Uniwersytet Śląski  
**Wydział Informatyki i Nauki o Materiałach**

**Projekt sklepu internetowego Xtra Shop.**

**Wykonali:**Arkadiusz Siuda  
Mateusz Samuel  
Jakub Stolarczyk  
Witold Pomazanka

Spis treści

[1 Cel systemu 2](#_Toc484708509)

[2 Podział prac 2](#_Toc484708510)

[3 Analiza finansowa 2](#_Toc484708511)

[4 Wykorzystane technologie 2](#_Toc484708512)

[5 Wymagania systemu 3](#_Toc484708513)

[6 Wygląd aplikacji 3](#_Toc484708514)

[7 Przypadki użycia 9](#_Toc484708522)  
[8 Diagram czynności 11](#_Toc484708522)

[9 Diagram sekwencji 13](#_Toc484708523)

[10 Diagram Klas 17](#_Toc484708533)

[11 Diagram bazy danych 20](#_Toc484708534)

[12 Diagram stanów 21](#_Toc484708535)

### 1 Cel systemu

Celem tego systemu jest stworzenie oprogramowania, które umożliwi dokonywanie zakupów przez Internet. Użytkownicy systemu, będą mogli przeglądać stronę sklepu internetowego, a także dodać produkty  
 do „koszyka”. Następnie finalizując swoją sesję, zakupią produkty po wykonaniu przelewu internetowego. System jest aplikacją internetową.

### 2 Podział prac

|  |  |
| --- | --- |
| **Jakub Stolarczyk** | Analiza finansowa Diagram klas Diagram sekwencji |
| **Arkadiusz Siuda** | Prezentacja aplikacji  Wymagania systemu |
| **Witold Pomazanka** | Programowanie aplikacji  Diagram przypadków użycia Diagram czynności |
| **Mateusz Samuel** | Diagram życia encji Diagram stanów |

Wszyscy autorzy projektu są odpowiedzialni za stworzenie dokumentacji.

### 3 Analiza finansowa

Poniższa tabela przestawia przybliżone koszty systemu, symulując analizę finansową dla przyszłego klienta.

|  |  |  |
| --- | --- | --- |
| System | Abonament za domenę | 56 zł na rok |
| Responsive Web Desing | 960 zł |
| Grafiki | 800 zł |
| Logo | 225 zł |
| Główna funkcjonalność sytemu | 5000 zł |
| Przeprowadzenie testów | 400 zł |
| Stworzenie dokumentacji | 500 zł |
| Baner reklamowy | 300 zł |
| Suma kosztów: | | 8241 zł |

Koszty stworzenia systemu oraz wdrożenie go na rynek szacuje się na 8241zł

### 4 Wykorzystane technologie

Sklep internetowy „Xtrashop” jest aplikacją internetową napisaną z użyciem dwóch profesjonalnych szkieletów do tworzenia oprogramowania: Spring Framework oraz AngularJS v. 1. Pierwsze rozwiązanie bazuje na języku programowania Java w wersji ósmej, kolejne opiera się o manipulacje wyświetlaną treścią DOM (Document Object Model) w języku JavaScript. Do obsługi bazy danych zostało wykorzystane wbudowane rozwiązanie plikowe H2 Database, a wypełnieniem program migracyjny Flyway. Rozwiązaniem mapowania relacyjno-obiektowego jest szkielet Hibernate – jest to implementacja Java Persistence API. Do poprawy widoków został użyty framework Bootstrap a całość wyświetlanej treści   
jest osadzona w języku znaczników HTML. Dostarczone rozwiązanie jest zgodne z konwencją MVC (Model View Controller) a do komunikacji pomiędzy warstwą widoku a kontrolera użyto rozwiązania REST.

### 5 Wymagania systemu

* Szybki i łatwy dostęp poprzez przeglądarkę, również z poziomu telefonów komórkowych
* Rejestracja użytkowników
* Możliwość nadawania uprawnień moderatora zwykłym użytkownikom
* Podział przedmiotów na kategorie
* Dodawanie przez moderatorów oraz administratorów nowych produktów
* Usuwanie produktów
* Aktualizację produktów
* Tworzenie kolekcji produktów, zwanych potocznie „koszykiem”
* Zakup wybranych przedmiotów

### 6 Wygląd aplikacji

### *Rys. 1 Ekran logowania*

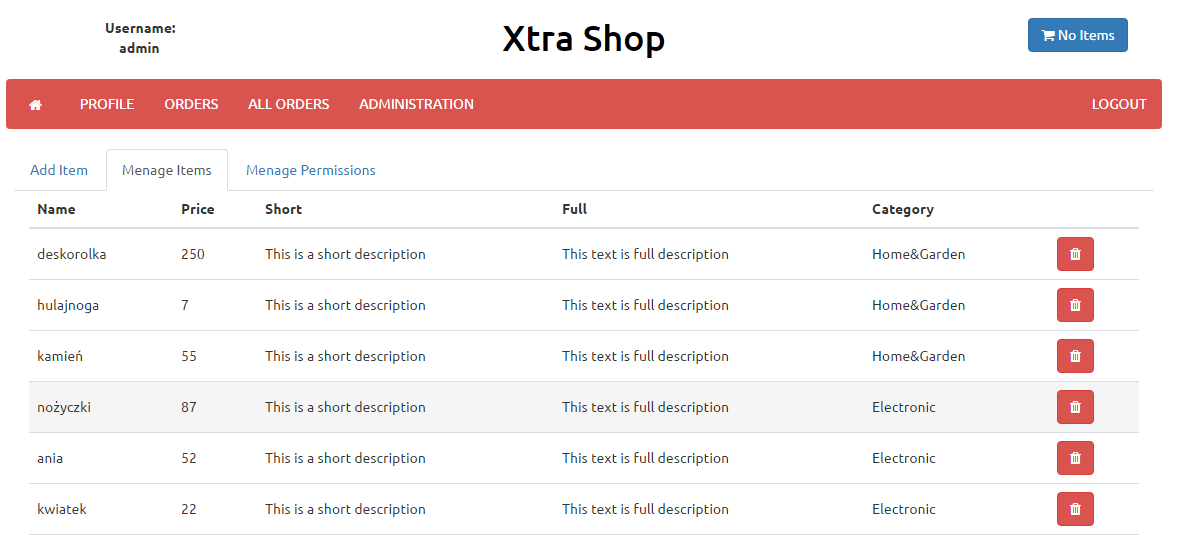
### *Rys. 2 Ekran rejestracji z widoczną walidacją danych* *Rys. 3 Widok główny sklepu. Przedmioty wraz z kategoriami*

