

# Projekt

*Modelowanie i analiza systemów  
informacyjnych*

# Analiza

## Wymagania

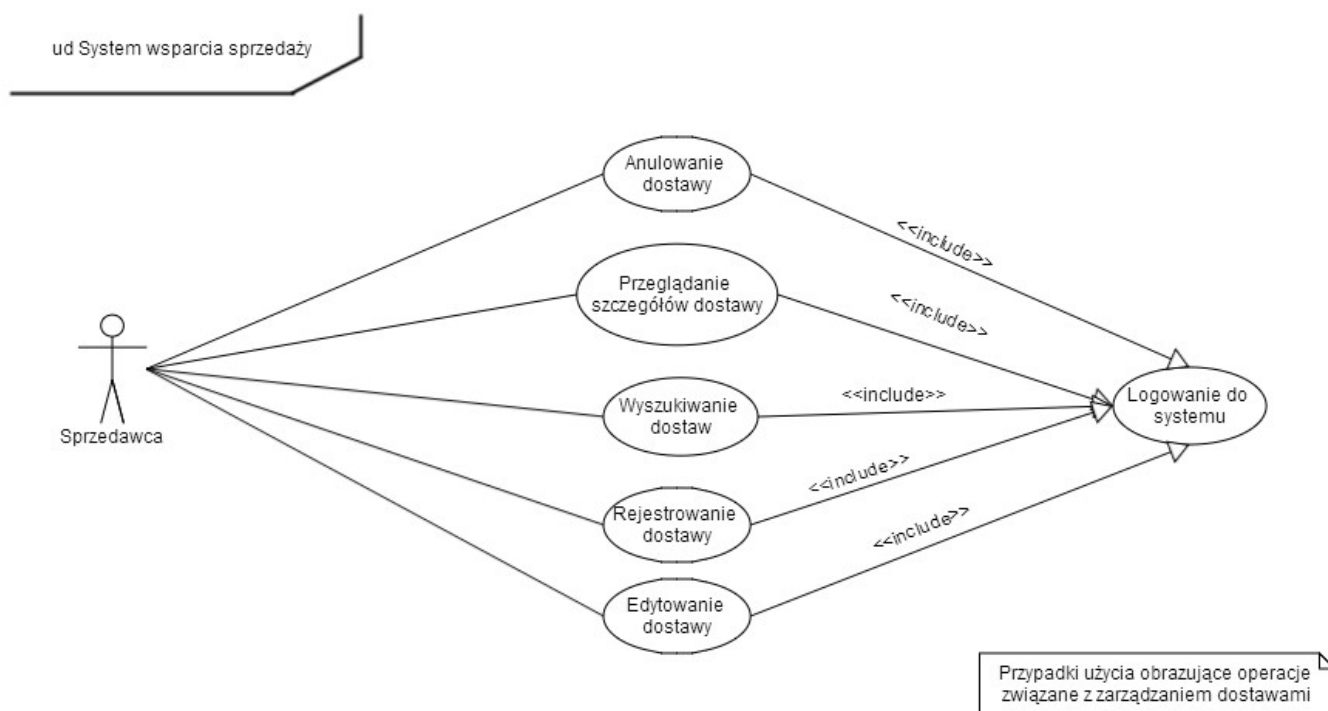
Właściciel sieci sklepów handlujących telefonami komórkowymi, poprosił firmę informatyczną o opracowanie systemu, który pozwoli rejestrować i śledzić towar znajdujący się w magazynach sklepów. System ma za zadanie wspierać rejestrowanie dostaw i sprzedaży produktów elektronicznych (telefonów komórkowych) poprzez umożliwienie przechowywania informacji na temat stanu magazynowego, zarządzaniem wieloma magazynami, identyfikację użytkowników systemu, przechowywanie informacji na temat dostaw, sprzedaży oraz odpowiednio kontrahentów i klientów firmy. Oprócz informacji o każdym urządzeniu (producent, model, kolor, marka, nr IMEI, cena) aktualnie znajdującym się w magazynach, system powinien przechowywać informacje na temat sprzedanych urządzeń, ich historii w systemie (czyli np. kto, kiedy dostarczył urządzenie, kto, kiedy i komu je sprzedał itd). System powinien uwzględniać możliwość składania skarg przez osoby kupujące w sklepie i dostarczające do niego towary. Dodatkowo system powinien zostać zaprojektowany w taki sposób, by w przyszłości była możliwa jego rozbudowa o dodatkowe funkcjonalności, takie jak: nowe poziomy uprawnień użytkowników, nowe moduły obsługujące rachunkowość, import/eksport danych z zewnętrznych źródeł.

## Aktor systemu

Sprzedawca

aktor reprezentujący sprzedawcę w sklepie, może wykonywać wszystkie operacje na produktach, dostawach i sprzedaży w obrębie sklepów do których jest przypisany.

