżetów związanych z wydarzeniem. W zależności od tego czy dana opcja jest płatna, cena wydarzenia może wzrosnąć. W zależności od wydarzenia wybór opcji może być narzucony od górnie przez organizatorów. Przykładowy wybór opcji został przedstawiony na rysunku 14.



Rysunek . Wybór dodatkowych opcji wydarzenia

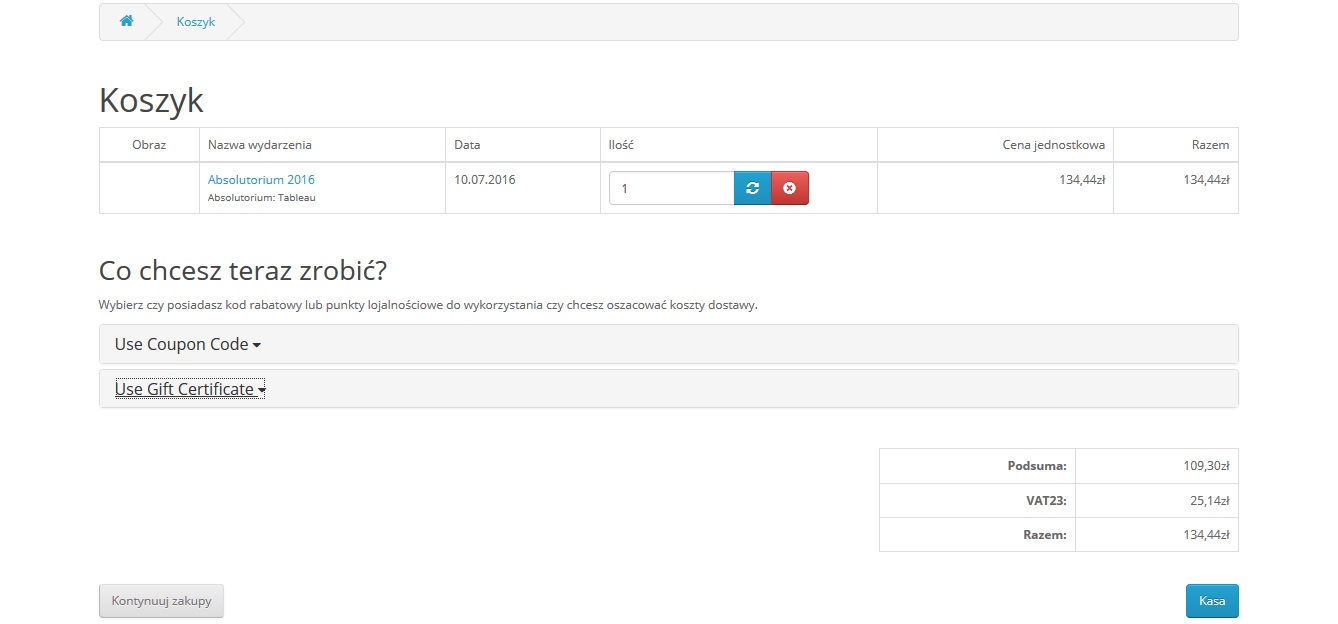
## Wyświetlanie koszyka

Użytkownik w każdej chwili może przejrzeć swój koszyk. Aby to uczynić może wybrać wizerunek koszyka przedstawiony w prawym, górnym rogu strony a następnie kliknąć na przycisk z ilością elementów koszyka. Zostało to przedstawione na rysunku 15.



Rysunek . Podgląd koszyka

W przypadku, gdy istnieje potrzeba zmiany zawartości koszyka - usunięcia wydarzenia, zwiększenia ilości biletów, można tego dokonać klikając opcje *Wyświetl koszyk* znajdującą się w omawianym powyżej podglądzie koszyka. Pełny widok koszyka został przedstawiony na rysunku 16.