### *Rys. 4 Koszyk*

### *Rys. 5 Widok zamówień złożonych, potwierdzonych oraz odrzuconych*

### *Rys. 6 Ekran dodawania przedmiotów*

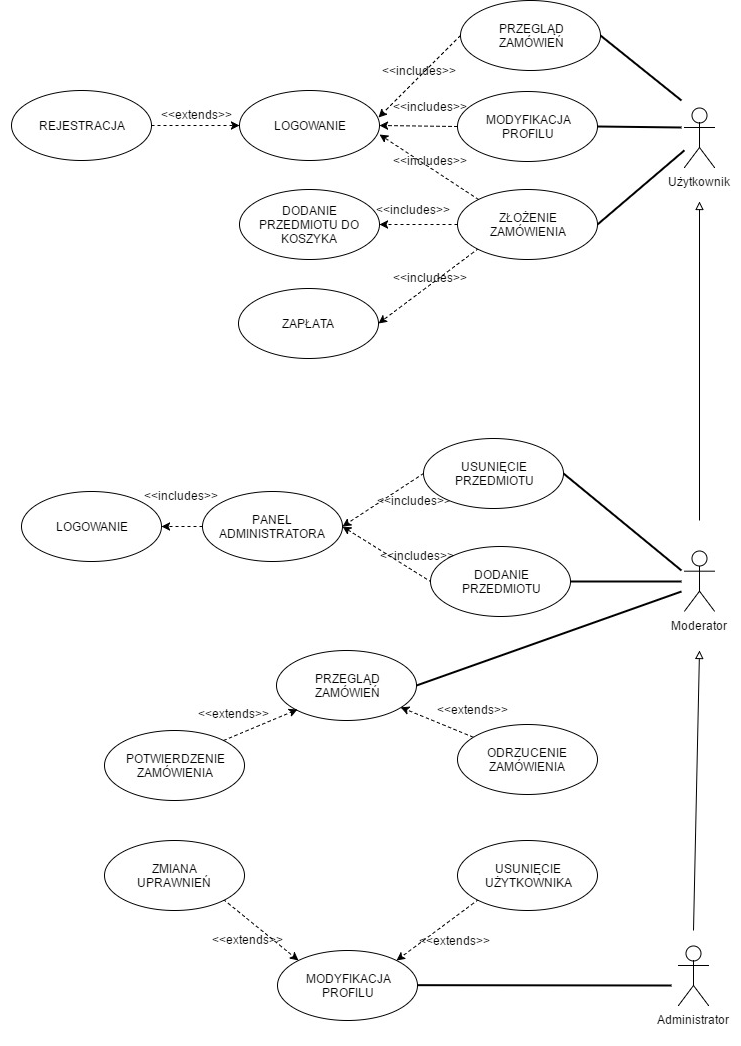
### *Rys. 7 Zarządzanie użytkownikami i rolami*



### *Rys. 8 Ekran usuwania przedmiotów*

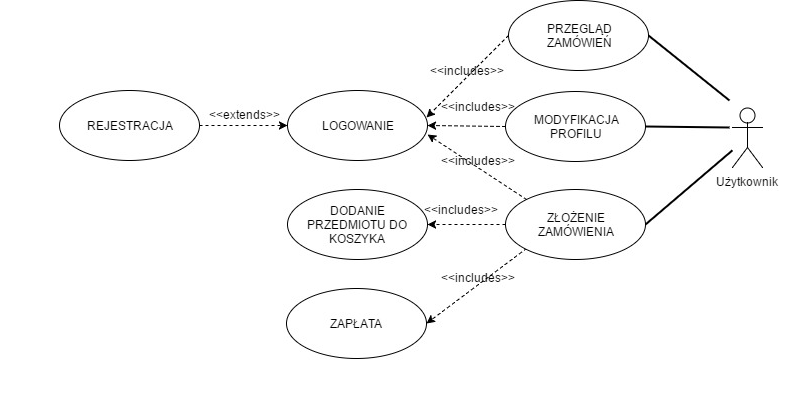
### 7 Przypadki użycia

Diagram przypadków użycia (ang. use case) jest wykorzystywany do opisu funkcjonalności systemu  
z perspektywy użytkowników. Definiuje ich uogólnione operacje i w prosty sposób opisuje działania   
które każdy z aktorów może wykonywać.

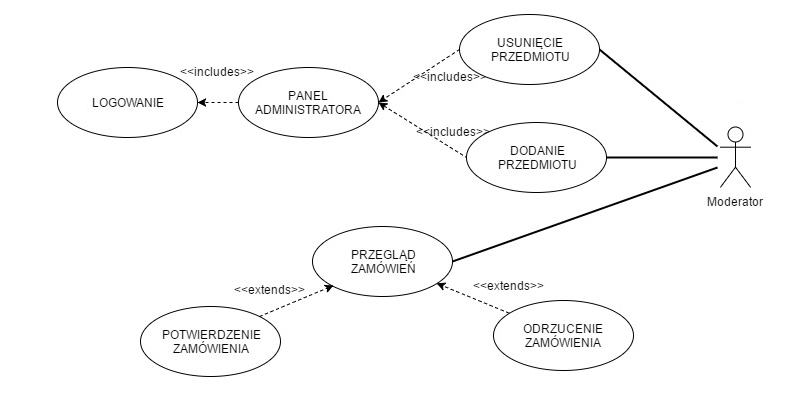


W przedstawionym systemie wyróżniamy trzech aktorów:

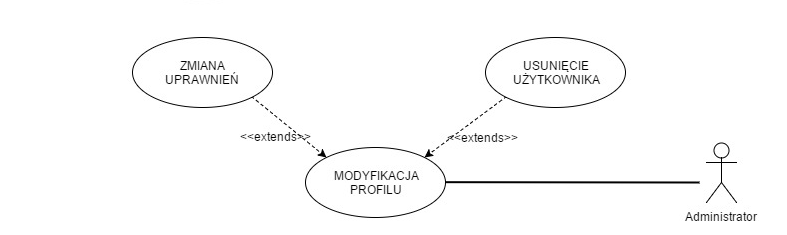
**Użytkownik** – bazowa rola systemowa po której dwie kolejne – moderator i administrator – dziedziczą operacje. Relacja dziedziczenia (uogólnienia) tworzy hierarchię działań od ogólnych do bardziej szczegółowych, pozwala także wyłączyć do roli bazowej części wspólne operacji. By korzystać z systemu użytkownik musi się zarejestrować, następnie po zalogowaniu otrzymuje możliwości złożenia zamówienia poprzez dodanie przedmiotu/ów do koszyka a przy akceptacji wyświetlane są dane do przelewu. Złożone zamówienia widoczne są w osobnej zakładce wraz z podsumowaniem i obecnym stanem.   
Ostatnią z możliwych operacji jest modyfikacja profilu, gdzie można edytować wszystko oprócz loginu użytkownika.



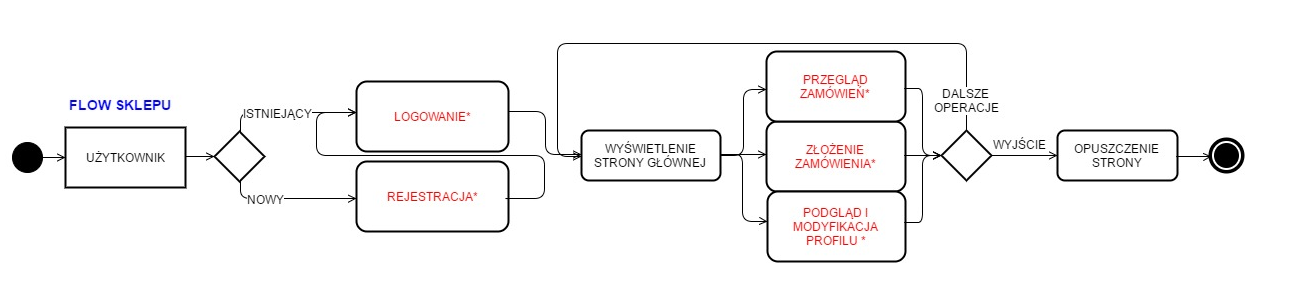
**Moderator** – Rola moderatora jest nadawana przez administratora. Otrzymuje możliwość dostępu do panelu administratora gdzie możliwe jest dodanie lub usunięcie przedmiotu. Poza operacjami na produktach może zatwierdzać lub odrzucać zamówienia.