## Diagram przypadków użycia



Zarządzanie dostawami

## Szczegółowy opis przypadków użycia

W celu pełnego odzwierciedlenia funkcjonalności systemu przyjęto następujący format opisu przypadków użycia, który składa się z :

**Nazwa:** <W postaci wyrażenia czasownikowego>

**Identyfikator:** <Symbol pozwalający jednoznacznie zidentyfikować przypadek użycia>

**Aktor główny:** <Nazwa głównego aktora, opis jego roli>

**Pozostali aktorzy i udziałowcy:** <Nazwy aktorów, ich interesy>

**Warunki początkowe:** <Co system zapewnia przed zezwoleniem na rozpoczęcie przypadku użycia?>

**Opis przebiegu interakcji:** <Poszczególne kroki wykonania przypadku użycia>

**Przebieg alternatywny n:** <Poszczególne kroki alternatywnego wykonania przypadku użycia>

**Warunki końcowe:** <Warunki spełnione po pomyślnym wykonaniu głównego scenariusza przypadku użycia>

**Powiązane:** <Referencje do powiązanych przypadków użycia lub/i wymagań funkcjonalnych>

**Uwagi:** <Informacje dodatkowe niepasujące do ww. punktów>

### 1. Logowanie do systemu

**Identyfikator :** PU1

**Aktor główny:** Sprzedawca

**Pozostali aktorzy i udziałowcy:** Brak

**Warunki początkowe:** Sprzedawca nie posiada zainicjowanej sesji w systemie lub też jego poprzednia sesja wygasła z powodu braku aktywności w systemie

**Opis przebiegu interakcji:** Sprzedawca wprowadza dane : adres email, hasło. Kliknięcie przycisku *zaloguj*. Po weryfikacji danych dla każdego typu aktora załadowanie odpowiednich praw użytkownika w systemie. Odnotowanie w logach systemu akcji wykonanej przez użytkownika. Przekierowanie użytkownika na stronę główną (dashboard).

**Przebieg alternatywny 1:** Sprzedawca wprowadza niepoprawne dane : adres email lub/i hasło. Kliknięcie przycisku *zaloguj*. Po weryfikacji, wyświetlany jest komunikat informujący o tym iż dane są niepoprawne.

**Przebieg alternatywny 2:** Sprzedawca zalogował się uprzednio do systemu, jednakże przez dłuższy czas nie wykonał w nim żadnej akcji

**Warunki końcowe:** Utworzenie ważnej sesji sprzedawcy w systemie. Załadowanie praw sprzedawcy w systemie. Przekierowanie na stronę główną (dashboard)

**Powiązane:** brak

**Uwagi:** brak

### 2. Rejestrowanie dostaw

**Identyfikator:** PU2

**Aktor główny:** Sprzedawca

**Pozostali aktorzy i udziałowcy:** Brak

**Warunki początkowe:** Sprzedawca posiada uprawnienia do rejestrowania nowych dostaw w sklepie

**Opis przypadku interakcji:** Sprzedawca klika przycisk *dodaj nową dostawę*. Wyświetlanie formularza, który pozwala wprowadzić informacje o dostawie – dacie, czasie, miejscu dostawy,

kontrahencie i liście produktów, które dostarczono – każdy produkt to numer IMEI, producent, model, cena dostawy, stawka opodatkowania . Po wprowadzeniu poprawnych danych i potwierdzeniu wysyłki danych następuje odnotowanie w logach systemu akcji wykonanej przez użytkownika i potwierdzenie utrwalenia zmian stosownym komunikatem. Sprzedawca zostaje przekierowany na stronę wyszukiwania produktów.

**Przebieg alternatywny 1:** Sprzedawca klika przycisk *dodaj nową dostawę*. Wyświetlanie formularza, który pozwala wprowadzić informacje o dostawie. Po wprowadzeniu niepoprawnych danych i potwierdzeniu wysyłki danych następuje wyświetlenie stosownego komunikatu i wskazaniu przez system danych, które są niepoprawne.

**Warunki końcowe:** Utrwalenie informacji o dostawie w systemie. Odnotowanie w logach systemu informacji o akcji wykonanej przez sprzedawcę.

**Powiązane:** brak

**Uwagi:** brak

### 3. Modyfikowanie dostaw

**Identyfikator:** PU3

**Aktor główny:** Sprzedawca

**Pozostali aktorzy i udziałowcy:** Brak

**Warunki początkowe:** Sprzedawca posiada uprawnienia do edytowania dostaw.

**Opis przypadku użycia:** Z listy wszystkich dostaw (PU5) sprzedawca klika przycisk *edytuj* przy dostawie, którą chce zmodyfikować. Otwarty zostaje formularz (taki sam przy dodawaniu dostawy) przy czym z uzupełnionymi danymi. Sprzedawca modyfikuje dane formularza. Potwierdza modyfikacje klikając *modyfikuj*. Po weryfikacji poprawności danych następuje odnotowanie w logach systemu akcji wykonanej przez użytkownika. Sprzedawca zostaje przekierowany na stronę wyszukiwania i zostaje poinformowany o utrwaleniu zmian stosownym komunikatem.

**Przebieg alternatywny 1:** Z listy wszystkich dostaw (PU5) sprzedawca klika przycisk *edytuj* przy dostawie, którą chce zmodyfikować. Otwarty zostaje formularz (taki sam przy dodawaniu dostawy) przy czym z uzupełnionymi danymi. Sprzedawca modyfikuje dane formularza. Potwierdza modyfikacje klikając *modyfikuj*. Weryfikacja poprawności danych nie powodzi się. Sprzedawca zostaje poinformowany o tym stosownym komunikatem.

**Warunki końcowe:** Utrwalenie zmian w systemie. Odnotowanie w logach systemu informacji o akcji wykonanej przez sprzedawcę.

**Powiązane:** PU5

**Uwagi:** brak

### 4. Anulowanie dostaw

**Identyfikator:** PU4

**Aktor główny:** Sprzedawca

**Pozostali aktorzy i udziałowcy:** Brak

**Warunki początkowe:** Uprawnienia do anulowania dostaw.

**Opis przypadku użycia:** Z listy wszystkich dostaw (PU5) sprzedawca klika przycisk *anuluj* przy dostawie, którą chce anulować. Po weryfikacji czy wszystkie produktu zawierające się w dostawie są niesprzedane następuje usunięcie informacji o dostawie i jej produktach z systemu.