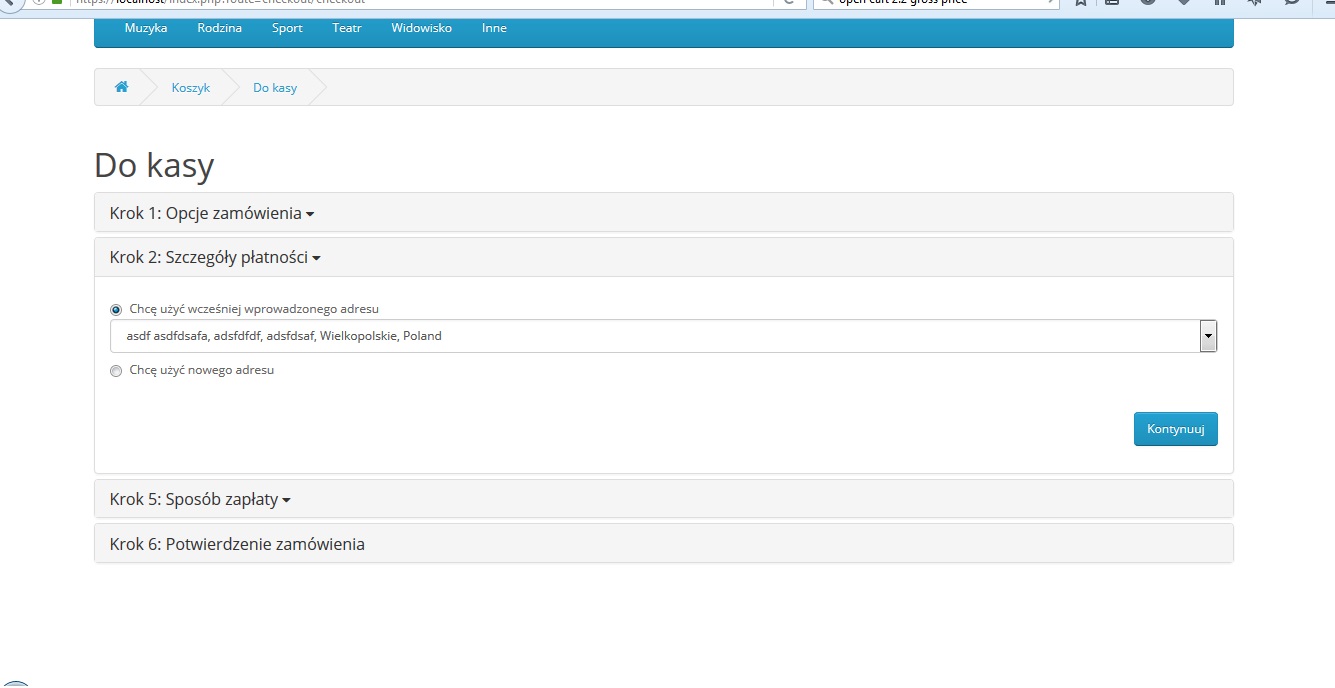


Rysunek . Pełny widok koszyka

## Zakup biletów

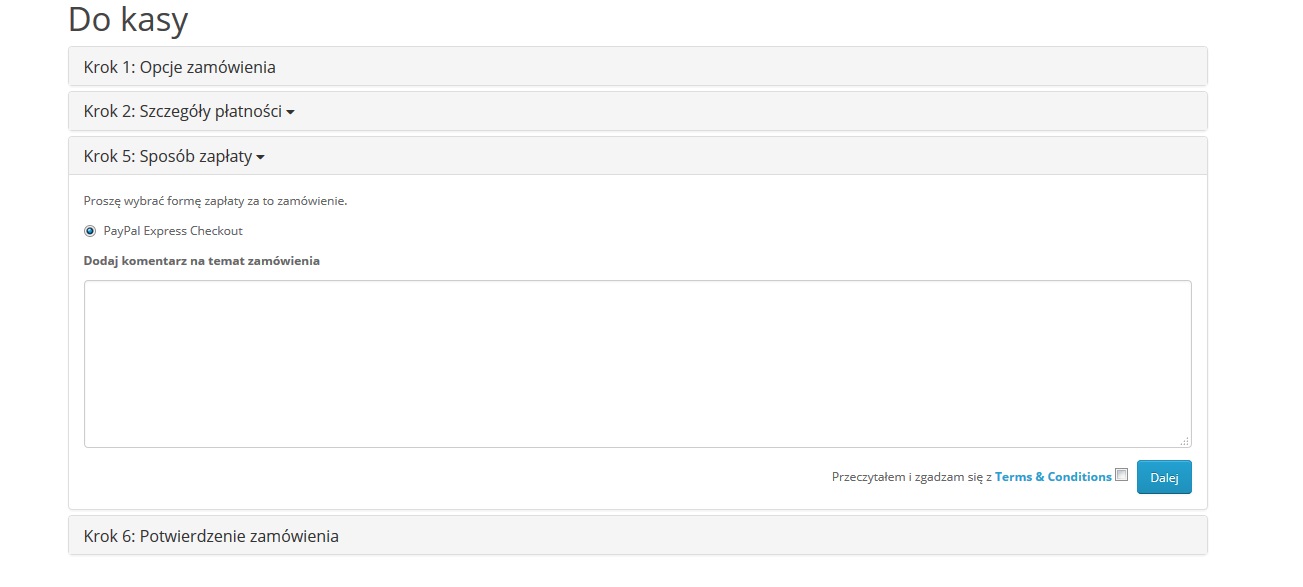
Aby zakupić bilety na wybrane wydarzenia należy wybrać opcję *Kasa* w pełnym widoku koszyka lub opcję *Do kasy* w podglądzie koszyka.

Po wybraniu tych opcji użytkownik zostanie przeniesiony do strony