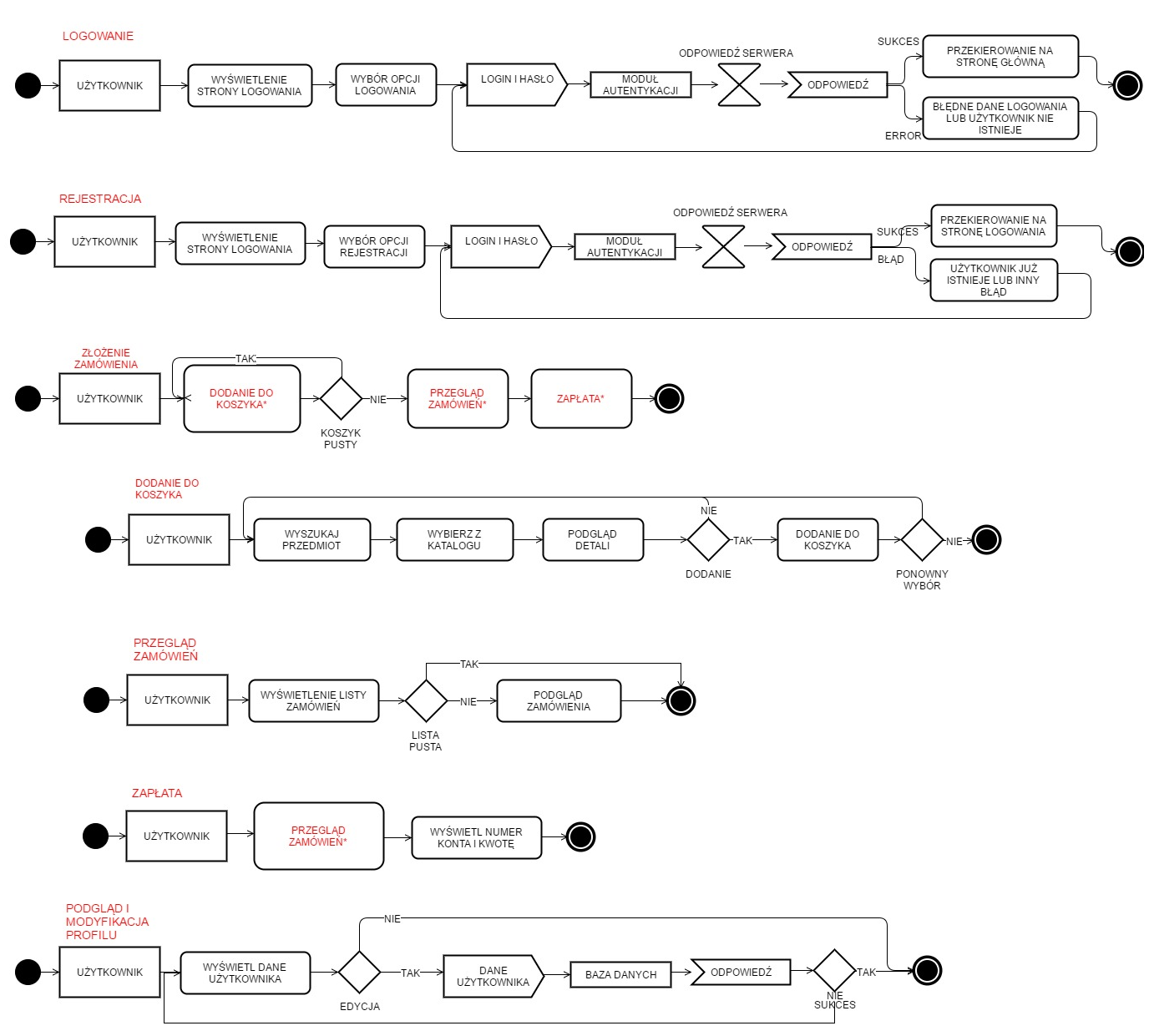
**Administrator** – najbardziej uszczegółowiona rola w systemie. Prócz wszystkich wyżej wymienionych operacji posiada możliwość zmiany uprawnień i usuwania użytkowników. Może również dezaktywować konto.



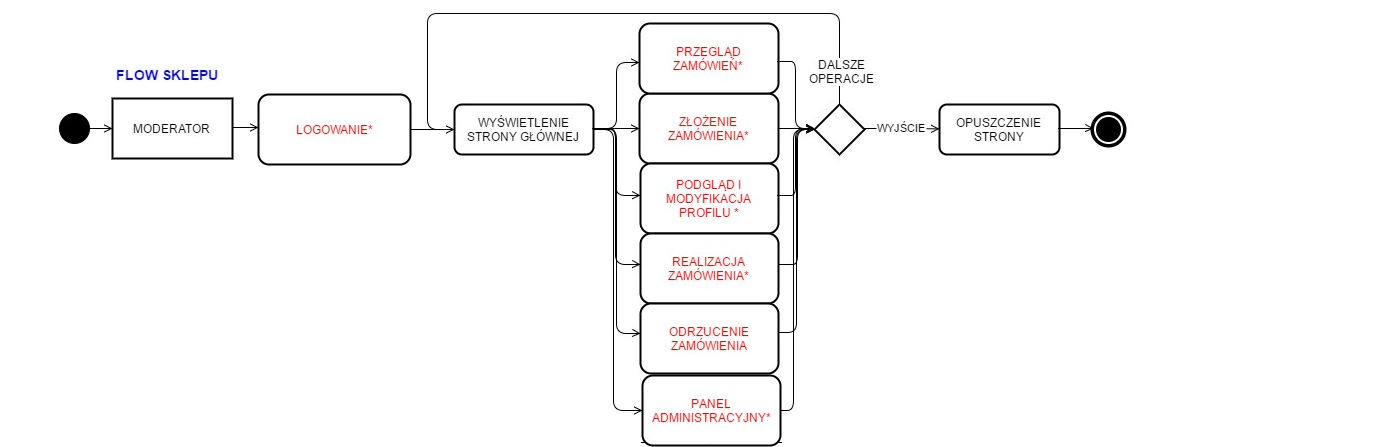
### 8 Diagram Czynności

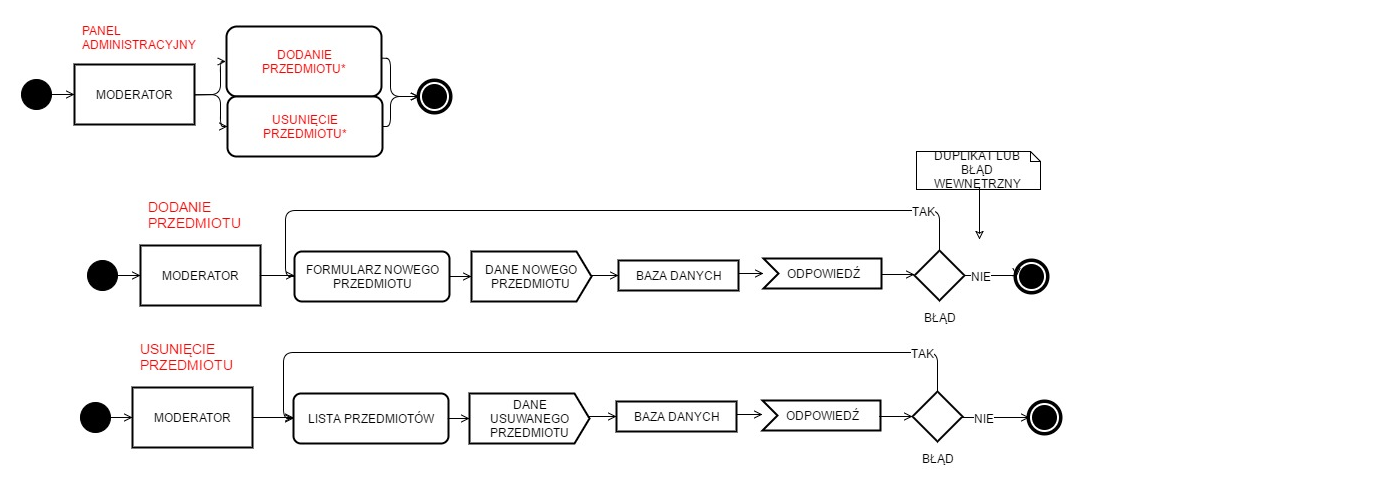
Diagram czynności zwany również diagramem aktywności jest używany w języku UML do opisu czynności i odpowiedzialności lub użytkowników systemu. Główne funkcjonalności systemu zostały przedstawione   
na poniższym diagramie, elementarne akcje oznaczone są kolorem czarnym zaś czynności czerwonym.