Odnotowanie w logach systemu akcji wykonanej przez sprzedawcę. Przekierowanie sprzedawcę na listę dostaw.

**Przebieg alternatywny 1:** Z listy wszystkich dostaw (PU5) sprzedawca klika przycisk *anuluj* przy dostawie, którą chce anulować. Po weryfikacji okazuje się, że co najmniej jeden produkt z dostawy został już sprzedany. Wtedy sprzedawca zostaje poinformowany o tym odpowiednim komunikatem. Anulowanie dostawy nie powiodło się.

**Warunki końcowe:** Usunięcie dostawy i jej produktów z systemu. Odnotowanie informacji o wykonanej akcji w systemie.

**Powiązane:** brak

**Uwagi:** brak

## 5. Wyszukiwanie dostaw

**Identyfikator:** PU5

**Aktor główny:** Sprzedawca

**Pozostali aktorzy i udziałowcy:** Brak

**Warunki początkowe:** Uprawnienia do wyświetlania dostaw.

**Opis przypadku użycia:** Wyświetlenie listy dostaw według aktualnie wybranych kryteriów wyszukiwania. Kryteriami wyszukiwania mogą być : sklep, data dodawania, data sprzedaży, model, producent, IMEI, ilość dostaw na stronie. Po wprowadzeniu wszystkich kryteriów sprzedawca klika przycisk *filtruj*. Wtedy tabela zawierająca listę dostaw zostaje odświeżona o wyniki spełniające zadane kryteria. Tabela jest stronicowana.

**Warunki końcowe:** Wyświetlenie listy dostaw w sklepie. Odnotowanie informacji o wykonanej akcji w systemie.

**Powiązane :** brak

**Uwagi :** brak

## 6. Przeglądanie szczegółów dostawy

**Identyfikator:** PU6

**Aktor główny:** Sprzedawca

**Pozostali aktorzy i udziałowcy:** Brak

**Warunki początkowe:** Uprawnienia do szczegółów dostawy

**Opis przypadku użycia:** Z listy wszystkich dostaw (PU5) sprzedawca klika przycisk *szczegóły* przy dostawie, którą chce przejrzeć. Wtedy zostaje przekierowany do strony zawierającej szczegółowe informacji nt. dostawy i jej produktów. Odnotowanie w logach systemu akcji wykonanej przez użytkownika.

**Warunki końcowe:** Wyświetlenie produktów dostawy. Odnotowanie w logach systemu akcji wykonanej przez użytkownika.

**Powiązane :** UP5

**Uwagi :** brak

# Projekt interfejsu

Login

Email

\*\*\*\*\*

Login

Strona logowania do systemu

Browse products

Home > Products > Browser

Products

Deliveries

Sales

Users

Complaints

Imei

Price from

Producer

Price to

Model

Tax from

Color

Tax to

Store

Filter

IMEI	Price	Tax rate	Producer	Model	Color	Store	Action
345792367	100.00	19%	Nokia	3310	black	Gliwice	<a href="#">Edytuj</a> <a href="#">Usuń</a>
985112242	100.00	19%	Nokia	3310	black	Warsaw	<a href="#">Edytuj</a> <a href="#">Usuń</a>
823421132	100.00	19%	Nokia	3310	black	Rybnik	<a href="#">Edytuj</a> <a href="#">Usuń</a>
234834112	100.00	19%	Nokia	3310	black	Gliwice	<a href="#">Edytuj</a> <a href="#">Usuń</a>
345792367	100.00	19%	Nokia	3310	black	Gliwice	<a href="#">Edytuj</a> <a href="#">Usuń</a>
985112242	100.00	19%	Nokia	3310	black	Warsaw	<a href="#">Edytuj</a> <a href="#">Usuń</a>
345792367	100.00	19%	Nokia	3310	black	Gliwice	<a href="#">Edytuj</a> <a href="#">Usuń</a>
985112242	100.00	19%	Nokia	3310	black	Warsaw	<a href="#">Edytuj</a> <a href="#">Usuń</a>

Strona wyszukiwania produktów sieci

New Delivery

Home > Delivery > New Delivery

[Products](#)  
[Deliveries](#)  
[Sales](#)  
[Users](#)  
[Complaints](#)

Label

Delivery date

Delivered by

Registered by

Model	Producer	IMEI	Color	Price(\$)	Tax (%)
3310	Nokia	23456723723	black	21,00	19
3310	Nokia	121567342210	white	21,00	19
3310	Nokia	823914012002	red	21,00	15
3310	Nokia	638210310203	black	21,00	19