związanej z wyborem adresu zatytułowanej *Szczegóły płatności*. Istnieje możliwość wprowadzenia nowych danych użytkownika (jednorazowo) lub wybrania tych, które zostały podane podczas rejestracji. Strona ta przedstawiona została na rysunku 17.



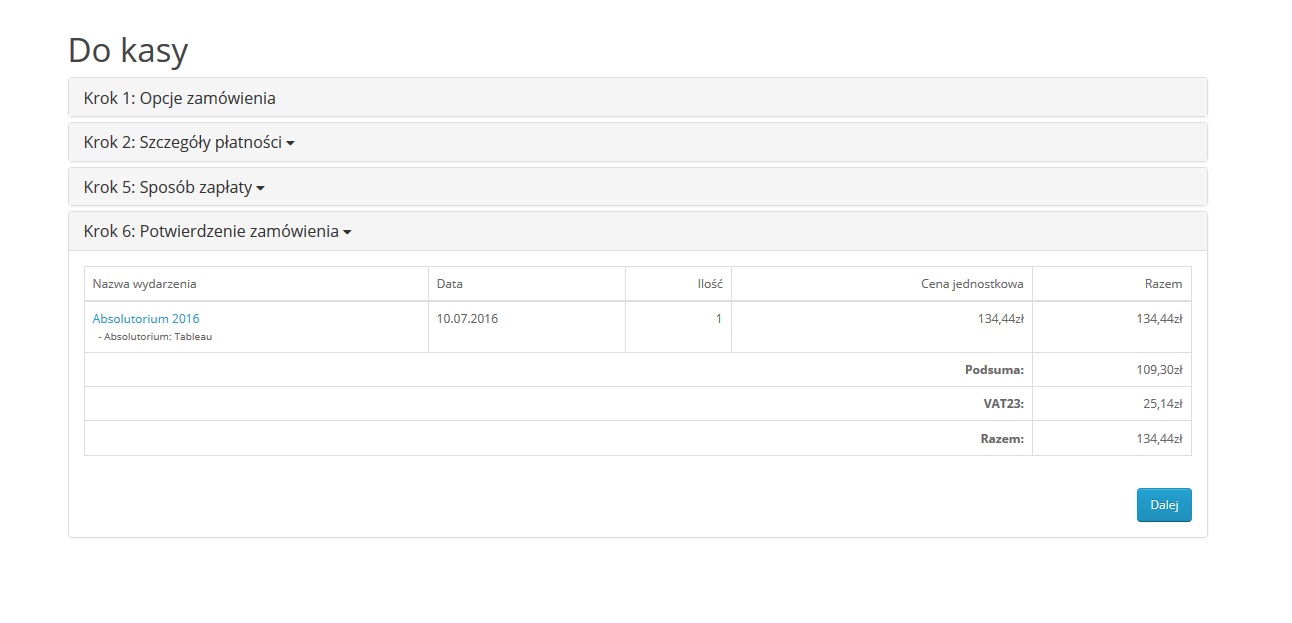
Rysunek . Wybór danych właściciela biletu