W projekcie zgodnie z przypadkami użycia diagramy sekwencji zostały podzielone pomiędzy role. Dla najbardziej uogólnionego użytkownika przedstawiają się następująco:

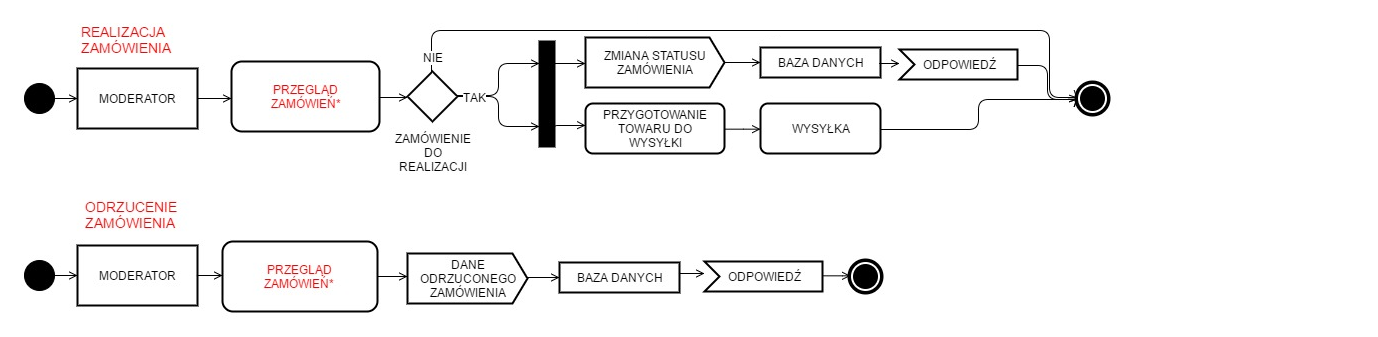


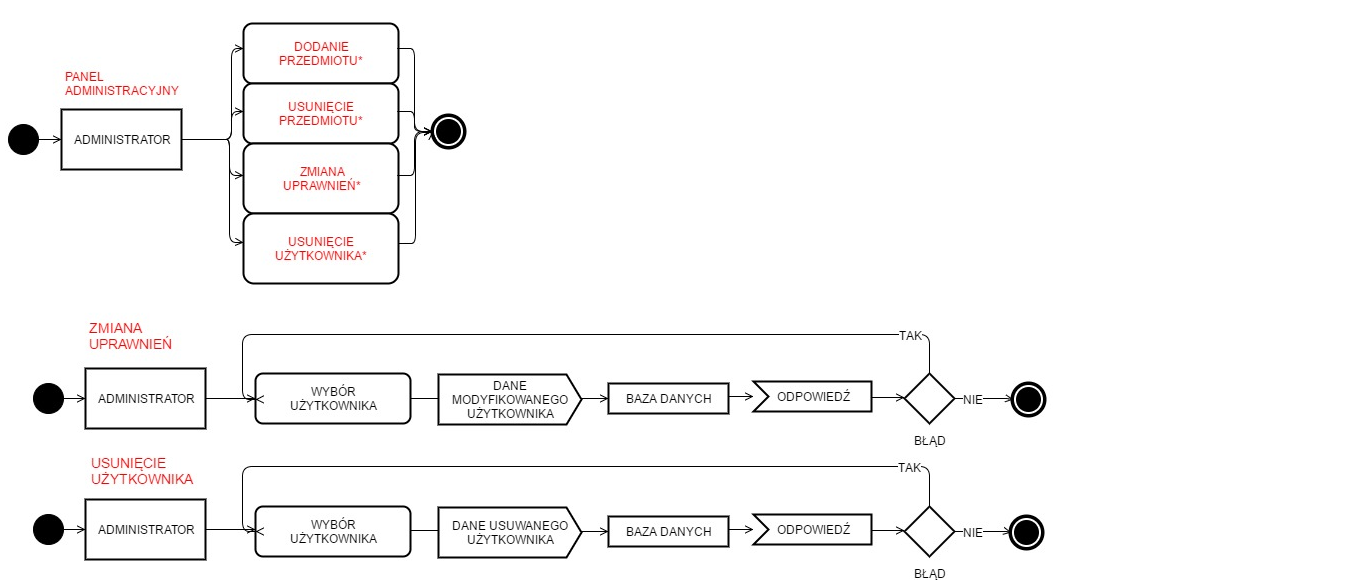
Dla moderatora i administratora działania w sklepie są rozszerzone o dodatkowe czynności i przedstawiają się następująco:



Panel administracyjny dla moderatora nie jest aż tak bardzo rozbudowany jak dla najbardziej uszczegółowionej roli w systemie:

Pozostałe możliwości moderatora:



  
Dla najbardziej uszczegółowionej roli w systemie dostępne są wszystkie wyżej wymienione operacje,  
różni się dwiema dodatkowymi funkcjonalnościami dostępnymi w panelu administracyjnym.

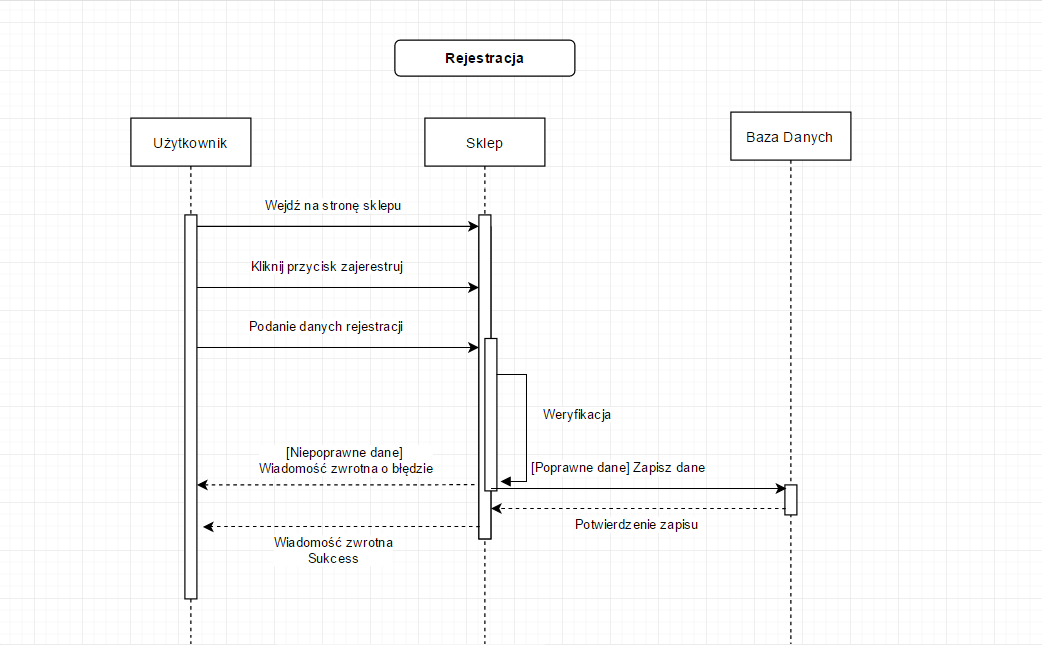
### 9 Diagram sekwencji

### Diagram sekwencji (ang. sequence diagram) służy do prezentowania interakcji pomiędzy obiektami wraz z uwzględnieniem w czasie komunikatów, jakie są przesyłane pomiędzy nimi.