Model

Producer

IMEI

Color

Price

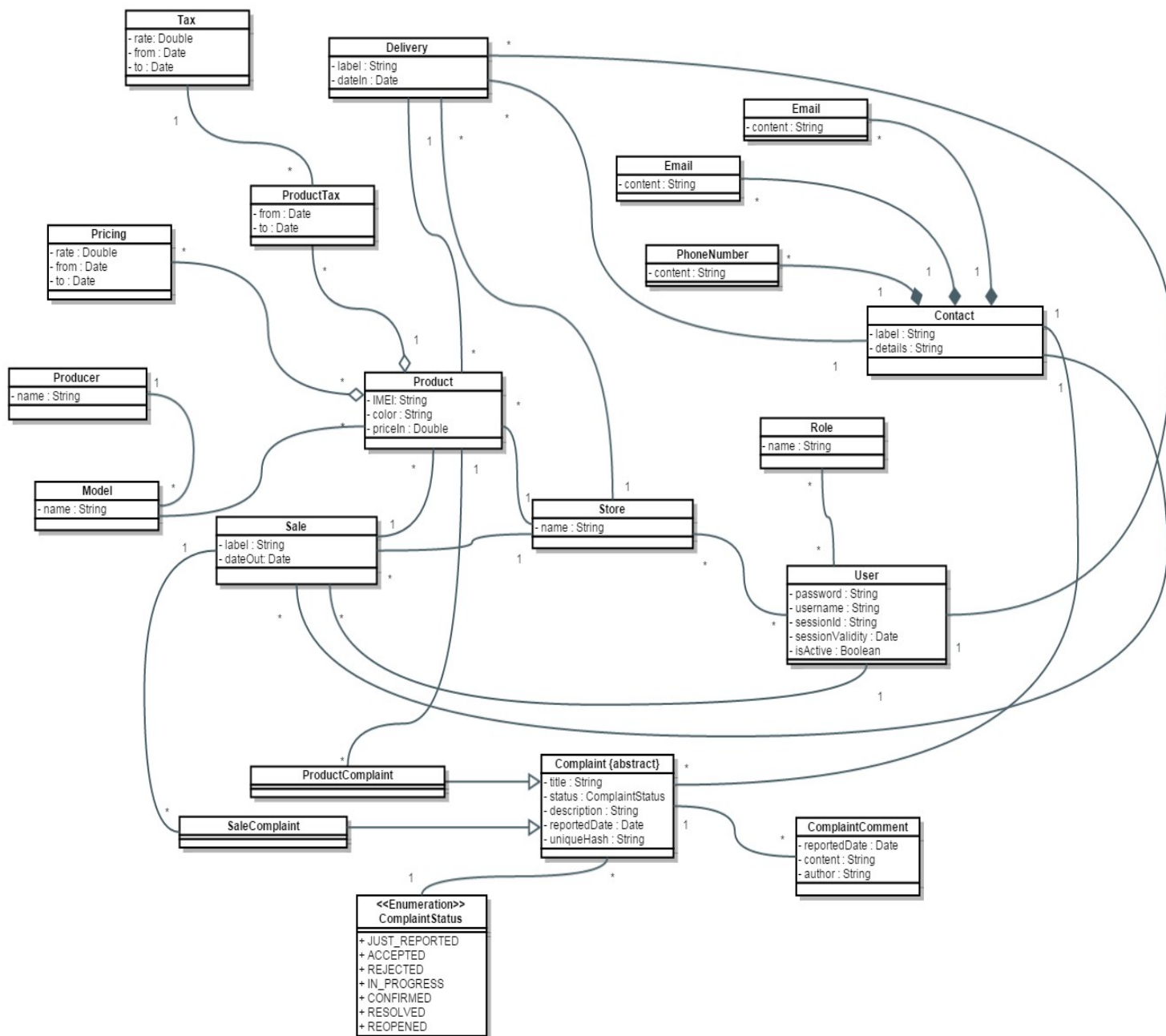
Tax

Add product

Confirm delivery

Strona dodawania nowej dostawy

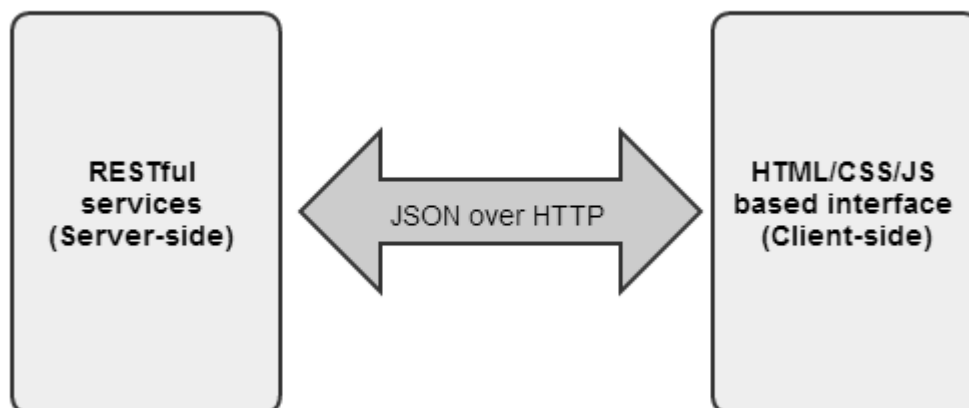
## Diagram klas analityczny



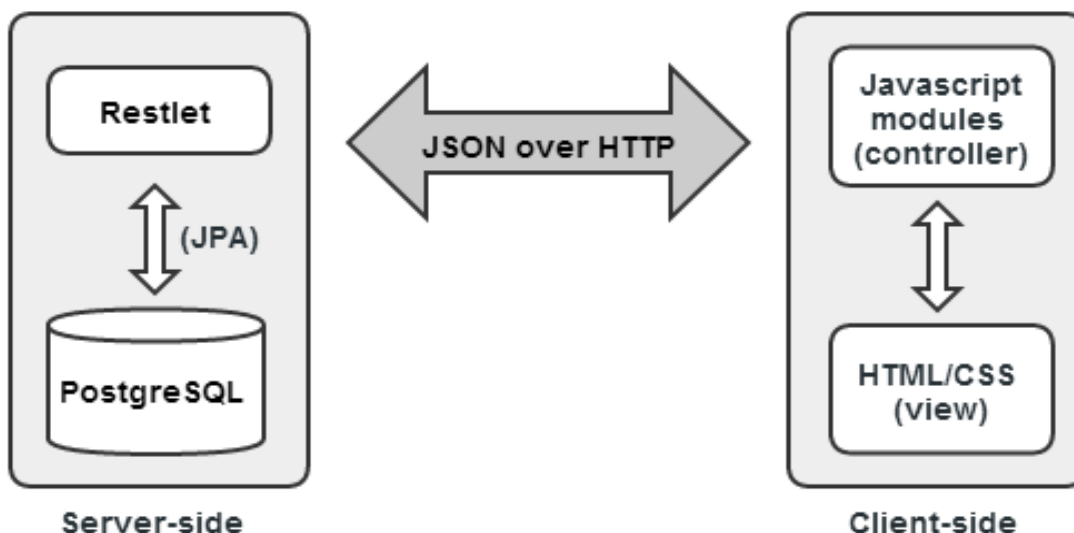


## Decyzje projektowe

System jest zrealizowany w architekturze klient-serwer. Część serwerowa implementuje usługi systemu według standardu [RESTful](#). Natomiast część kliencka zrealizowana zostanie w standardowych technologiach webowych – HTML, Javascript, CSS. Obie technologie/standardy gwarantują długoterminową i stabilną pracę nad systemem w przyszłości.

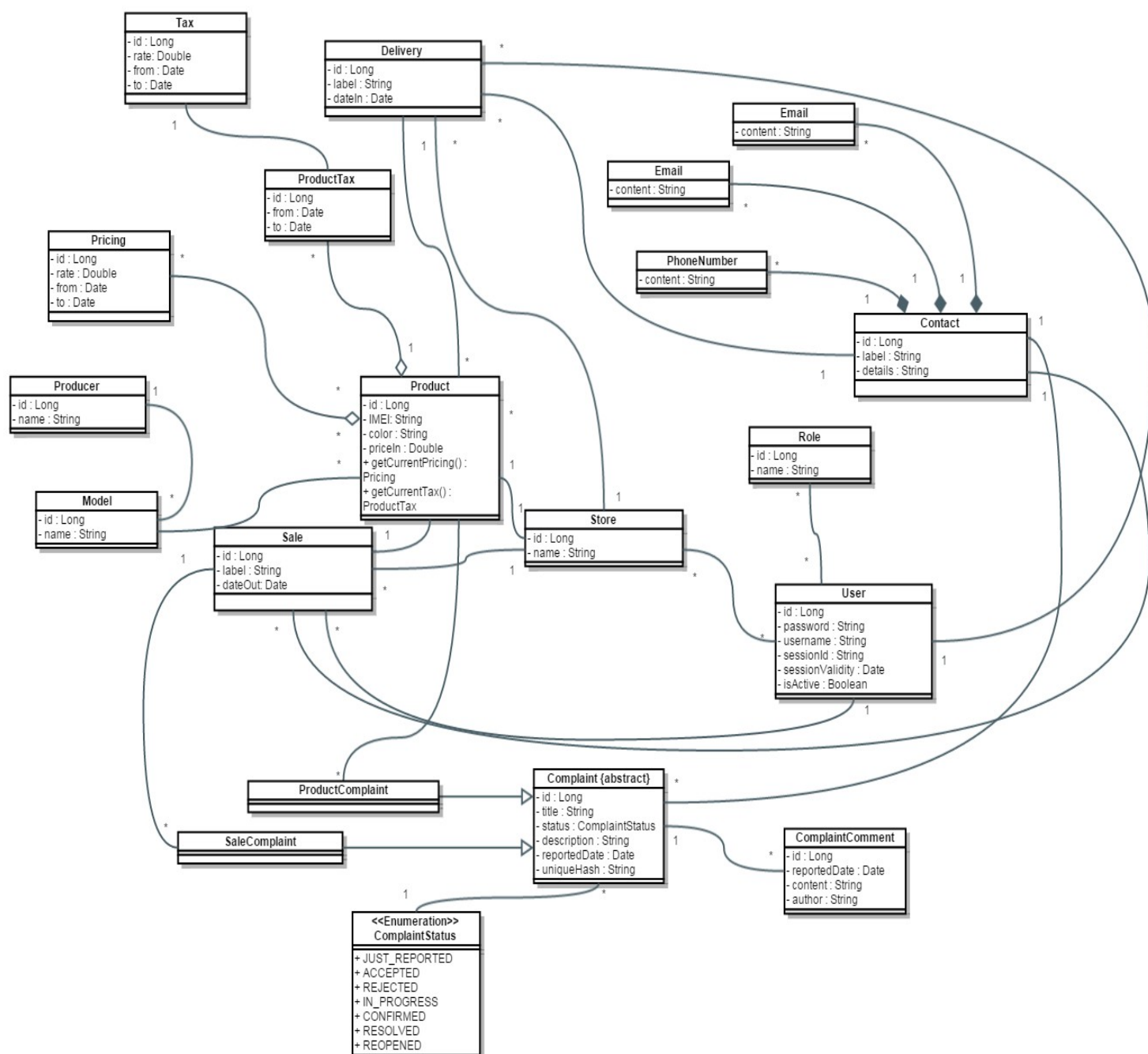


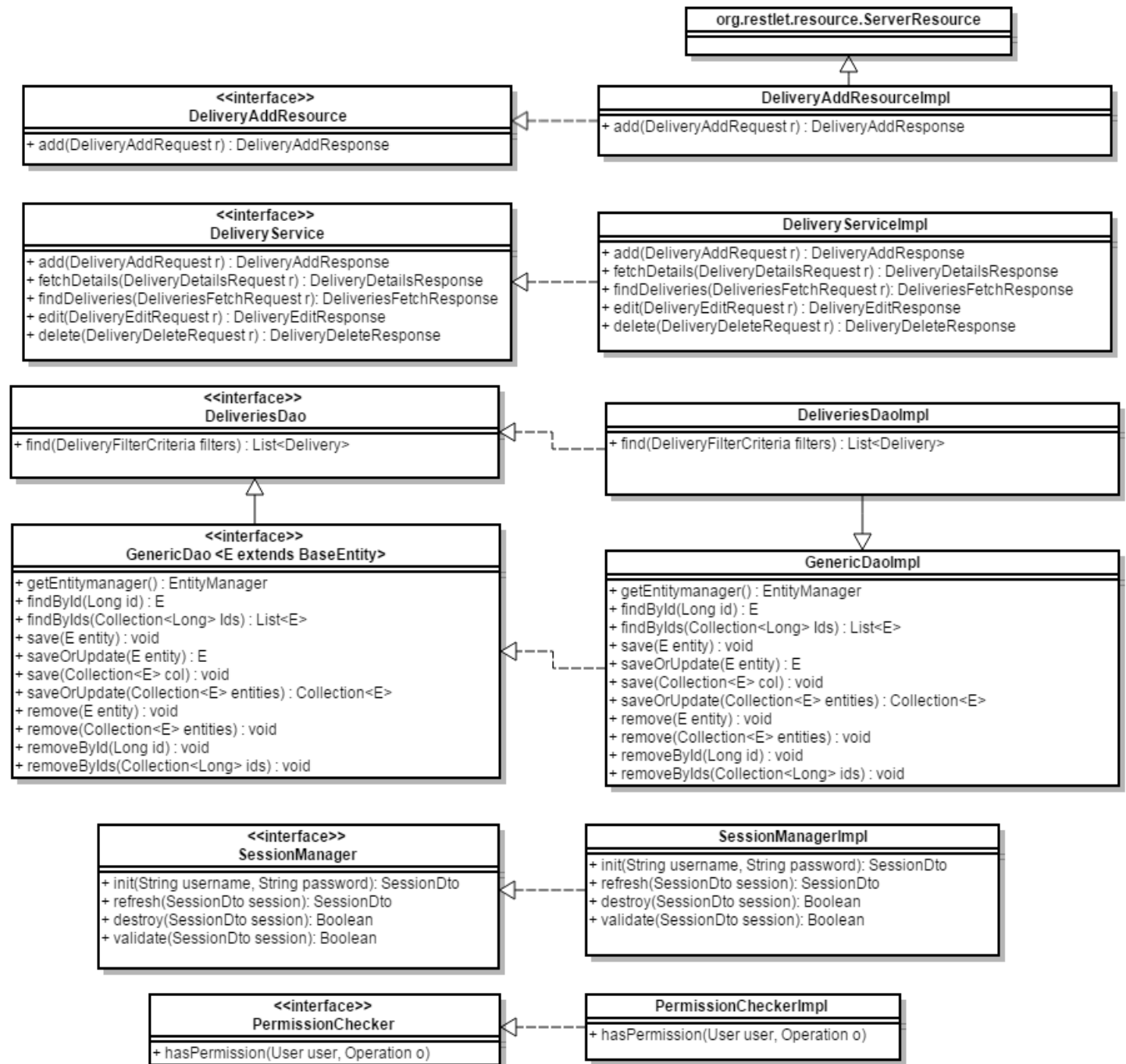
Zagłębując pod powierzchnię systemu należałoby zwrócić uwagę na główne biblioteki, które w dużym stopniu ułatwiają i przyspieszają prace nad taką architekturą systemu. Do realizacji [web service](#)ów w standardzie REST wykorzystano pakiet [Restlet](#), popularny, dobrze udokumentowany i z szeroką gamą rozszerzeń. Po stronie klienckiej zaś najważniejsze komponenty graficzne zrealizowane są w oparciu o front-endowy framework [bootstrap](#). Komunikacja z serwerem odbywa się z wykorzystaniem technologii [AJAX](#), gdzie kluczową rolę odgrywają moduły napisane w języku javascript.

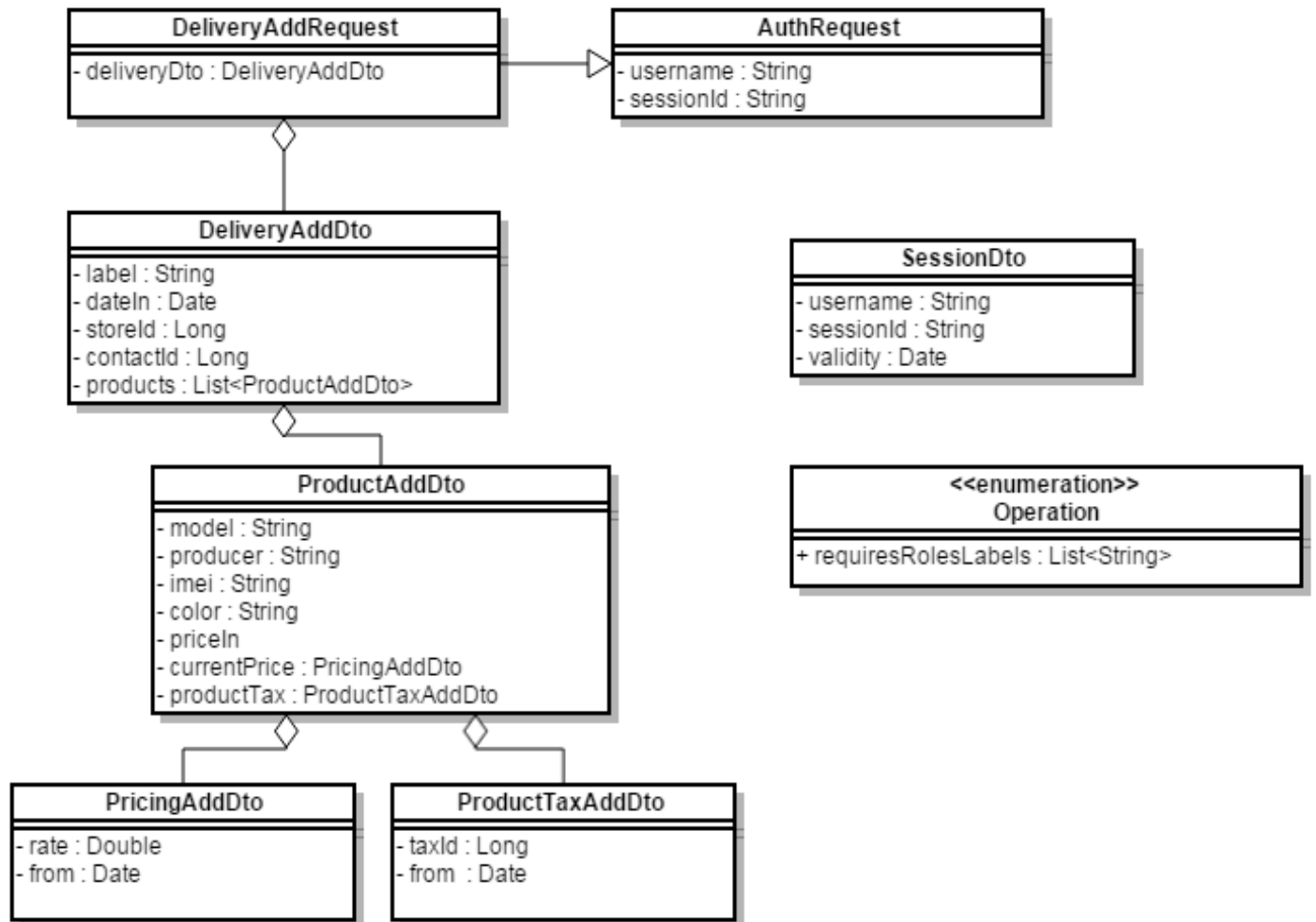


Po stronie serwera należy również zwrócić uwagę na warstwę bazy danych, która jest zarządzana poprzez framework [JPA 2](#), który jest wiodącym standardem realizującym operacje dostępu do baz danych na platformie [JEE](#).

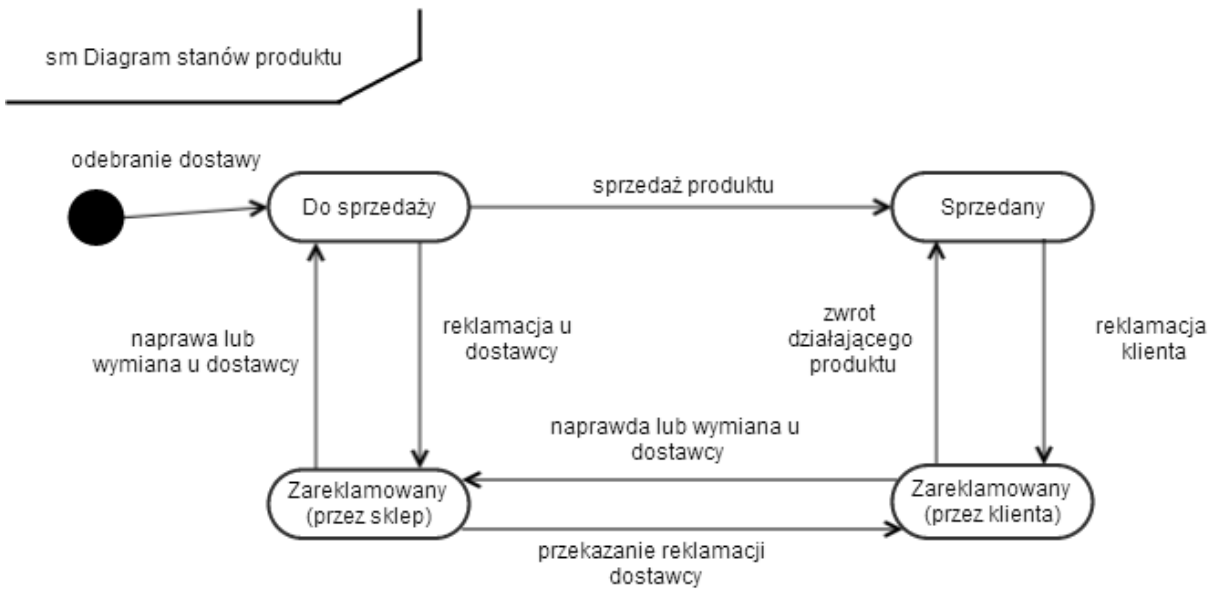
## Diagram klas projektowy



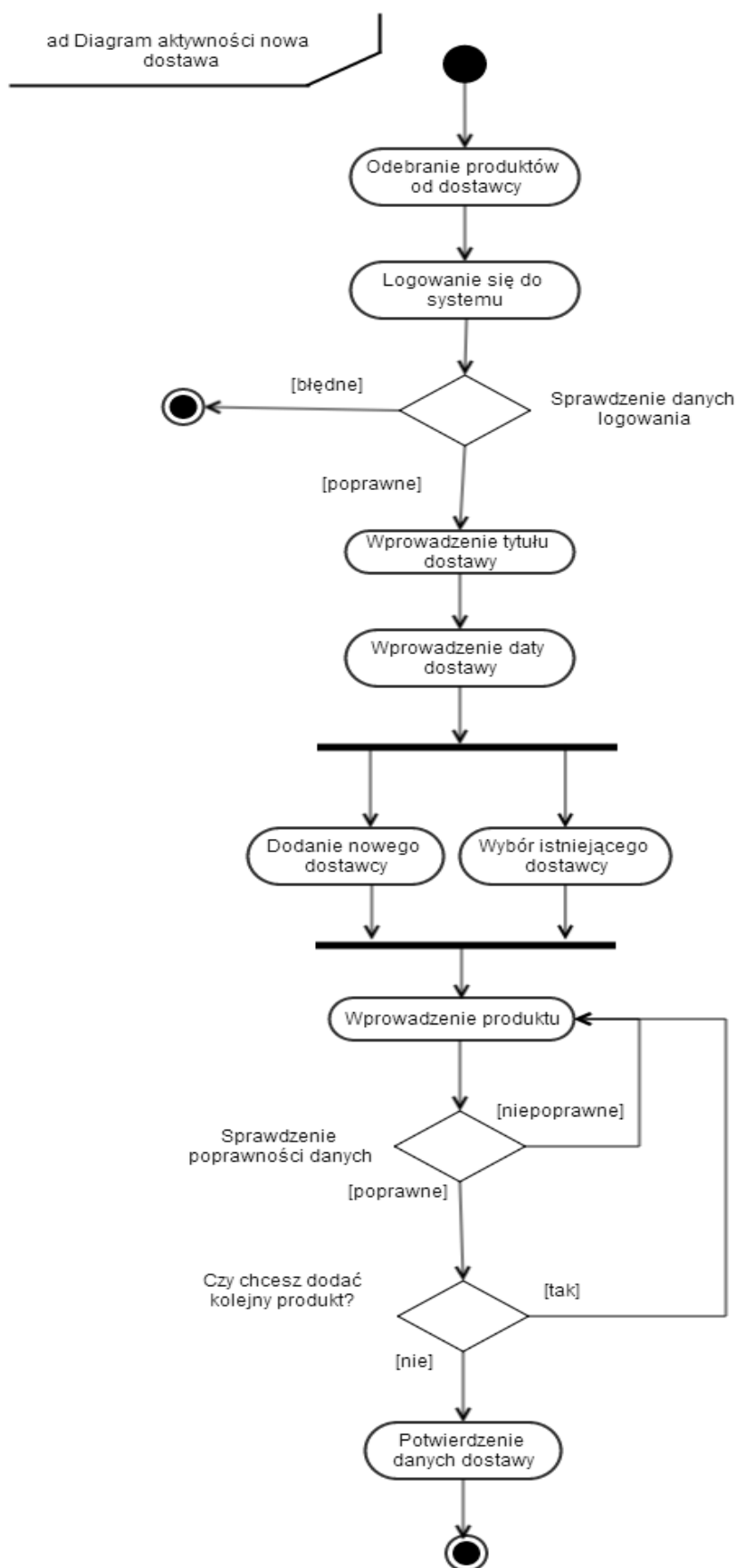




# Diagram stanu klasy



# Diagram aktywności



# Podsumowanie

Po analizie dynamicznej zdecydowałem iż należy dokonać następujących zmian w strukturze klas:

- Dodać do klasy Complaint, odpowiednie pole typu enum , które będą określać dokładniej status obiektu (ComplaintStatus).
- Dodać do klasy Product metody getCurrentTax() i getCurrentPricing() pozwalające na pobieranie
- Jako że persystencja będzie realizowana przez Hibernate, zaznaczyłem na diagramie projektowym identyfikatory encji
- Jako że aplikacja ma składać się z dwóch części (klienta webowego oraz webserviców REST) przyjąłem założenie że wszystkie klasy po stronie serwera o nazwie \*Resource (reprezentujące obiekt REST wg podejścia stosowanego przez twórców frameworka Restlet) będą konsumować i zwracać obiekty proste obiekty DTO, odpowiedzialne tylko za transfer danych
- Klasy \*Resource są odpowiedzialne tylko za : sparsowanie przychodzących danych (robi to właściwie [Jackson](#)): przekazaniem do transakcyjnych serwisów (np. DeliveryService), ewentualnego przechwycenia wyjątków gdy wystąpi błąd oraz przygotowaniem odpowiedniej odpowiedzi nagłówek http w zależności od przebiegu wywołania serwisu (np. 200, 301, 500) oraz sparsowanie obiektu do odpowiedzi JSON (znowu zrealizowane przez Jackson)
- Wszystkie operacje biznesowe są realizowane przez serwisy, metody serwisów powinny bezpośrednio konsumować dane przychodzące do webserwisu.
- Obiektu serwisowe są wspomagane przez obiekty implementujące SessionManager i PermissionChecker, które odpowiednio pozwalają zarządzać sesją użytkownika i sprawdzać czy posiada on wystarczające uprawnienia do wykonania operacji
- Dodatkowa warstwa DAO realizuje proste operacje na encjach
- Początkowo wydawać by się mogło że taka ilość warstw i obiektów to przerost formy nad treścią, jednak jak się okazało w czasie prac nad pracą inżynierską odseparowanie tak wielu elementów od siebie, przyniosło korzyści w postaci : schludnych, łatwych do utrzymania klas, odseparowanych od siebie, co ułatwia utrzymanie i dokonywanie zmian