W kolejnym kroku użytkownik wybiera sposób zapłaty. W chwili obecnej istnieje wyłącznie metoda płatności przez serwis *PayPal*. Użytkownik ma także możliwość pozostawienia informacji dla właściciela sklepu poprzez pole *Dodaj komentarz do zamówienia.* Przed przejściem dalej użytkownik musi zaakceptować warunku sprzedaży. Zostało to przedstawione na rysunku 18.



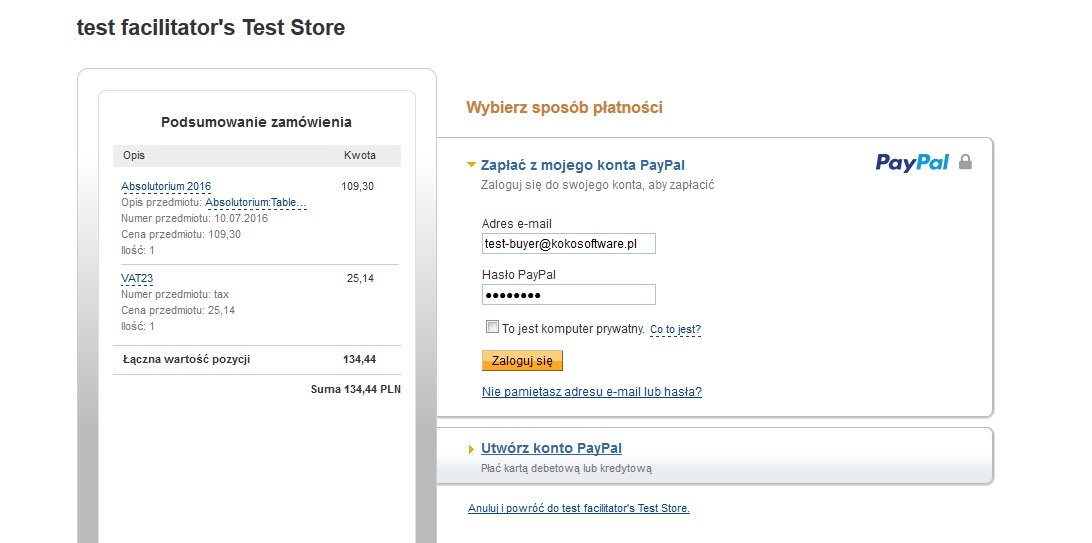
Rysunek . Wybór metody płatności i pozostawianie komentarza

Ostatnim krokiem wykonywanym na stronie przed dokonaniem zakupu jest sprawdzenie poprawności zamówienia w oknie podsumowania. Okno to zostało przedstawione na rysunku 19.



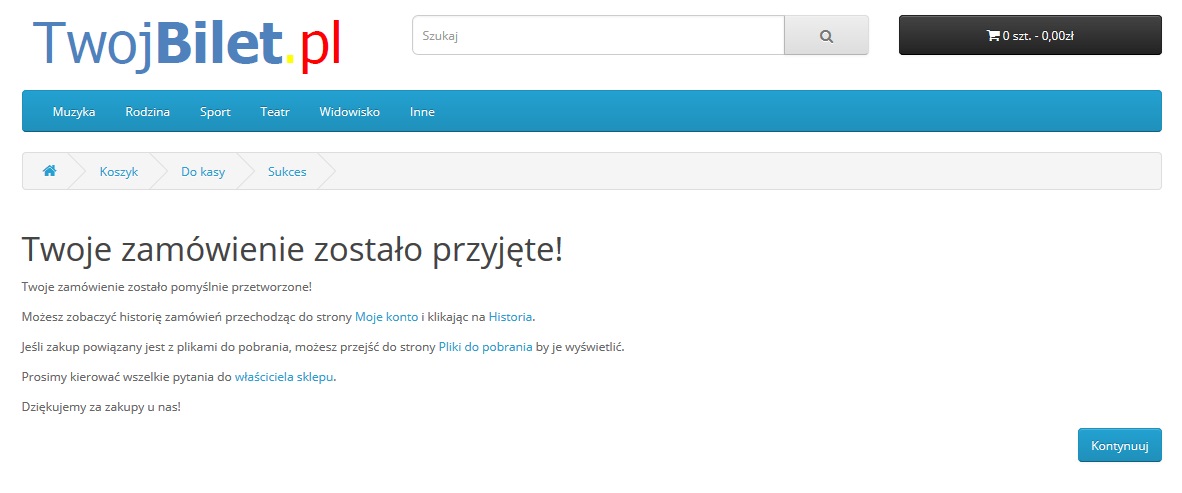
Rysunek . Podsumowanie zamówienia

Po kliknięciu przycisku *Dalej* użytkownik zostanie przeniesiony na stronę płatności *PayPal,* gdzie należy dokonać płatności. Przykład testowej strony logowania przy płatności za zamówienie został przedstawiony na rysunku 20.



Rysunek . Płatność przy użyciu serwisu PayPal

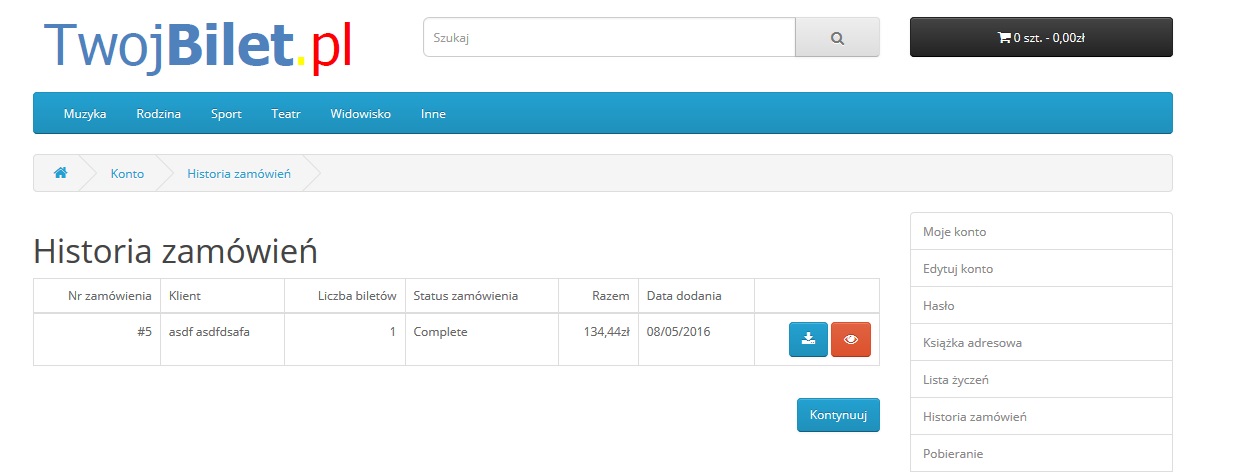
Po dokonaniu płatności użytkownik zostanie ponownie przeniesiony do strony sklepu z informacją o dokonaniu płatności za wydarzenie. Zostało to przedstawione na rysunku 21.



Rysunek . Potwierdzenie przyjęcia zamówienia

## Pobieranie biletów

Po dokonaniu płatności możemy przejść do okna *Historia zamówień*. W tym oknie istnieje możliwość pobrania biletu na wydarzenie w zależności od tego czy płatność została zaksięgowana przez sklep. Jeśli tak, przy zamówieniu pokazany jest przycisk umożliwiający pobranie biletu na wydarzenie w formacie PDF. Przykładowe okno *Historii zamówień* zostało przedstawione na rysunku 22.



Rysunek . Okno Historii zamówień z przyciskiem do pobrania biletu