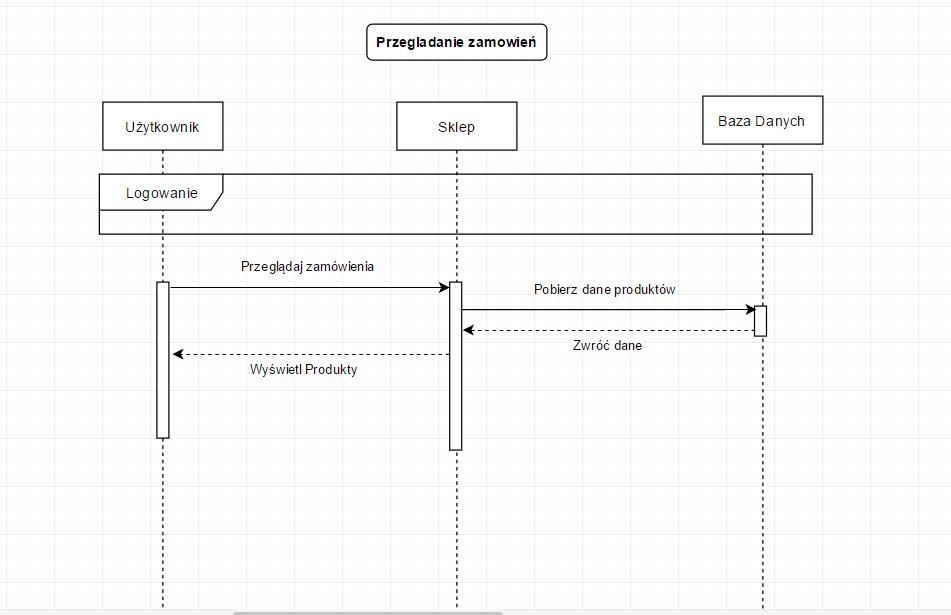
### Do projektu stworzono 6 diagramów sekwencji. 1) Diagram logowania 2) Diagram rejestracji 3) Diagram przeglądania zamówień 4) Diagram składania zamówienia 5) Diagram modyfikacji uprawnień 6) Diagram usuwania użytkownika Diagram logowania jest diagramem ref. (Ref – reference – wskazanie miejsca włączenia interakcji pokazanej na innym diagramie, fragment zawiera odwołanie do innego diagramu). Pozostałe diagramy są diagramami synchronicznymi. Podstawowymi obiektami występującymi w diagramie są: - Użytkownik (Klient, moderator lub admin) - Sklep - Baza danych

### C:\Users\Lolix\AppData\Local\Microsoft\Windows\INetCache\Content.Word\LOGOWANIE - SD.PNG

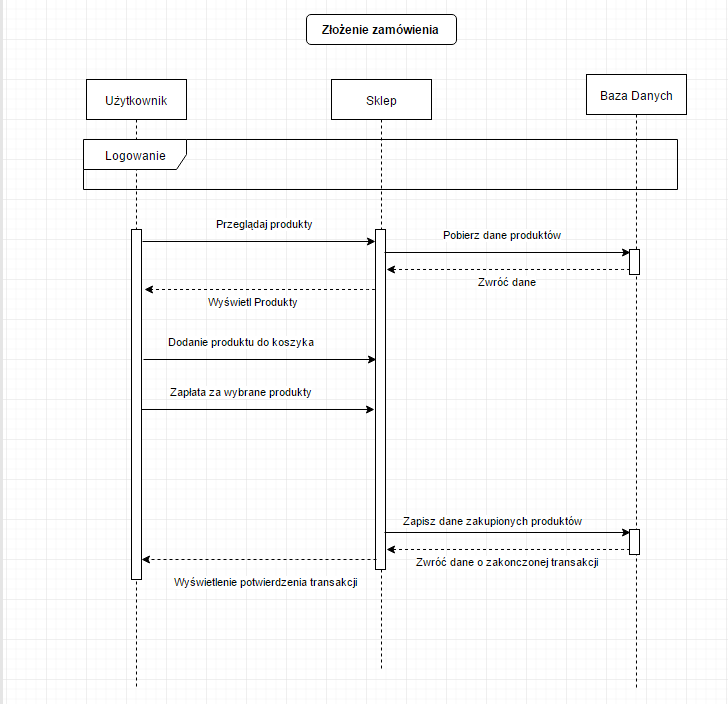
### *Rys. 9 Diagram logowania*



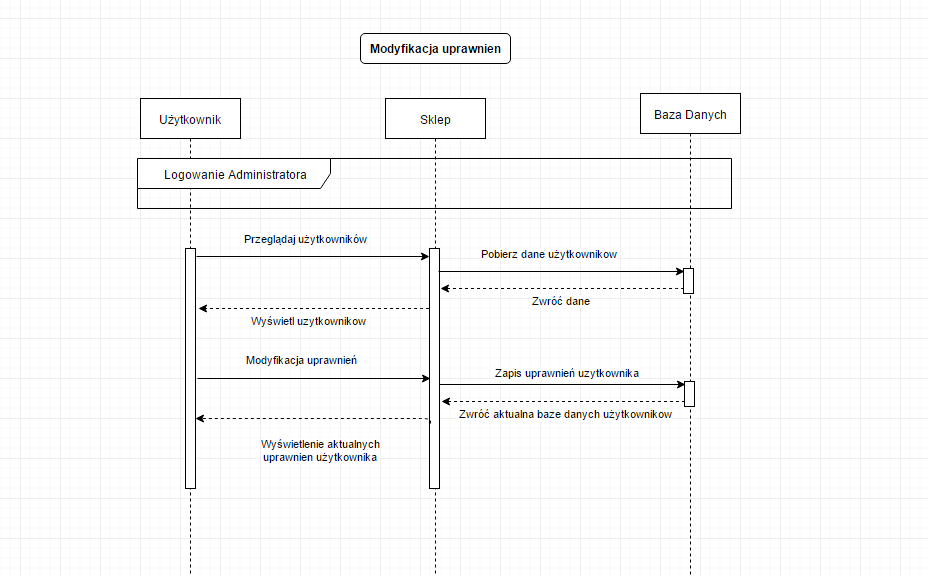
### *Rys. 10 Diagram rejestracji*



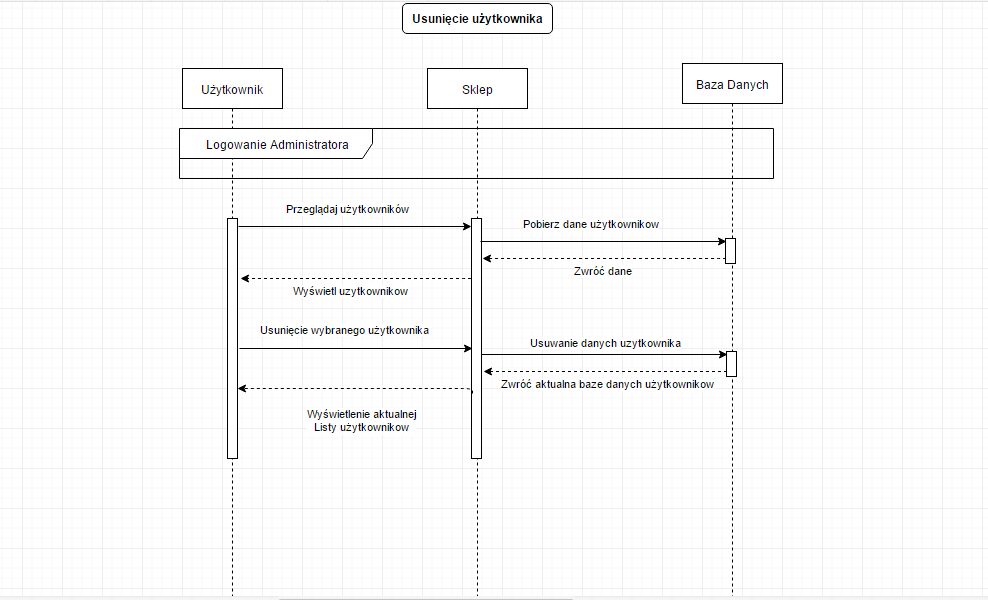
### *Rys. 11 Diagram przeglądania zamówienia*



### *Rys. 12 Diagram złożenia zamówienia*



### *Rys. 13 Diagram modyfikacji uprawnień*



### *Rys. 14 Diagram usuwania użytkownika*

### 

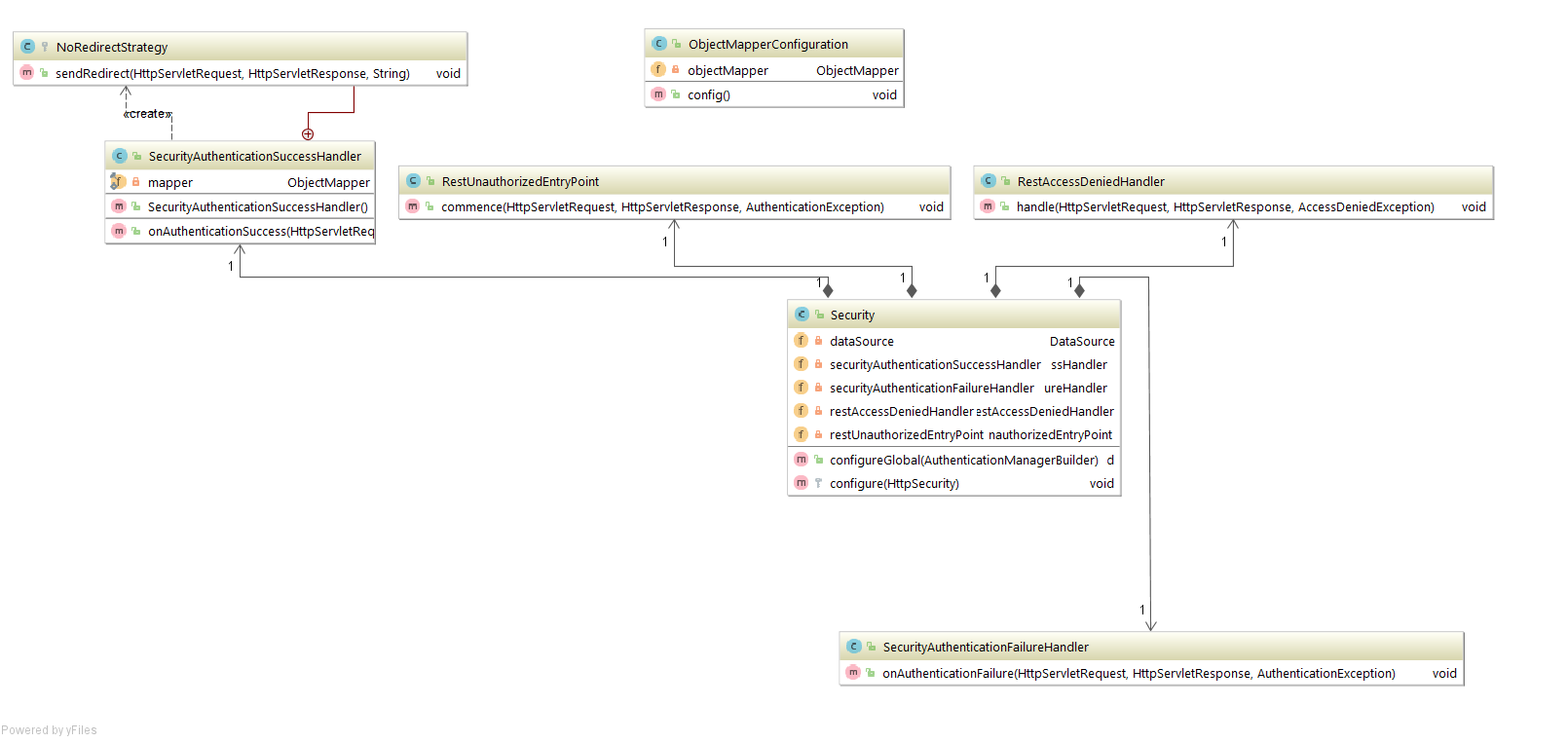
### 10 Diagram Klas

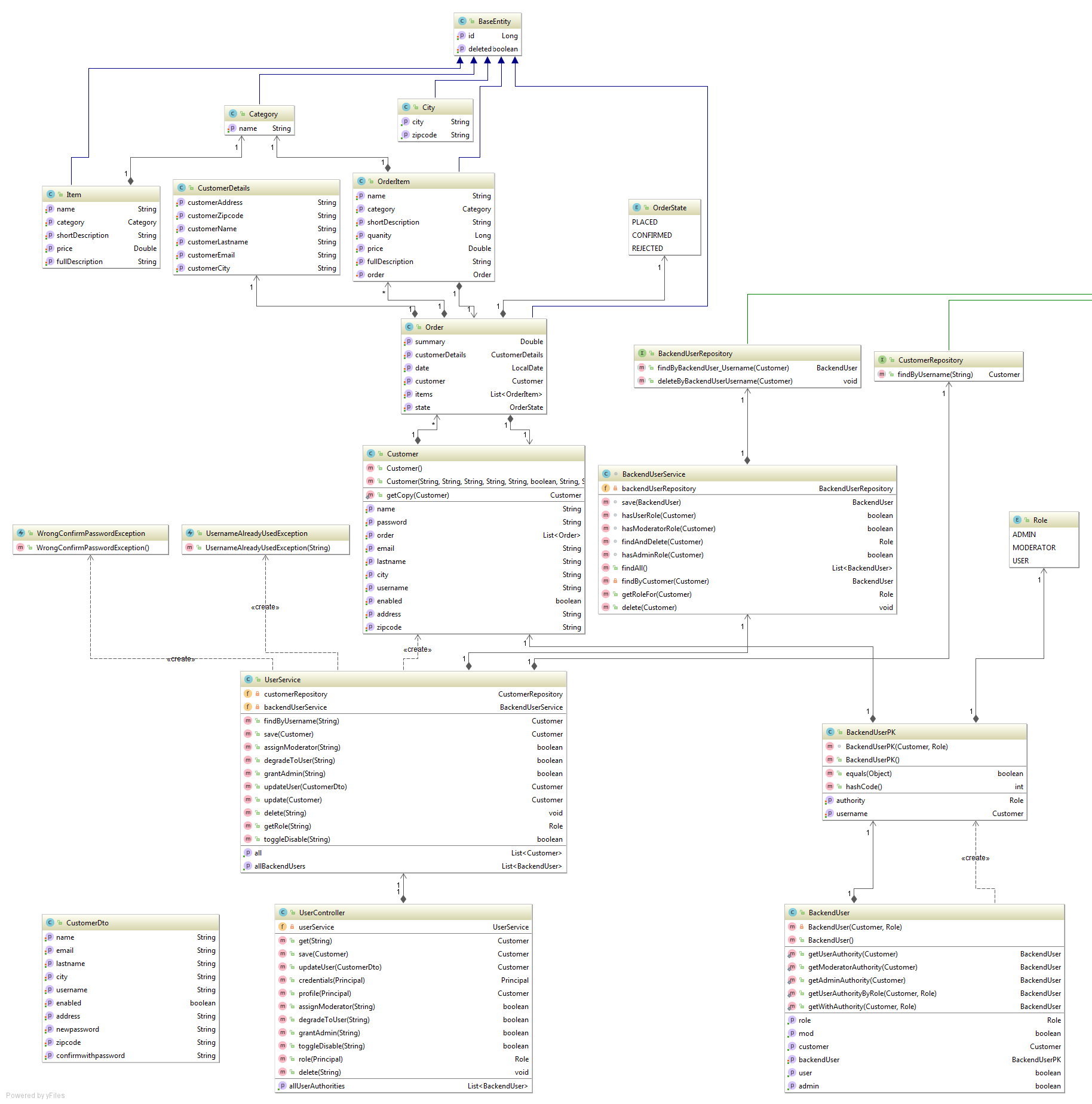
Poniżej znajdują się 3 diagramy klas. Podstawowe sposoby przedstawiania klas, różne poziomy szczegółowości trzy pola: nazwy, atrybutów, metod. Pierwszy z nich dotyczy zabezpieczeń.  
Drugi z nich zamawiania przedmiotów przez klienta. Trzeci natomiast modyfikacjami oraz administracja użytkowników. W diagramach występują relacje takie jak:

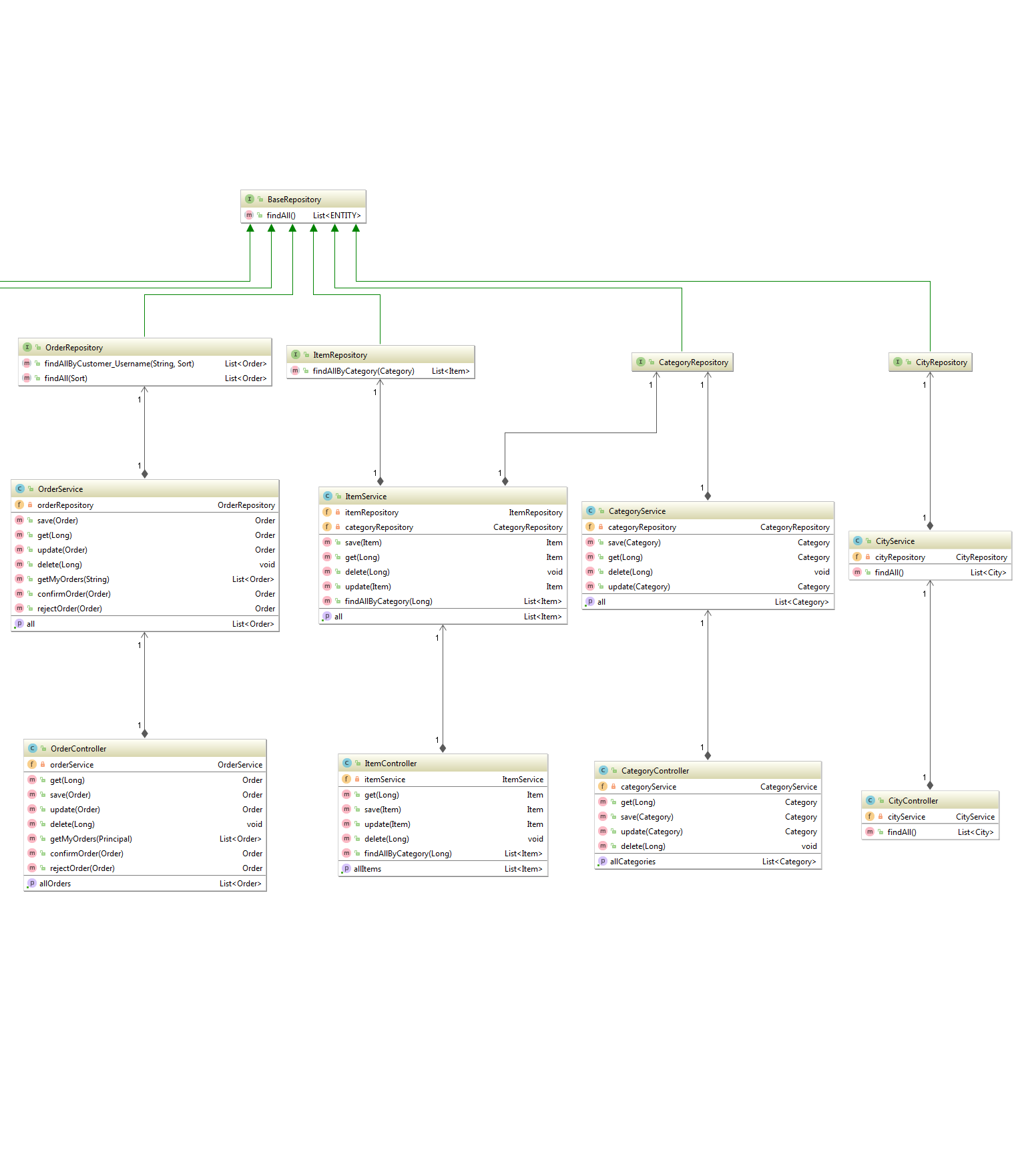
**- Agregacja** reprezentuje relację typu całość-część, w której część może należeć do kilku całości,  
 a całość nie zarządza czasem istnienia części.

**- Zależność** pomiędzy dwiema klasami informuje, że jedna z nich, aby używać obiektów innej,   
musi mieć o niej informacje. Zależność występuje, gdy zmiana specyfikacji jednej klasy, może powodować konieczność wprowadzania zmiany w innej klasie. Najczęściej używa się zależności do pokazania, że jedna klasa używa innej jako parametru jakiejś operacji.

**- Dziedziczenie** tworzy hierarchię klas, od ogólnych do bardziej szczegółowych. Pozwala wyłączyć części wspólne klas

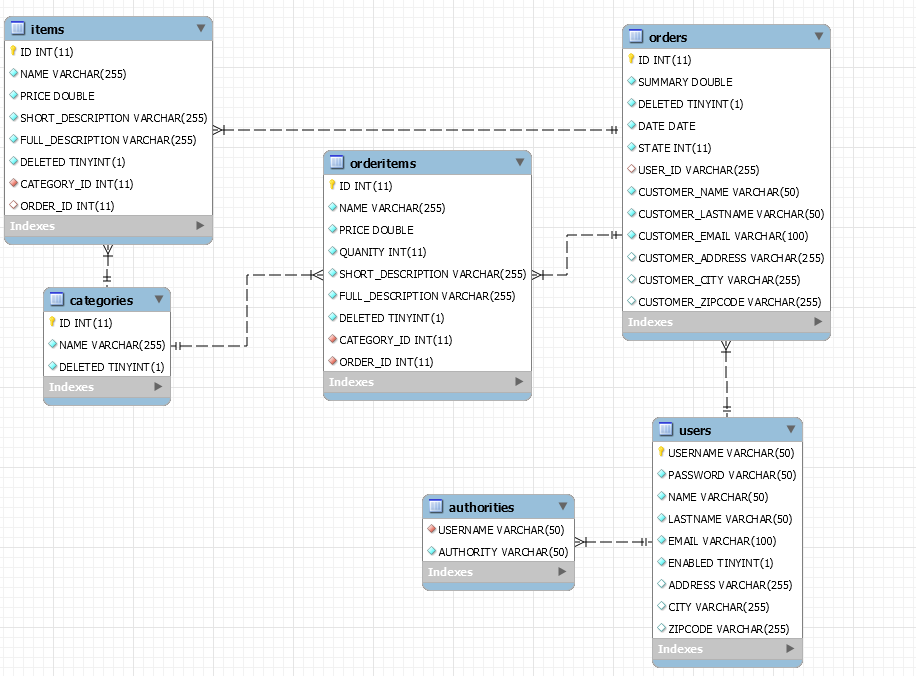






### 11 Diagram bazy danych

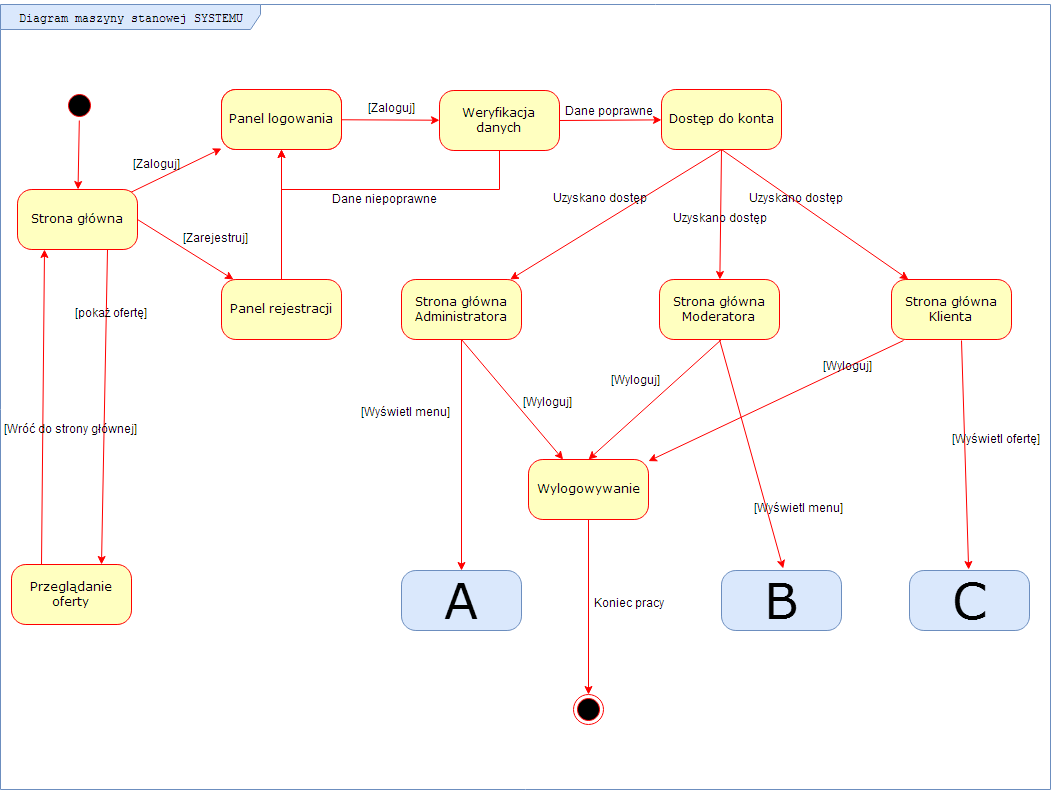
Fizyczny model ERD reprezentuje rzeczywisty projekt wzorcowej relacyjnej bazy danych. Przedstawia sposób w jaki dane powinny być powiązane w konkretnym DBMS, dlatego ważne jest, aby rozważyć konwencję i ograniczenie DBMS używane podczas projektowania fizycznego modelu ERD. Oznacza to, że konieczne jest dokładne wykorzystanie typu danych dla kolumn i użycie zastrzeżonych słów w nazwach i kolumnach. Poza tym w fizycznym modelu bazy danych powinny się znadjować klucze podstawowe, klucze obce i ograniczenia do projektu.

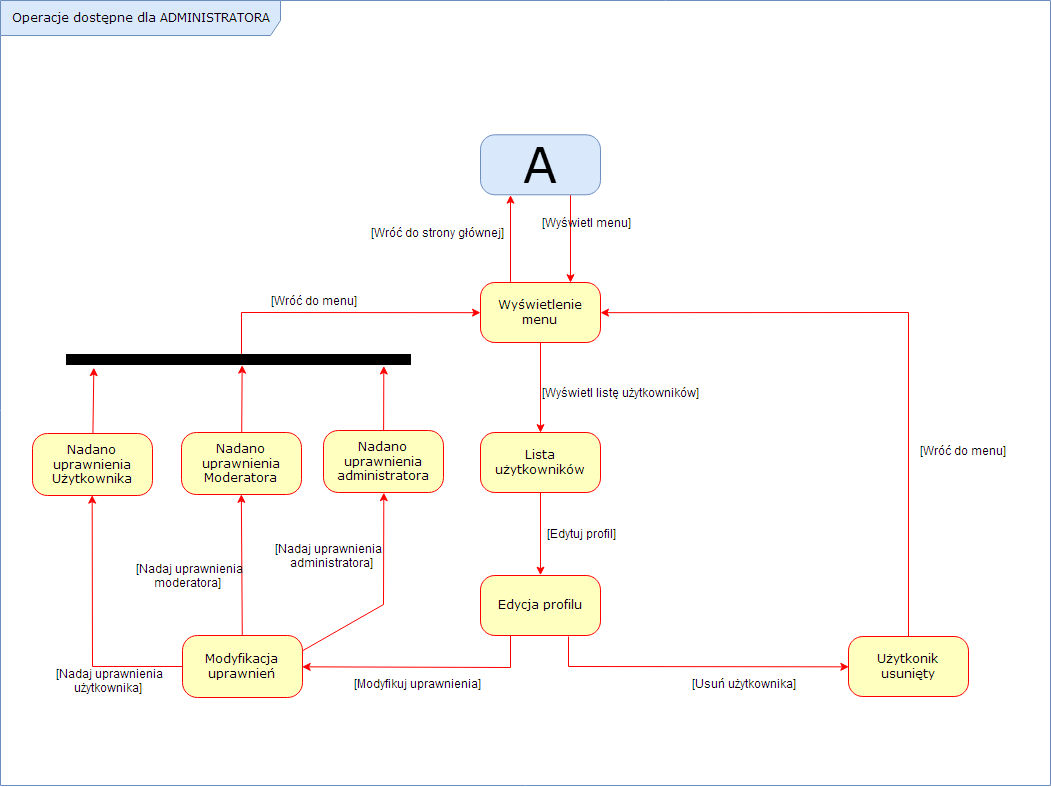


### 

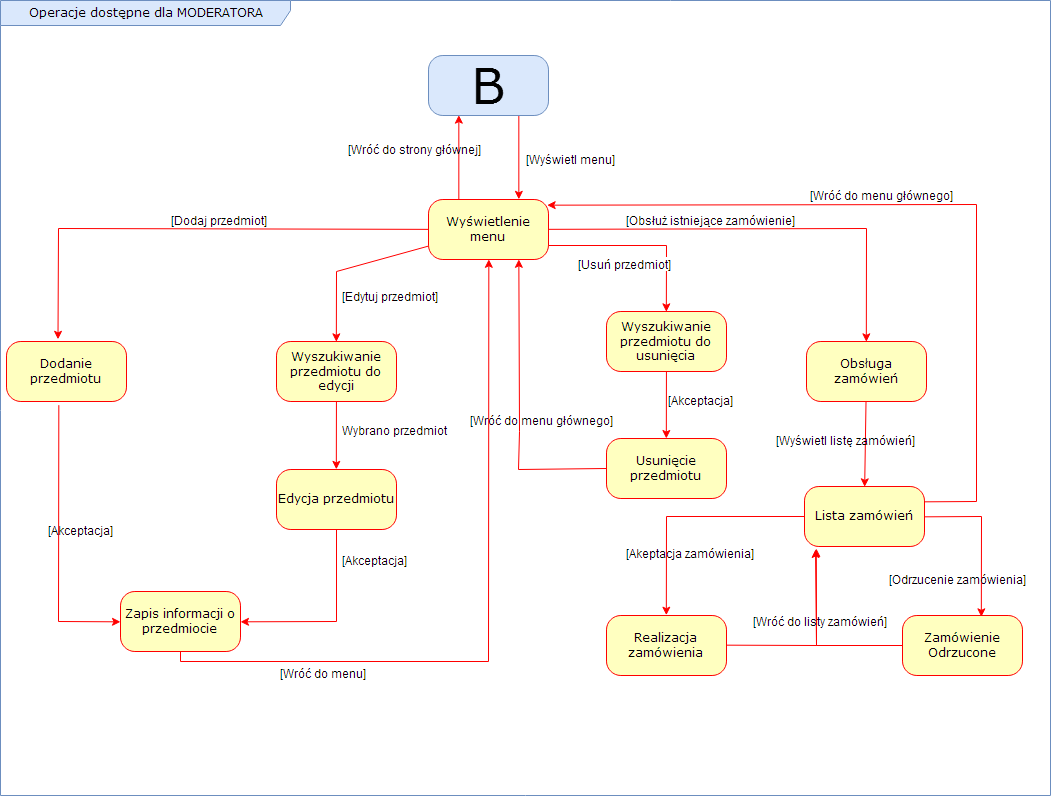
### 12 Diagram stanów

Schemat diagramu stanu jak sama nazwa opisuje różne stany obiektu systemu. Stany są specyficzne dla każdego obiektu. Schemat diagramu stanowego opisuje maszynę stanu. Maszyna stanu może być zdefiniowana jako maszyna, która definiuje różne stany obiektu, a stany te są kontrolowane przez zdarzenia zewnętrzne lub wewnętrzne.  
Diagram stanów jest pewną mutacją diagramu aktywności, z tą różnicą, że diagram aktywności skupia się raczej na opisaniu jakiegoś procesu, w którym uczestniczy wiele obiektów, zaś diagram stanów pokazuje, jakie są możliwe stany konkretnego obiektu.

1. Na pierwszym diagramie przedstawiono ogólny diagram stanów dla systemu sklepu internetowego:
2. Diagram drugi skupia się głównie na funkcjach dostępnych dla administratora systemu:



1. Trzeci diagram pokazuje główne funkcje moderatora systemu:



1. Na diagramie czwartym przedstawiono funkcjonalności dostępne dla zwykłego użytkownika – klienta sklepu internetowego:

