

1. Sprawozdanie .....	2
1.1 Strona tytułowa .....	2
1.1.1 Konta aplikacji .....	4
1.1.2 Opis funkcjonalności podstawowej i dodatkowej .....	5
1.2 Sprawozdanie wstępne (20 pkt) .....	5
1.2.1 1. Dziedzina wykorzystania 2 pkt .....	5
1.2.2 2. Moduły funkcjonalne, aktorzy, przypadki użycia 4 pkt .....	8
1.2.3 3. Struktury relacyjnej bazy danych 2 pkt .....	18
1.2.4 4. Użytkownicy bazodanowi 2 pkt .....	37
1.2.5 5. Identyfikacja obiektów encji 2 pkt .....	39
1.2.6 6. Diagram klas encji 2 pkt .....	40
1.2.7 7. Diagramy sekwencji oraz identyfikacja transakcji bazodanowych 4 pkt .....	42
1.2.8 8. Konfiguracja uwierzytelniania w aplikacji 2 pkt .....	66
1.3 Sprawozdanie szczegółowe (25 pkt) .....	69
1.3.1 9. Diagram UML klas komponentów EJB 4 pkt .....	69
1.3.2 10. Schemat bezpieczeństwa komponentów EJB 5 pkt .....	75
1.3.3 11. Identyfikacja transakcji aplikacyjnych 6 pkt .....	91
1.3.4 12. Zgłaszane wyjątki 4 pkt .....	107
1.3.5 13. Interfejs użytkownika 3 pkt .....	109
1.3.6 14. Zabezpieczenia interfejsu użytkownika 3 pkt .....	115
1.3.7 15. Zmiany - projekt szczegółowy .....	121
1.4 Sprawozdanie końcowe (15 pkt) .....	122
1.4.1 16. Zabezpieczenie transmisji HTTP 1 pkt .....	122
1.4.2 17. Weryfikacja poprawności danych 3 pkt .....	123
1.4.3 18. Obsługa błędów 3 pkt .....	124
1.4.4 19. Wersje językowe interfejsu użytkownika 2 pkt .....	125
1.4.5 20. Opis przypadków użycia MVCv2 6 pkt .....	128
1.4.6 21. Zmiany - projekt końcowy .....	135

# Sprawozdanie

## SPRAWOZDANIE

Z REALIZACJI LABORATORIUM PRZEDMIOTU

SIECIOWE SYSTEMY BAZ DANYCH

EDYCJA 2018

### Strona tytułowa

Projekt SSBD01: "Kooperatywa sprzedaży produktów rolnych"

Członkowie grupy projektowej i funkcje projektowe:

- Michał Malec 203934 michalmalec, Funkcje pełnione w projekcie: szef, kontrola zgodności, kontrola jakości
  - Nietrywialne przypadki użycia: [MOP.3](#), [MOP.4](#)
- Agata Kania 203899 agkan, Funkcje pełnione w projekcie: baza, dokumentacja
  - Nietrywialne przypadki użycia: [MOP.1](#), [MOZ.4](#)
- Piotr Biernacki 203855 piotrekbie, Funkcje pełnione w projekcie: wdrożenie, kontrola jakości
  - Nietrywialne przypadki użycia: [MOZ.1](#), [MOP.2](#)
- Filip Różański 206275 fifi, Funkcje pełnione w projekcie: repozytorium, kontrola jakości
  - Nietrywialne przypadki użycia: [MOP.10](#), [MOZ.3](#)
- Dominik Lange 203924 dlange, Funkcje pełnione w projekcie: architektura, dokumentacja
  - Nietrywialne przypadki użycia: [MOZ.5](#), [MOP.7](#)

Zasoby:

- Zarządzanie projektem, system rejestracji zadań JIRA: <https://atlas.it.p.lodz.pl/jira>
- Repozytorium projektu Bitbucket: <https://atlas.it.p.lodz.pl/bitbucket>
- System gromadzenia i wymiany informacji Confluence: <https://atlas.it.p.lodz.pl/confluence>
- Witryna aplikacji : <http://studapp.it.p.lodz.pl:8001>
- Zarządzanie serwerem aplikacyjnym: <https://studapp.it.p.lodz.pl:4801>

Macierz decyzyjna przypadków użycia

P.U.	Opis	Gość	Użytkownik	Moderator	Administrator
<a href="#">MOK.1</a>	Zarejestruj	V			
<a href="#">MOK.2</a>	Utwórz konto				V
<a href="#">MOK.3</a>	Zablokuj konto			V	V
<a href="#">MOK.4</a>	Odblokuj konto			V	V
<a href="#">MOK.5</a>	Dołącz poziom dostępu do konta				V
<a href="#">MOK.6</a>	Odłącz poziom dostępu od konta				V
<a href="#">MOK.7</a>	Zmień własne hasło		V	V	V

MOK.8	Zmień hasło innego użytkownika				V
MOK.9	Edytuj dane własnego konta		V	V	V
MOK.10	Edytuj dane konta innego użytkownika				V
MOK.11	Wyloguj		V	V	V
MOK.12	Przeglądaj listę kont			V	V
MOK.13	Filtruj listę kont			V	V
MOK.14	Wyświetl statystyki konta użytkownika			V	V
MOP.1	Dodaj produkt		V		
MOP.2	Usuń produkt		V		
MOP.3	Edytuj produkt		V		
MOP.4	Zmień stan produktu (aktywny, nieaktywny)		V		
MOP.5	Przeglądaj listę produktów		V	V	V
MOP.6	Filtruj listę produktów		V	V	V
MOP.7	Dodaj kategorię produktów			V	
MOP.8	Przeglądaj listę własnych produktów		V		
MOP.9	Przeglądaj listę kategorii produktów			V	
MOP.10	Aktywuj/Dezaktywuj kategorię produktów			V	
MOZ.1	Dokonaj zakupu		V		
MOZ.2	Przeglądaj listę metod wysyłki			V	
MOZ.3	Dodaj metodę wysyłki			V	
MOZ.4	Aktywacja/dezaktywacja metod wysyłki			V	
MOZ.5	Zmień status zamówienia		V		
MOZ.6	Przeglądaj wszystkie zamówienia			V	
MOZ.7	Przeglądaj listę własnych zamówień		V		
MOZ.8	Filtruj listę z zamówieniami		V	V	

Żółte podświetlenie komórek oznacza **trywialne** przypadki użycia.

Podział funkcjonalności projektu na moduły i przypadki użycia

Moduł Obsługi Produktów (MOP)

P.U.	Opis	Osoba odpowiedzialna
MOP.1	Dodaj produkt	Agata Kania
MOP.2	Usuń produkt	Piotr Biernacki
MOP.3	Edytuj produkt	Michał Malec
MOP.4	Zmień stan produktu (aktywny, nieaktywny)	Michał Malec
MOP.5	Przeglądaj listę produktów	Agata Kania, Michał Malec
MOP.6	Filtruj listę produktów	Agata Kania
MOP.7	Dodaj kategorię produktów	Dominik Lange
MOP.8	Przeglądaj listę własnych produktów	Agata Kania, Michał Malec
MOP.9	Przeglądaj listę kategorii produktów	Dominik Lange, Filip Różański

MOP.10	Aktywuj/Dezaktywuj kategorię produktów	Filip Różański
--------	--	----------------

**Moduł Obsługi Zamówień (MOZ)**

P.U.	Opis	Osoba odpowiedzialna
MOZ.1	Dokonaj zakupu	Piotr Biernacki
MOZ.2	Przeglądaj listę metod wysyłki	Agata Kania, Michał Malec
MOZ.3	Dodaj metodę wysyłki	Filip Różański
MOZ.4	Aktywacja/dezaktywacja metod wysyłki	Agata Kania
MOZ.5	Zmień status zamówienia	Dominik Lange
MOZ.6	Przeglądaj wszystkie zamówienia	Dominik Lange, Michał Malec
MOZ.7	Przeglądaj listę własnych zamówień	Dominik Lange, Michał Malec
MOZ.8	Filtruj listę z zamówieniami	Dominik Lange

**Spis rozdziałów sprawozdania**

***Sprawozdanie wstępne***

- 1. Dziedzina wykorzystania
- 2. Moduły funkcjonalne, aktorzy, przypadki użycia
- 3. Struktury relacyjnej bazy danych
- 4. Użytkownicy bazodanowi
- 5. Identyfikacja obiektów encji
- 6. Diagram klas encji
- 7. Diagramy sekwencji oraz identyfikacja transakcji bazodanowych
- 8. Konfiguracja uwierzytelniania w aplikacji

***Sprawozdanie szczegółowe***

- 9. Diagram UML klas komponentów EJB
- 10. Schemat bezpieczeństwa komponentów EJB
- 11. Identyfikacja transakcji aplikacyjnych
- 12. Zgłaszane wyjątki
- 13. Interfejs użytkownika
- 14. Zabezpieczenia interfejsu użytkownika
- 15. Zmiany - projekt szczegółowy

***Sprawozdanie końcowe***

- 16. Zabezpieczenie transmisji HTTP
- 17. Weryfikacja poprawności danych
- 18. Obsługa błędów
- 19. Wersje językowe interfejsu użytkownika
- 20. Opis przypadków użycia MVCv2
- 21. Zmiany - projekt końcowy

**Konta aplikacji**

**Serwer aplikacyjny studapp**

Login: admin  
Hasło: yowfOrkAz3

**Użytkownicy bazy danych**

Login	Hasło	Odpowiedzialność
ssbd01glassfish	Pot5Rui9+3t	Logowanie użytkowników
ssbd01admin	adminSSBD01	Konto administracyjne

ssbd01mok	mokSSBD01	Obsługa mok
ssbd01mop	mopSSBD01	Obsługa mop
ssbd01moz	mozSSBD01	Obsługa moz

**Testowi użytkownicy systemu**

login	password	active	confirme	level
admin	ssbd01user1	true	true	admin, manager, user
client	ssbd01user2	true	true	user
manager	ssbd01user3	true	true	manager, user
manager2	ssbd01user4	true	true	manager
admin2	ssbd01user5	true	true	admin

**Opis funkcjonalności podstawowej i dodatkowej**

Podstawową funkcjonalność systemu będzie stanowiła możliwość sprzedawania i kupowania produktów przez użytkowników. Dostępne będą również możliwości zarządzania kontami, zamówieniami i produktami.

Dodatkowa funkcjonalność systemu będzie polegała na możliwości podglądu podstawowych statystyk użytkowników, opłacaniu zamówień, a także edycji i tworzeniu kategorii produktów i metod wysyłki.

**Sprawozdanie wstępne (20 pkt)**

**Sprawozdanie wstępne**

**1. Dziedzina wykorzystania 2 pkt**

**Przeznaczenie aplikacji**

Aplikacja będzie pozwalała użytkownikom na sprzedaż i kupno produktów rolnych wraz z możliwością dokonywania płatności. Wszystkie operacje wykonywane w systemie będą opierały się na danych pobranych z bazy danych jak i będą miały swoje odzwierciedlenie w niej. Aplikacja jest systemem OLTP (On-line Transaction Processing) ponieważ w ramach transakcji bazodanowych zapewnia ona zgodność danych ze stanem rzeczywistym obiektów istniejących w systemie, kładziony jest zatem nacisk na integralność i spójność danych w czasie, w ramach systemu wielodostępowego. Aplikacja zawsze będzie operowała na aktualnych danych.

**Słownik nazw obiektów w modelu danych**

Obiekt	Opis	Cykl życia	Zależności	Ograniczenia
--------	------	------------	------------	--------------

<b>Account</b>	Reprezentuje dane konta w systemie (również dane adresowe). Konto w chwili utworzenia posiada poziom dostępu Użytkownik, kolejne poziomy nadaje Administrator systemu.	Od momentu utworzenia konta bez ograniczenia. Konto może zostać utworzone przez Gościa lub Administratora systemu.	Powiązanie z wieloma poziomami dostępu oraz pojedynczym obiektem danych użytkownika.  Powiązanie również z: <ul style="list-style-type: none"> <li>zamówieniami jako sprzedawca lub kupujący</li> <li>produktami jako właściciel</li> </ul>	<ul style="list-style-type: none"> <li>Login, email oraz numer telefonu jest unikatowy w ramach systemu, a hasło spełnia wymogi bezpieczeństwa (co najmniej 8 znaków).</li> <li>Login przypisany do konta nie może być dłuższy niż 32 znaki.</li> <li>Imię użytkownika nie może być dłuższe niż 32 znaki.</li> <li>Nazwisko użytkownika nie może być dłuższe niż 32 znaki.</li> <li>Adres e-mail użytkownika nie może być dłuższy niż 64 znaki.</li> <li>Numer kontaktowy użytkownika nie może być dłuższy niż 18 znaków.</li> <li>Ulica użytkownika nie może być dłuższa niż 60 znaków.</li> <li>Numer budynku użytkownika nie może być dłuższy niż 10 znaków.</li> <li>Numer mieszkania użytkownika nie może być dłuższy niż 10 znaków (może nie zostać określony).</li> <li>Nazwa miasta użytkownika nie może być dłuższa niż 60 znaków.</li> <li>Nazwa państwa użytkownika nie może być dłuższa niż 60 znaków.</li> </ul>
<b>AccessLevel</b>	Określa, do których zasobów aplikacji użytkownik ma dostęp. Konto w chwili uwierzytelnienia będzie posiadało wszystkie nadane mu poziomy dostępu. Możliwe poziomy dostępu to: <ul style="list-style-type: none"> <li>Administrator</li> <li>Moderator</li> <li>Użytkownik</li> </ul>	Od chwili uruchomienia systemu, be z ograniczenia końcowego.	Wielokrotne powiązanie z AccountAlevel.	<ul style="list-style-type: none"> <li>Brak możliwości usunięcia, dodania nowego i edycji poziomu dostępu.</li> <li>Nazwa poziomu dostępu nie może być dłuższa niż 16 znaków.</li> </ul>
<b>AccountAlevel</b>	Określa powiązanie konta użytkownika z jego poziomem dostępu.	Od momentu utworzenia konta, z którym jest powiązane do momentu odebrania poziomu dostępu.	Wielokrotne powiązanie z kontem użytkownika oraz poziomem dostępu.	<ul style="list-style-type: none"> <li>Identyfikatory należą do istniejącego użytkownika i poziomów dostępu.</li> <li>Konto aktywne musi posiadać co najmniej jeden poziom dostępu.</li> </ul>
<b>Order1</b>	Określa postęp transakcji handlowej między użytkownikami.	Od momentu zakupu, czyli utworzenia nowego zamówienia, bez ograniczenia końcowego.	Pojedyncze powiązanie z : <ul style="list-style-type: none"> <li>Statusem zamówienia</li> <li>Sposobem i adresem dostawy</li> <li>Kontem kupującego</li> <li>Kontem sprzedającego</li> </ul> Wielokrotne powiązanie z Listą zamówionych produktów.	<ul style="list-style-type: none"> <li>Maksymalna wartość zamówienia to 99999999.99, a minimalna 0 zł.</li> </ul>
<b>OrderStatus</b>	Określa stan zamówienia.	Od chwili uruchomienia systemu, bez ograniczenia końcowego.	Powiązanie z zamówieniami.	<ul style="list-style-type: none"> <li>Brak możliwości ich usunięcia, modyfikacji i dodania nowych.</li> <li>Nazwa statusu zamówienia nie może być dłuższa niż 16 znaków.</li> </ul>
<b>OrderProducts</b>	Reprezentuje powiązanie zamówienia z produktami, które obejmuje.	Od momentu utworzenia zamówienia, bez ograniczenia końcowego.	Powiązanie z zamówieniami.	<ul style="list-style-type: none"> <li>Nazwa produktu nie może być dłuższa niż 32 znaki.</li> <li>Ilość produktu nie może być większa niż 999999.999, ani równa lub mniejsza od zera.</li> <li>Wartość produktu nie może przekroczyć 9999999.99, ani być mniejsza od zera.</li> </ul>

<b>OrderShipping</b>	Określa informacje dotyczące metody dostawy oraz osoby do której zostanie dostarczone zamówienie.	Od momentu opłacenia zamówienia, bez ograniczenia końcowego.	Pojedyncze powiązanie z zamówieniami.	<ul style="list-style-type: none"> <li>Nazwa metody dostawy nie może być dłuższa niż 16 znaków.</li> <li>Imię zamawiającego nie może być dłuższe niż 32 znaki.</li> <li>Nazwisko zamawiającego nie może być dłuższe niż 32 znaki.</li> <li>Ulica zamawiającego nie może być dłuższa niż 60 znaków.</li> <li>Numer budynku zamawiającego nie może być dłuższy niż 10 znaków.</li> <li>Numer mieszkania zamawiającego nie może być dłuższy niż 10 znaków.</li> <li>Nazwa miasta zamawiającego nie może być dłuższa niż 60 znaków.</li> <li>Nazwa państwa zamawiającego nie może być dłuższa niż 60 znaków.</li> <li>Wszystkie powyższe pola poza numerem mieszkania muszą być wypełnione.</li> </ul>
<b>ShippingMethod</b>	Określa formę dostarczenia produktu do użytkownika.	Od momentu dodania metody dostawy, bez ograniczenia końcowego.	Brak.	<ul style="list-style-type: none"> <li>Nazwa sposoby dostawy nie dłuższa niż 16 znaków.</li> <li>Maksymalna kwota to 999.99 zł, a minimalna 0 zł.</li> </ul>
<b>Product</b>	Reprezentuje produkt rolny w systemie, który użytkownicy mogą zakupić.	Od momentu utworzenia produktu przez użytkownika, bez ograniczenia końcowego.	Pojedyncze powiązanie z kategorią, jednostką i danymi użytkownika.	<ul style="list-style-type: none"> <li>Nazwa produktu nie może być dłuższa niż 32 znaków.</li> <li>Opis produktu nie może być dłuższy niż 800 znaków.</li> <li>Maksymalna kwota to 999999.99 zł, a minimalna 0 zł.</li> <li>Liczba produktów nie może być większa niż 999999.999</li> </ul>
<b>Category</b>	Określa kategorię produktu. Jest zarządzana przez konto z poziomem dostępu Moderator.	Od momentu utworzenia kategorii, bez ograniczenia końcowego.	Wielokrotne powiązanie z produktami.	<ul style="list-style-type: none"> <li>Nazwa kategorii nie dłuższa niż 16 znaków.</li> </ul>
<b>Unit</b>	Określa jednostkę, zgodnie z którą określana jest liczebność danego produktu.	Od momentu utworzenia jednostki, bez ograniczenia końcowego.	Wielokrotne powiązanie z produktami.	<ul style="list-style-type: none"> <li>Nazwa jednostki nie dłuższa niż 16 znaków.</li> </ul>

Żadnej z krotek bazy danych oprócz krotek tabeli account\_alevel nie można usunąć.

#### Słownik pozostałych pojęć i terminów

Obiekt	Opis
<b>Administrator</b>	Konto posiadające poziom dostępu oferujący możliwość korzystania z funkcji systemu określonych na <a href="#">stronie głównej</a> .
<b>Moderator</b>	Konto posiadające poziom dostępu oferujący możliwość korzystania z funkcji systemu określonych na <a href="#">stronie głównej</a> .
<b>Uzytkownik</b>	Konto posiadające poziom dostępu oferujący możliwość korzystania z funkcji systemu określonych na <a href="#">stronie głównej</a> . <i>Konto to określany jest również jako "User".</i>
<b>Gosc</b>	Osoba przed procesem uwierzytelnienia lub z negatywnym jego wynikiem. Ma możliwość korzystania z funkcji systemu określonych na <a href="#">stronie głównej</a> .

Produkt rolny	Produkt, który został umieszczony w systemie przez użytkownika zamiarem sprzedania go.
Warunki bezpieczeństwa hasła	Hasło spełniające te warunki powinno być łańcuchem znaków składającym się z co najmniej 8 znaków.

## 2. Moduły funkcjonalne, aktorzy, przypadki użycia 4 pkt

Lista poziomów dostępu (aktorów)

Aktor	Opis
Gość	<ul style="list-style-type: none"> <li>Jest to osoba, która odwiedza strony systemu bez wcześniejszego zalogowania.</li> <li>Gość ma możliwość rejestracji, a także późniejszego zalogowania.</li> </ul>
Użytkownik	<ul style="list-style-type: none"> <li>Jest to osoba, która przeszła proces rejestracji (wraz z aktywacją konta) i logowania.</li> <li>Ma możliwość sprzedawania i kupowania produktów, zarządzania własnymi produktami i zamówieniami, a także własnym kontem.</li> <li>Użytkownik jest domyślnie nadawanym poziomem dostępu po samodzielnym zarejestrowaniu w systemie.</li> </ul>
Moderator	<ul style="list-style-type: none"> <li>Jest to osoba, która przeszła proces rejestracji (wraz z aktywacją konta) i logowania, a także otrzymała od Administratora poziom dostępu Moderator.</li> <li>Moderator jest odpowiedzialny za podstawowe zarządzanie systemem, do którego należy między innymi: edycja i dodawanie kategorii produktów, edycja i dodawanie metod wysyłki, przeglądanie statystyk kont użytkowników, a także blokowanie i odblokowywanie innych kont.</li> </ul>
Administrator	<ul style="list-style-type: none"> <li>Jest to osoba, która przeszła proces rejestracji (wraz z aktywacją konta) i logowania, a także otrzymała od innego Administratora poziom dostępu Administrator.</li> <li>Administrator ma możliwość rejestracji innych użytkowników, a także modyfikacji większości ich danych (w tym hasła).</li> <li>Ważną funkcją Administratora jest możliwość przydzielania i odbierania poziomów dostępu.</li> </ul>

Poziom dostępu Gość wyklucza się z pozostałymi poziomami dostępu. Konto posiadając poziom dostępu Użytkownik lub Moderator lub Administrator nie może jednocześnie posiadać poziomu dostępu Gość.

Diagram przypadków użycia dla aktora Gość



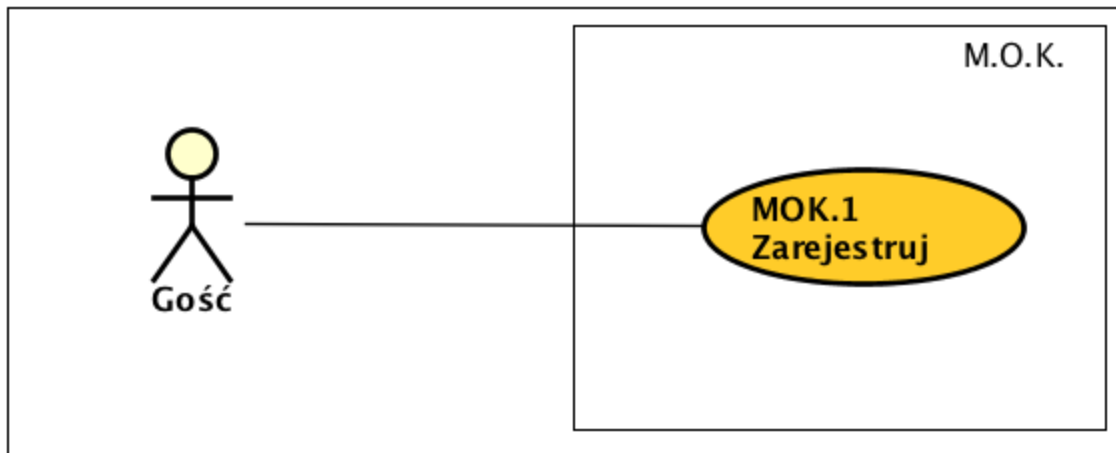


Diagram przypadków użycia dla aktora Użytkownik

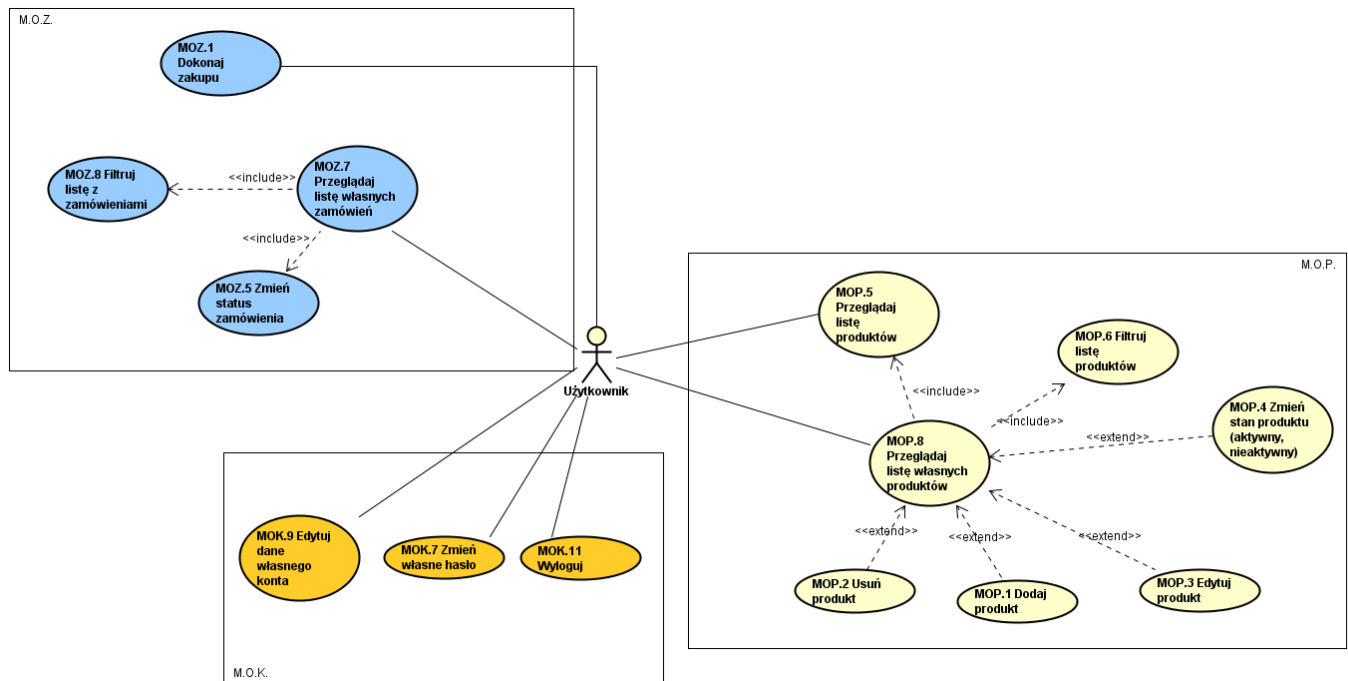


Diagram przypadków użycia dla aktora Moderator

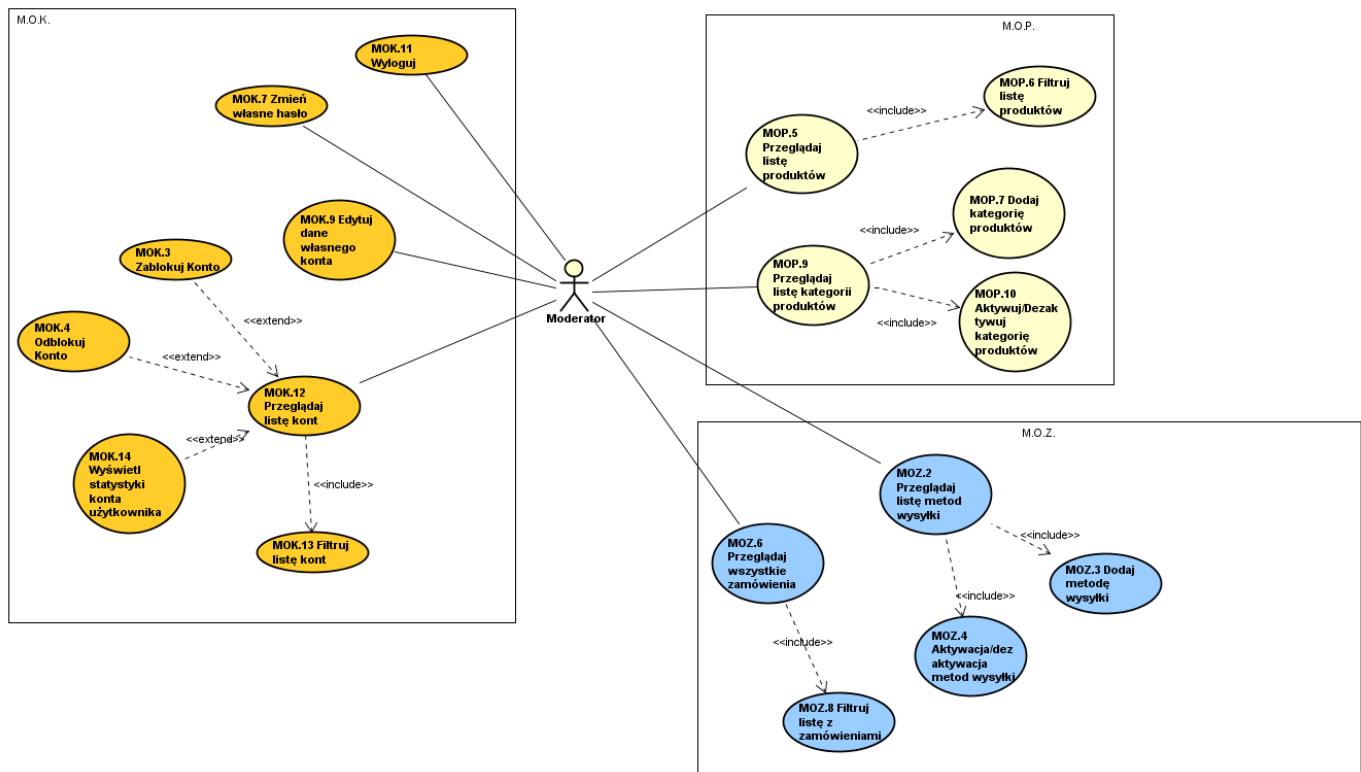
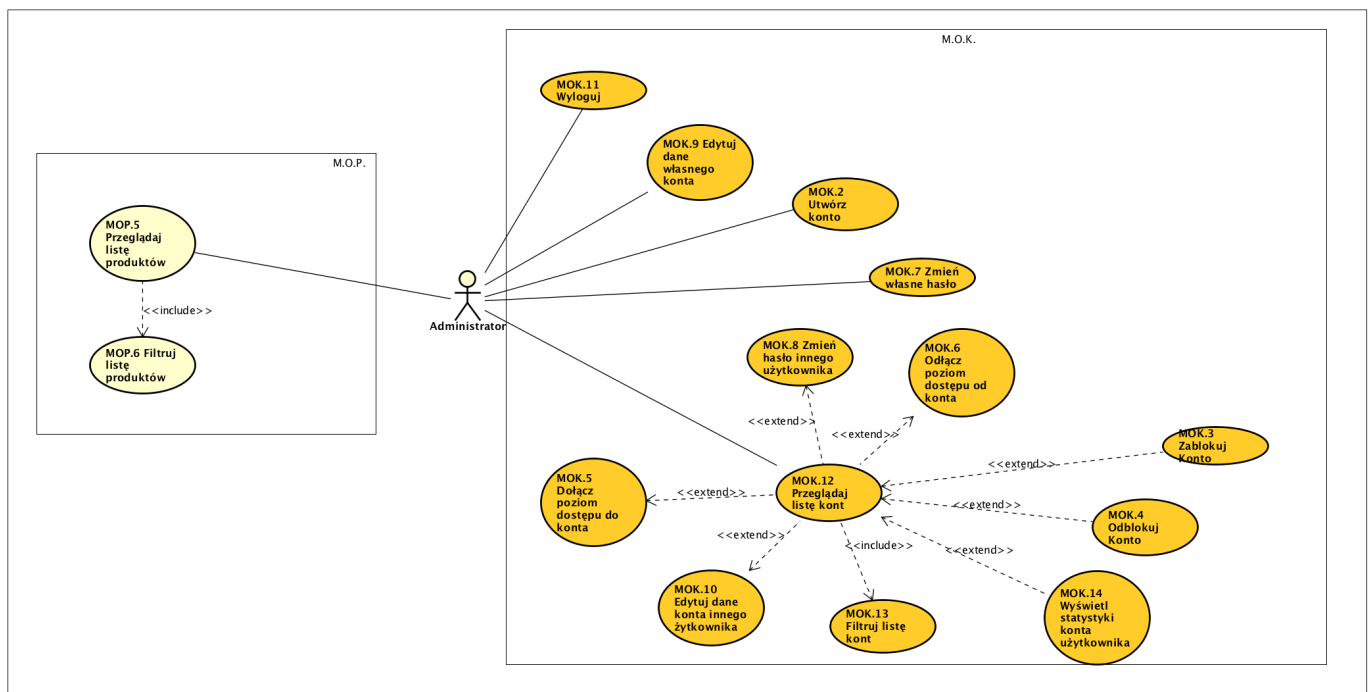


Diagram przypadków użycia dla aktora Administrator



Scenariusze błędów stałe dla każdego przypadku użycia
Błąd połączenia z bazą danych
Brak odpowiedzi serwera
W przypadku modyfikacji obiektu: ten sam obiekt został zmodyfikowany wcześniej przez innego użytkownika

P.U.	Opis	Scenariusze główne	Scenariusze błędów
MOK.1	Zarejestruj	<ol style="list-style-type: none"> <li>Scenariusz wykonywany jest przez użytkownika Gość.</li> <li>System wyświetla formularz rejestracji zawierający pola: <ol style="list-style-type: none"> <li>login</li> <li>hasło</li> <li>powtórne hasło</li> <li>adres e-mail użytkownika</li> <li>imię</li> <li>nazwisko</li> <li>adres</li> </ol> </li> <li>Użytkownik wypełnia formularz i zatwierdza.</li> <li>System wysyła wiadomość z hiperłączem aktywacyjnym na podany adres e-mail</li> <li>System wyświetla informację o wysłaniu wiadomości e-mail.</li> <li>Użytkownik potwierdza rejestrację poprzez odnośnik w wiadomości e-mail.</li> </ol>	<p>Błędy mogą wystąpić w formularzu rejestracji nowego użytkownika.</p> <ol style="list-style-type: none"> <li>Dane wprowadzone do formularza nie spełniają warunków walidacji. - Wyświetlenie komunikatu objaśniającego reguły walidacji danego pola. <ol style="list-style-type: none"> <li>Konto użytkownika o podanym loginie lub adresie e-mail już istnieje.</li> </ol> </li> <li>Nie aktywowano konta poprzez odnośnik aktywacyjny wysłany na skrzynkę mailową.</li> </ol>
MOK.2	Utwórz konto	<ol style="list-style-type: none"> <li>Scenariusz wykonywany jest przez użytkownika Administrator.</li> <li>System wyświetla formularz rejestracji zawierający pola: <ol style="list-style-type: none"> <li>login</li> <li>hasło</li> <li>powtórne hasło</li> <li>adres e-mail użytkownika</li> <li>imię</li> <li>nazwisko</li> <li>adres</li> </ol> </li> <li>Użytkownik wypełnia formularz i zatwierdza.</li> <li>System wysyła wiadomość z hiperłączem aktywacyjnym na podany adres e-mail</li> <li>System wyświetla informację o wysłaniu wiadomości e-mail.</li> <li>Użytkownik utworzonego konta potwierdza rejestrację poprzez odnośnik w wiadomości e-mail.</li> </ol>	<p>Błędy mogą wystąpić w formularzu tworzenia nowego użytkownika.</p> <ol style="list-style-type: none"> <li>Błędnie wprowadzone dane użytkownika</li> <li>Konto użytkownika o podanym loginie lub adresie e-mail już istnieje.</li> <li>Nie aktywowano konta poprzez odnośnik aktywacyjny wysłany na skrzynkę mailową.</li> </ol>

MOK.3	Zablokuj konto	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Administrator/Moderator.</li> <li>2. System wyświetla listę statystyk wszystkich kont.</li> <li>3. W każdym wierszu listy umieszczone jest pole wyboru oznaczające, czy konto jest aktywne.</li> <li>4. Użytkownik zmienia wartość pola wyboru na wartość prawdziwą dla wybranych użytkowników.</li> <li>5. Użytkownik wysyła formularz</li> </ol>	
MOK.4	Odblokuj konto	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Administrator/Moderator.</li> <li>2. System wyświetla listę statystyk wszystkich kont.</li> <li>3. W każdym wierszu listy umieszczone jest pole wyboru oznaczające, czy konto jest aktywne.</li> <li>4. Użytkownik zmienia wartość pola wyboru na wartość fałszywą dla wybranych użytkowników.</li> <li>5. Użytkownik wysyła formularz</li> </ol>	
MOK.5	Dołącz poziom dostępu do konta	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Administrator.</li> <li>2. System wyświetla listę wszystkich użytkowników.</li> <li>3. Użytkownik wybiera użytkownika, którego chce zmodyfikować z listy.</li> <li>4. Każdy wiersz listy zawiera listę wielokrotnego wyboru z dostępnymi poziomami dostępu.</li> <li>5. Użytkownik dodaje poziom dostępu poprzez zaznaczenie odpowiednich poziomów dostępu i wciśnięcie przycisku "Zatwierdź poziomy dostęp".</li> </ol>	
MOK.6	Odlącz poziom dostępu od konta	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Administrator.</li> <li>2. System wyświetla listę wszystkich użytkowników.</li> <li>3. Użytkownik wybiera użytkownika, którego chce zmodyfikować z listy.</li> <li>4. Każdy wiersz listy zawiera listę wielokrotnego wyboru z dostępnymi poziomami dostępu.</li> <li>5. Użytkownik usuwa poziom dostępu poprzez zaznaczenie odpowiednich poziomów dostępu i wciśnięcie przycisku "Wyślij".</li> </ol>	

MOK.7	Zmień własne hasło	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Administrator/Moderator /Użytkownik.</li> <li>2. System wyświetla użytkownikowi formularz zawierający następujące pola: <ol style="list-style-type: none"> <li>1. Stare hasło</li> <li>2. Nowe hasło</li> <li>3. Powtórne hasło</li> </ol> </li> <li>3. Użytkownik wypełnia wszystkie pola formularza</li> <li>4. Użytkownik wysyła formularz, tym samym zmieniając hasło.</li> </ol>	<p>Błędy mogą wystąpić w formularzu zmiany hasła.</p> <ol style="list-style-type: none"> <li>1. Podane hasło nie spełnia warunków bezpieczeństwa.</li> <li>2. Podane stare hasło jest niepoprawne.</li> </ol>
MOK.8	Zmień hasło innego użytkownika	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Administrator.</li> <li>2. System wyświetla użytkownikowi formularz zawierający następujące pola: <ol style="list-style-type: none"> <li>1. login (login użytkownika, którego hasło ma być zmienione)</li> <li>2. Nowe hasło</li> <li>3. Powtórne hasło</li> </ol> </li> <li>3. Użytkownik wypełnia wszystkie pola formularza</li> <li>4. Użytkownik wysyła formularz, tym samym zmieniając hasło.</li> </ol>	<p>Błędy mogą wystąpić w formularzu zmiany hasła.</p> <ol style="list-style-type: none"> <li>1. Podane hasło nie spełnia warunków bezpieczeństwa.</li> </ol>
MOK.9	Edytuj dane własnego konta	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Administrator/Moderator /Użytkownik.</li> <li>2. System wyświetla formularz zawierający następujące pola: <ol style="list-style-type: none"> <li>1. imię</li> <li>2. nazwisko</li> <li>3. adres e-mail</li> <li>4. numer telefonu</li> <li>5. adres <ol style="list-style-type: none"> <li>1. ulica</li> <li>2. numer domu</li> <li>3. numer mieszkania</li> <li>4. kod pocztowy</li> <li>5. miasto</li> <li>6. kraj</li> </ol> </li> </ol> </li> <li>3. Użytkownik wypełnia wszystkie pola formularza</li> <li>4. Użytkownik wysyła formularz, tym samym zmieniając dane.</li> </ol>	<p>Błędy mogą wystąpić w formularzu edycji danych własnego konta.</p> <ol style="list-style-type: none"> <li>1. Podane dane są nieprawidłowe.</li> <li>2. Nie podano wartości w obowiązkowym polu formularza.</li> </ol>

MOK.10	Edytuj dane konta innego użytkownika	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Administrator.</li> <li>2. System wyświetla formularz zawierający następujące pola: <ol style="list-style-type: none"> <li>1. imię</li> <li>2. nazwisko</li> <li>3. adres e-mail</li> <li>4. numer telefonu</li> <li>5. adres <ol style="list-style-type: none"> <li>1. ulica</li> <li>2. numer domu</li> <li>3. numer mieszkania</li> <li>4. kod pocztowy</li> <li>5. miasto</li> <li>6. kraj</li> </ol> </li> </ol> </li> <li>3. Użytkownik wypełnia wszystkie pola formularza</li> <li>4. Użytkownik wysyła formularz, tym samym zmieniając dane.</li> </ol>	<p>Błędy mogą wystąpić w widoku "Edycja danych konta".</p> <ol style="list-style-type: none"> <li>1. Podane dane są nieprawidłowe.</li> <li>2. Nie podano wartości w obowiązkowym polu formularza.</li> </ol>
MOK.11	Wyloguj	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Administrator/Moderator /Użytkownik.</li> <li>2. W każdym widoku, w prawym górnym rogu system wyświetla przycisk "Wyloguj".</li> <li>3. Użytkownik dokonuje wylogowania wciskając przycisk "Wyloguj".</li> </ol>	
MOK.12	Przeglądaj listę kont	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Administrator/Moderator.</li> <li>2. System wyświetla listę kont użytkowników.</li> </ol>	
MOK.13	Filtruj listę kont	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Administrator/Moderator.</li> <li>2. System wyświetla listę kont użytkowników wraz z polem służącym do wyszukiwania użytkowników na podstawie loginu.</li> <li>3. Użytkownik wypełnia formularz.</li> <li>4. Użytkownik wysyła formularz.</li> <li>5. System aktualizuje listę użytkowników wyświetlając jedynie tych, których login zawiera podane znaki.</li> </ol>	<p>Komunikat błędu zostanie wyświetlony w przypadku:</p> <ol style="list-style-type: none"> <li>1. Braku kont spełniających podane kryterium filtrowania</li> <li>2. Nieprawidłowego podania wzorca do filtrowania</li> </ol>

MOK.14	Wyświetl statystyki konta użytkownika	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Administrator/Moderator.</li> <li>2. System wyświetla widok statystyk użytkownika: <ol style="list-style-type: none"> <li>1. liczbę produktów wystawionych przez użytkownika</li> <li>2. liczbę zamówień użytkownika</li> <li>3. liczbę wizyt w serwisie</li> </ol> </li> </ol>	<p>Komunikat błędu zostanie wyświetlony w przypadku:</p> <ol style="list-style-type: none"> <li>1. Braku istnienia podanego użytkownika w bazie danych</li> </ol>
MOP.1	Dodaj produkt	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Użytkownik.</li> <li>2. System wyświetla formularz zawierający pola: <ol style="list-style-type: none"> <li>1. nazwa</li> <li>2. kategoria (lista wyboru z wartości pobranych z bazy danych)</li> <li>3. opis</li> <li>4. jednostka (lista wyboru z wartości pobranych z bazy danych)</li> <li>5. cena</li> <li>6. ilość (należy podać liczbę całkowitą)</li> </ol> </li> <li>3. Użytkownik wypełnia formularz.</li> <li>4. Użytkownik wysyła formularz.</li> </ol>	<p>Komunikat błędu zostanie wyświetlony w przypadku:</p> <ol style="list-style-type: none"> <li>1. Niewypełnieniu lub nieprawidłowym wypełnieniu pól produktu</li> </ol>
MOP.2	Usuń produkt	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Użytkownik.</li> <li>2. System wyświetla widok zawierający listę produktów wraz z polem wyboru przy każdym elemencie listy. Pod listą znajduje się przycisk "Usuń".</li> <li>3. Użytkownik zaznacza produkty, które zamierza usunąć.</li> <li>4. Użytkownik wciska przycisk "Usuń".</li> </ol>	<p>Komunikat błędu zostanie wyświetlony w przypadku:</p> <ol style="list-style-type: none"> <li>1. Usunięcia lub modyfikacja produktu wcześniej przez innego użytkownika</li> </ol>

MOP.3	Edytuj produkt	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Użytkownik.</li> <li>2. System wyświetla formularz zawierający pola: <ol style="list-style-type: none"> <li>1. nazwa</li> <li>2. kategoria (lista wyboru z wartości pobranych z bazy danych)</li> <li>3. opis</li> <li>4. jednostka (lista wyboru z wartości pobranych z bazy danych)</li> <li>5. cena</li> <li>6. ilość (należy podać liczbę całkowitą)</li> </ol> </li> <li>3. Użytkownik wypełnia formularz.</li> <li>4. Użytkownik wysła formularz.</li> </ol>	<p>Komunikat błędu zostanie wyświetlony w przypadku:</p> <ol style="list-style-type: none"> <li>1. Wprowadzenia nieprawidłowych danych</li> <li>2. Niewprowadzenia danych obowiązkowych</li> </ol>
MOP.4	Zmień stan produktu (aktywny, nieaktywny)	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Użytkownik.</li> <li>2. System wyświetla formularz zmiany stanu produktu zawierający listę wyboru, która zawiera stany produktu oraz przycisk "Zatwierdź".</li> <li>3. Użytkownik wybiera stan produktu z listy.</li> <li>4. Użytkownik wciska przycisk "Zatwierdź".</li> </ol>	<p>Komunikat błędu zostanie wyświetlony w przypadku:</p> <ol style="list-style-type: none"> <li>1. Nieprawidłowego określania stanu (stan produktu został zmieniony uprzednio przez innego użytkownika).</li> </ol>
MOP.5	Przeglądaj listę produktów	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Administrator/Moderator /Użytkownik.</li> <li>2. System wyświetla listę produktów.</li> </ol>	
MOP.6	Filtruj listę produktów	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Administrator/Moderator /Użytkownik.</li> <li>2. System wyświetla listę produktów wraz z polem służącym do wyszukiwania produktów na podstawie nazwy.</li> <li>3. Użytkownik wypełnia formularz.</li> <li>4. Użytkownik wysła formularz.</li> <li>5. System aktualizuje listę produktów wyświetlając jedynie te, których nazwa zawiera podane znaki.</li> </ol>	<p>Komunikat błędu zostanie wyświetlony w przypadku:</p> <ol style="list-style-type: none"> <li>1. Braku produktów</li> <li>2. Braku produktów spełniających podane kryterium</li> <li>3. Nieprawidłowego wskazania wzorca</li> </ol>



MOP.7	Dodaj kategorię produktów	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Moderator.</li> <li>2. System wyświetla widok edycji kategorii produktów, który zawiera: <ol style="list-style-type: none"> <li>1. formularz z polem umożliwiającym wybranie nazwy nowej kategorii</li> </ol> </li> <li>3. Użytkownik wypełnia pola formularzy.</li> <li>4. Użytkownik wysyła formularz.</li> </ol>	<p>Komunikat błędu zostanie wyświetlony w widoku listy kategorii produktów.</p> <ol style="list-style-type: none"> <li>1. Kategoria o podanej nazwie już istnieje.</li> <li>2. Długość nazwy kategorii jest niepoprawna.</li> </ol>
MOP.8	Przeglądaj listę własnych produktów	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Użytkownik.</li> <li>2. System wyświetla listę wszystkich produktów wystawionych przez zalogowanego użytkownika.</li> </ol>	
MOP.9	Przeglądaj listę kategorii produktów	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany przez użytkownika Moderator.</li> <li>2. System wyświetla listę wszystkich kategorii produktów dostępnych w systemie.</li> </ol>	
MOP.10	Aktywuj/Dezaktywuj kategorię produktów	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Moderator.</li> <li>2. System wyświetla tabelę z listą wszystkich kategorii. Każdy wiersz tabeli zawiera formularz z polem wyboru umożliwiającym aktywację lub dezaktywację kategorii produktów.</li> <li>3. Użytkownik wypełnia pola formularzy.</li> <li>4. Użytkownik wysyła formularz.</li> </ol>	
MOZ.1	Dokonaj zakupu	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Użytkownik.</li> <li>2. System wyświetla widok listy produktów, na której każdy element zawiera przycisk "Kup".</li> <li>3. Użytkownik wybiera rodzaj dostawy.</li> <li>4. Użytkownik wciska przycisk "Kup".</li> <li>5. System wyświetla widok z informacją o pomyślnym zakupie oraz prośbą o dokonanie płatności.</li> </ol>	<p>Komunikaty o błędach zostaną wyświetlone w widoku listy produktów.</p> <ol style="list-style-type: none"> <li>1. Produkt został już wyprzedany.</li> <li>2. Produkty zdefiniowano w liczbie większej niż dostępność w magazynie.</li> <li>3. Nie wybrano rodzaju dostawy.</li> </ol>
MOZ.2	Przeglądaj listę metod wysyłki	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany przez użytkownika Moderator.</li> <li>2. System wyświetla listę wszystkich metod wysyłki dostępnych w systemie.</li> </ol>	

MOZ.3	Dodaj metodę wysyłki	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Moderator.</li> <li>2. System wyświetla formularz z polami "nazwa" oraz "koszt"</li> <li>3. Użytkownik wypełnia pola formularzy.</li> <li>4. Użytkownik wysyła formularz.</li> </ol>	<p>Komunikaty o błędach zostaną wyświetlone w widoku listy metod wysyłki..</p> <ol style="list-style-type: none"> <li>1. Metoda wysyłki już istnieje.</li> <li>2. Wprowadzono nieprawidłowe wartości pól formularza.</li> </ol>
MOZ.4	Aktywacja/deaktywacja metod wysyłki	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Moderator.</li> <li>2. System wyświetla widok listy metod wysyłki. Każdy element listy zawiera pole wyboru określające, czy metoda wysyłki jest aktywna, czy nie.</li> <li>3. Użytkownik wypełnia pola formularzy.</li> <li>4. Użytkownik wysyła formularz.</li> </ol>	<p>Komunikaty będą wyświetlone w widoku edycji metod wysyłki.</p> <ol style="list-style-type: none"> <li>1. Nieprawidłowe dane wprowadzone do formularza.</li> </ol>
MOZ.5	Zmiana statusu zamówienia	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Użytkownik.</li> <li>2. System wyświetla formularz zmiany statusu danego zamówienia, który zawiera listę wyboru z dostępnymi opcjami statusu zamówienia.</li> </ol>	<p>Komunikaty o błędach zostaną wyświetlone w widoku danego zamówienia.</p> <ol style="list-style-type: none"> <li>1. Zamówienie zostało anulowane.</li> </ol>
MOZ.6	Przeglądaj wszystkie zamówienia	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Moderator.</li> <li>2. System wyświetla listę wszystkich zamówień.</li> </ol>	
MOZ.7	Przeglądaj listę własnych zamówień	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Użytkownik.</li> <li>2. System wyświetla listę wszystkich zamówień należących do zalogowanego użytkownika.</li> </ol>	
MOZ.8	Filtruj listę z zamówieniami	<ol style="list-style-type: none"> <li>1. Scenariusz wykonywany jest przez użytkownika Użytkownik/Moderator.</li> <li>2. System wyświetla formularz zawierający pola: <ol style="list-style-type: none"> <li>1. data rozpoczęcia</li> <li>2. cena</li> </ol> </li> <li>3. Użytkownik wypełnia formularz</li> <li>4. Użytkownik zatwierdza formularz</li> <li>5. System wyświetla listę zamówień zgodną z podanymi przez użytkownika kryteriami.</li> </ol>	

### 3 Struktury relacyjnej bazy danych 2 pkt

#### Struktury tabel

## Moduł Obsługi Klienta (MOK)

Tabela: account

Kolumna	Typ	Nullowalne	Znaczenie	Domyślnie
id	bigint	not null	id konta użytkownika	
login	character varying(32)	not null	login użytkownika	
password	character(64)	not null	hasło użytkownika	
confirm	boolean	not null	czy konto zostało potwierdzone	false
active	boolean	not null	czy konto jest aktywne	true
number_of_products	bigint	not null	ilość produktów użytkownika	0
number_of_orders	bigint	not null	ilość zamówień użytkownika	0
number_of_logins	bigint	not null	ilość logowań użytkownika	0
version	bigint	not null	numer wersji	
last_login_date	timestamp without time zone	null	data ostatniego logowania	
last_login_ip	character varying(15)	null	adres IP, z którego nastąpiło ostatnie uwierzytelnienie	

Indeksy:

**"account\_pk" PRIMARY KEY, btree (id)**

"login\_unique" UNIQUE CONSTRAINT, btree (login)

Wskazywany przez:

TABLE "shipping\_method" CONSTRAINT "account\_fk" FOREIGN KEY (created\_by) REFERENCES account(id)

TABLE "account\_alevel" CONSTRAINT "account\_fk" FOREIGN KEY (id\_account) REFERENCES account(id)

TABLE "archival\_password" CONSTRAINT "archival\_password\_id\_account\_fkey" FOREIGN KEY (id\_account) REFERENCES account(id)

TABLE ""order"" CONSTRAINT "order\_buyer\_user\_fk" FOREIGN KEY (buyer\_id) REFERENCES account(id)

TABLE ""order"" CONSTRAINT "order\_seller\_user\_fk" FOREIGN KEY (seller\_id) REFERENCES account(id)

TABLE "product" CONSTRAINT "product\_owner\_fk" FOREIGN KEY (owner\_id) REFERENCES account(id)

TABLE "user\_data" CONSTRAINT "user\_data\_account\_fk" FOREIGN KEY (id) REFERENCES account(id)

TABLE "verification\_token" CONSTRAINT "verification\_token\_account\_fk" FOREIGN KEY (id\_account) REFERENCES account(id)

## account

```
CREATE TABLE public.account
(
    id bigint NOT NULL,
    login character varying(32) COLLATE pg_catalog."default" NOT NULL,
    password character(64) COLLATE pg_catalog."default" NOT NULL,
    confirm boolean NOT NULL DEFAULT false,
    active boolean NOT NULL DEFAULT true,
    number_of_products bigint NOT NULL DEFAULT 0,
    number_of_orders bigint NOT NULL DEFAULT 0,
    number_of_logins bigint NOT NULL DEFAULT 0,
    version bigint NOT NULL,
    last_login_date timestamp without time zone,
    last_login_ip character varying(15) COLLATE pg_catalog."default",
    CONSTRAINT account_pk PRIMARY KEY (id),
    CONSTRAINT login_unique UNIQUE (login)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;
```

Tabela: access\_level

Kolumna	Typ	Nullowalne	Znaczenie	Domyślnie
id	bigint	not null	id poziomu dostępu	
level	character varying(16)	not null	nazwa poziomu	
active	boolean	not null	czy poziom jest aktywny	true
version	bigint	not null	numer wersji	

Indeksy:

**"access\_level\_pk" PRIMARY KEY, btree (id)**

**"level\_unique" UNIQUE CONSTRAINT, btree (level)**

Wskazywany przez:

TABLE "account\_alevel" CONSTRAINT "access\_level\_fk" FOREIGN KEY (id\_alevel) REFERENCES access\_level(id)

### access\_level

```
CREATE TABLE public.access_level
(
    id bigint NOT NULL,
    level character varying(16) COLLATE pg_catalog."default" NOT NULL,
    active boolean NOT NULL DEFAULT true,
    version bigint NOT NULL,
    CONSTRAINT access_level_pk PRIMARY KEY (id),
    CONSTRAINT level_unique UNIQUE (level)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;
```

Tabela: account\_alevel

Kolumna	Typ	Nullowalne	Znaczenie	Domyślnie
id_account	bigint	not null	powiązanie z kontem w tabeli account	
id_alevel	bigint	not null	powiązanie z poziomem dostępu w tabeli access_level	
<b>id</b>	<b>bigint</b>	<b>not null</b>	<b>id pary konto - poziom dostępu</b>	
version	bigint	not null	numer wersji	

Indeksy:

**"account\_alevel\_pk" PRIMARY KEY, btree (id)**

"account\_alevel\_unique" UNIQUE CONSTRAINT, btree (id\_account, id\_alevel)

"account\_alevel\_id\_account" btree (id\_account)

"account\_alevel\_id\_alevel" btree (id\_alevel)

Ograniczenia kluczy obcych:

"access\_level\_fk" FOREIGN KEY (id\_alevel) REFERENCES access\_level(id)

"account\_fk" FOREIGN KEY (id\_account) REFERENCES account(id)

### account\_alevel

```
CREATE TABLE public.account_alevel
(
    id_account bigint NOT NULL,
    id_alevel bigint NOT NULL,
    id bigint NOT NULL,
    version bigint NOT NULL,
    CONSTRAINT account_alevel_pk PRIMARY KEY (id),
    CONSTRAINT account_alevel_unique UNIQUE (id_account, id_alevel),
    CONSTRAINT access_level_fk FOREIGN KEY (id_alevel)
        REFERENCES public.access_level (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT account_fk FOREIGN KEY (id_account)
        REFERENCES public.account (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;
```

Widok: glassfish\_auth\_view

Kolumna	Typ	Nullowalne	Znaczenie
login	character varying(32)	null	login użytkownika
password	character(64)	null	hasło użytkownika
level	character varying(16)	null	poziom dostępu

### glassfish\_auth\_view

```
CREATE OR REPLACE VIEW public.glassfish_auth_view AS
SELECT account.login,
       account.password,
       'virtual'::character varying(16) AS level
FROM account
     JOIN account_alevel ON account.id = account_alevel.id_account
     LEFT JOIN access_level ON account_alevel.id_alevel = access_level.id
     JOIN verification_token ON account.id = verification_token.id_account
WHERE account.confirm = false AND account.active = true AND access_level.
active = true AND verification_token.used = false
UNION
SELECT account.login,
       account.password,
       access_level.level
FROM account
     JOIN account_alevel ON account.id = account_alevel.id_account
     JOIN access_level ON account_alevel.id_alevel = access_level.id
WHERE account.confirm = true AND account.active = true AND access_level.
active = true;
```

Tabela: user\_data

Kolumna	Typ	Nullowalne	Znaczenie	Domyślnie
id	bigint	not null	id rekordu z danymi użytkownika	
name	character varying(32)	not null	imię użytkownika	
surname	character varying(32)	not null	nazwisko użytkownika	
email	character varying(64)	not null	adres email użytkownika	
phone	character varying(18)	not null	telefon użytkownika	
version	bigint	not null	numer wersji	
street	character varying(60)	not null	ulica	
street_number	character varying(10)	not null	numer ulicy	
flat_number	character varying(10)	null	numer mieszkania	
postal_code	character varying(10)	not null	kod pocztowy	
city	character varying(60)	not null	miasto	
country	character varying(60)	not null	kraj	

Indeksy:

"user\_data\_pk" PRIMARY KEY, btree (id)

"user\_data\_email\_unique" UNIQUE CONSTRAINT, btree (email)

"user\_data\_phone\_unique" UNIQUE CONSTRAINT, btree (phone)

Ograniczenia kluczy obcych:

"user\_data\_account\_fk" FOREIGN KEY (id) REFERENCES account(id)

### user\_data

```
CREATE TABLE public.user_data
(
    id bigint NOT NULL,
    name character varying(32) COLLATE pg_catalog."default" NOT NULL,
    surname character varying(32) COLLATE pg_catalog."default" NOT NULL,
    email character varying(64) COLLATE pg_catalog."default" NOT NULL,
    phone character varying(18) COLLATE pg_catalog."default" NOT NULL,
    version bigint NOT NULL,
    street character varying(60) COLLATE pg_catalog."default" NOT NULL,
    street_number character varying(10) COLLATE pg_catalog."default" NOT
NULL,
    flat_number character varying(10) COLLATE pg_catalog."default",
    postal_code character varying(10) COLLATE pg_catalog."default" NOT
NULL,
    city character varying(60) COLLATE pg_catalog."default" NOT NULL,
    country character varying(60) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT user_data_pk PRIMARY KEY (id),
    CONSTRAINT user_data_email_unique UNIQUE (email),
    CONSTRAINT user_data_phone_unique UNIQUE (phone),
    CONSTRAINT user_data_account_fk FOREIGN KEY (id)
        REFERENCES public.account (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;
```

Tabela: verification\_token

Kolumna	Typ	Nullowalne	Znaczenie	Domyślnie
token	character varying(32)	not null	ciąg znaków reprezentujących token weryfikacji konta	
expiry_date	timestamp without time zone	not null	data wygaśnięcia tokenu	
id_account	bigint	not null	powiązanie z kontem w tabeli account	
used	boolean	not null	czy token został wykorzystany	false
confirmation_date	timestamp without time zone	null	data potwierdzenia konta przy użyciu tokenu	
version	bigint	not null	numer wersji	

Indeksy:

"verification\_token\_pkey" PRIMARY KEY, btree (id\_account)

"token\_unique" UNIQUE CONSTRAINT, btree (token)

Ograniczenia kluczy obcych:

"verification\_token\_account\_fk" FOREIGN KEY (id\_account) REFERENCES account(id)



### verification\_token

```
CREATE TABLE public.verification_token
(
    token character varying(32) COLLATE pg_catalog."default" NOT NULL,
    expiry_date timestamp without time zone NOT NULL,
    id_account bigint NOT NULL,
    used boolean NOT NULL DEFAULT false,
    confirmation_date timestamp without time zone,
    version bigint NOT NULL,
    CONSTRAINT verification_token_pkey PRIMARY KEY (id_account),
    CONSTRAINT token_unique UNIQUE (token),
    CONSTRAINT verification_token_account_fk FOREIGN KEY (id_account)
        REFERENCES public.account (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;
```

Tabela: archival\_password

Kolumna	Typ	Nullowalne	Znaczenie	Domyślnie
id	bigint	not null	id rekordu przechowującego dane o ustalonym hasle konta	
password	character(64)	not null	hasło użytkownika	
id_account	bigint	not null	powiązanie z kontem w tabeli account	
setting_date	timestamp without time zone	not null	data ustawienia hasła	
version	bigint	not null	numer wersji	

Indeksy:

"archival\_password\_pk" PRIMARY KEY, btree (id)

"archival\_password\_id\_account" btree (id\_account)

Ograniczenia kluczy obcych:

"archival\_password\_id\_account\_fkey" FOREIGN KEY (id\_account) REFERENCES account(id)

### archival\_password

```
CREATE TABLE public.archival_password
(
    id bigint NOT NULL,
    password character(64) COLLATE pg_catalog."default" NOT NULL,
    id_account bigint NOT NULL,
    setting_date timestamp without time zone NOT NULL,
    version bigint NOT NULL,
    CONSTRAINT archival_password_pk PRIMARY KEY (id),
    CONSTRAINT archival_password_id_account_fkey FOREIGN KEY (id_account)
        REFERENCES public.account (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;
```

## Moduł Obsługi Produktów (MOP)

Tabela: product

Kolumna	Typ	Nullowalne	Znaczenie	Domyslnie
id	bigint	not null	id produktu	
category_id	bigint	not null	powiązanie z kategorią w tabeli category	
owner_id	bigint	not null	powiązanie z właścicielem w tabeli account	
name	character varying(32)	not null	nazwa produktu	
description	character varying(800)	not null	opis produktu	
unit_id	bigint	not null	powiązanie z jednostką miary w tabeli unit	
price	numeric(8,2)	not null	cena jednostkowa produktu	
qty	numeric(9,3)	not null	ilość produktu	
active	boolean	not null	czy produkt jest aktywny	true
version	bigint	not null	numer wersji	

Indeksy:

"product\_pk" PRIMARY KEY, btree (id)

"product\_category\_id" btree (category\_id)

"product\_owner\_id" btree (owner\_id)

"product\_unit\_id" btree (unit\_id)

Ograniczenie kontrolne:

"product\_price\_check" CHECK (price >= 0::numeric)

"product\_qty\_check" CHECK (qty >= 0::numeric)

Ograniczenia kluczy obcych:

"product\_category\_fk" FOREIGN KEY (category\_id) REFERENCES category(id)

"product\_owner\_fk" FOREIGN KEY (owner\_id) REFERENCES account(id)

"product\_unit\_fk" FOREIGN KEY (unit\_id) REFERENCES unit(id)

## product

```
CREATE TABLE public.product
(
    id bigint NOT NULL,
    category_id bigint NOT NULL,
    owner_id bigint NOT NULL,
    name character varying(32) COLLATE pg_catalog."default" NOT NULL,
    description character varying(800) COLLATE pg_catalog."default" NOT
NULL,
    unit_id bigint NOT NULL,
    price numeric(8,2) NOT NULL,
    qty numeric(9,3) NOT NULL,
    active boolean NOT NULL DEFAULT true,
    version bigint NOT NULL,
    CONSTRAINT product_pk PRIMARY KEY (id),
    CONSTRAINT product_category_fk FOREIGN KEY (category_id)
        REFERENCES public.category (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT product_owner_fk FOREIGN KEY (owner_id)
        REFERENCES public.account (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT product_unit_fk FOREIGN KEY (unit_id)
        REFERENCES public.unit (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT product_price_check CHECK (price >= 0::numeric),
    CONSTRAINT product_qty_check CHECK (qty >= 0::numeric)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;
```

Tabela: unit

Kolumna	Typ	Nullowalne	Znaczenie	Domyślnie
id	bigint	not null	id jednostki miary	
unit_name	character varying(16)	not null	nazwa jednostki miary	
version	bigint	not null	numer wersji	

Indeksy:

"unit\_pk" PRIMARY KEY, btree (id)

"unit\_name\_unique" UNIQUE CONSTRAINT, btree (unit\_name)

Wskazywany przez:

TABLE "product" CONSTRAINT "product\_unit\_fk" FOREIGN KEY (unit\_id) REFERENCES unit(id)

### unit

```
CREATE TABLE public.unit
(
    id bigint NOT NULL,
    unit_name character varying(16) COLLATE pg_catalog."default" NOT NULL,
    version bigint NOT NULL,
    CONSTRAINT unit_pk PRIMARY KEY (id),
    CONSTRAINT unit_name_unique UNIQUE (unit_name)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;
```

Tabela: category

Kolumna	Typ	Nullowalne	Znaczenie	Domyslnie
id	bigint	not null	id kategorii produktu	
category_name	character varying(16)	not null	nazwa kategorii produktu	
active	boolean	not null	czy kategoria jest aktywna	true
version	bigint	not null	numer wersji	

Indeksy:

**"category\_pk" PRIMARY KEY, btree (id)**

"category\_name\_unique" UNIQUE CONSTRAINT, btree (category\_name)

Wskazywany przez:

TABLE "product" CONSTRAINT "product\_category\_fk" FOREIGN KEY (category\_id) REFERENCES category(id)

### category

```
CREATE TABLE public.category
(
    id bigint NOT NULL,
    category_name character varying(16) COLLATE pg_catalog."default" NOT NULL,
    active boolean NOT NULL DEFAULT true,
    version bigint NOT NULL,
    CONSTRAINT category_pk PRIMARY KEY (id),
    CONSTRAINT category_name_unique UNIQUE (category_name)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;
```

Tabela: order

Kolumna	Typ	Nullowalne	Znaczenie	Domyślnie
id	bigint	not null	id zamówienia	
seller_id	bigint	not null	powiązanie z kontem sprzedawcy w tabeli account	
buyer_id	bigint	not null	powiązanie z kontem kupującego w tabeli account	
shipping_id	bigint	not null	powiązanie z dostawą w tabeli order_shipping	
order_placed_date	timestamp (without time zone)	not null	data złożenia zamówienia	
total_price	numeric(10,2)	not null	całkowita cena zamówienia	
status_id	bigint	not null	powiązanie ze statusem w tabeli order_status	
is_closed	boolean	not null	czy zamówienie zostało zamknięte	false
version	bigint	not null	numer wersji	

Indeksy:

"order\_pk" PRIMARY KEY, btree (id)

"order\_buyer\_id" btree (buyer\_id)

"order\_seller\_id" btree (seller\_id)

"order\_shipping\_id" btree (shipping\_id)

"order\_status\_id" btree (status\_id)

Ograniczenie kontrolne:

"order\_total\_price\_check" CHECK (total\_price >= 0::numeric)

Ograniczenia kluczy obcych:

"order\_buyer\_user\_fk" FOREIGN KEY (buyer\_id) REFERENCES account(id)

"order\_seller\_user\_fk" FOREIGN KEY (seller\_id) REFERENCES account(id)

"order\_shipping\_fk" FOREIGN KEY (shipping\_id) REFERENCES order\_shipping(id)

"order\_status\_fk" FOREIGN KEY (status\_id) REFERENCES order\_status(id)

Wskazywany przez:

TABLE "order\_products" CONSTRAINT "order\_products\_order\_fk" FOREIGN KEY (order\_id) REFERENCES "order"(id)

## order

```
CREATE TABLE public."order"
(
    id bigint NOT NULL,
    seller_id bigint NOT NULL,
    buyer_id bigint NOT NULL,
    shipping_id bigint NOT NULL,
    order_placed_date timestamp without time zone NOT NULL,
    total_price numeric(10,2) NOT NULL,
    status_id bigint NOT NULL,
    is_closed boolean NOT NULL DEFAULT false,
    version bigint NOT NULL,
    CONSTRAINT order_pk PRIMARY KEY (id),
    CONSTRAINT order_buyer_user_fk FOREIGN KEY (buyer_id)
        REFERENCES public.account (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT order_seller_user_fk FOREIGN KEY (seller_id)
        REFERENCES public.account (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT order_shipping_fk FOREIGN KEY (shipping_id)
        REFERENCES public.order_shipping (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT order_status_fk FOREIGN KEY (status_id)
        REFERENCES public.order_status (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT order_total_price_check CHECK (total_price >= 0::numeric)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;
```

Tabela: order\_products

Kolumna	Typ	Nullowalne	Znaczenie	Domyślnie
<b>id</b>	<b>bigint</b>	<b>not null</b>	<b>id pary zamówienie - produkt</b>	
order_id	bigint	not null	id zamówienia	
product_name	character varying(32)	not null	nazwa produktu	
product_qty	numeric(9,3)	not null	ilość produktu	
product_unit_name	character varying(16)	not null	jednostka miary, w jakiej podana jest ilość produktu	
product_value	numeric(9,2)	not null	cena, po której kupiono produkt	
version	bigint	not null	numer wersji	

Indeksy:  
"order\_products\_pk" PRIMARY KEY, btree (id)  
"order\_products\_unique" UNIQUE CONSTRAINT, btree (order\_id, product\_name)  
"order\_products\_order\_id" btree (order\_id)  
Ograniczenie kontrolne:  
"order\_products\_product\_qty\_check" CHECK (product\_qty > 0::numeric)  
"order\_products\_product\_value\_check" CHECK (product\_value >= 0::numeric)  
Ograniczenia kluczy obcych:  
"order\_products\_order\_fk" FOREIGN KEY (order\_id) REFERENCES "order"(id)

order\_products

```
CREATE TABLE public.order_products
(
    id bigint NOT NULL,
    order_id bigint NOT NULL,
    product_name character varying(32) COLLATE pg_catalog."default" NOT
NULL,
    product_qty numeric(9,3) NOT NULL,
    product_unit_name character varying(16) COLLATE pg_catalog."default"
NOT NULL,
    product_value numeric(9,2) NOT NULL,
    version bigint NOT NULL,
    CONSTRAINT order_products_pk PRIMARY KEY (id),
    CONSTRAINT order_products_unique UNIQUE (order_id, product_name),
    CONSTRAINT order_products_order_fk FOREIGN KEY (order_id)
REFERENCES public."order" (id) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION,
    CONSTRAINT order_products_product_value_check CHECK (product_value >=
0::numeric),
    CONSTRAINT order_products_product_qty_check CHECK (product_qty > 0::
numeric)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;
```

Tabela: order\_shipping

Kolumna	Typ	Nullowalne	Znaczenie	Domyślnie
id	bigint	not null	id rekordu z adresem przypisanym do zamówienia	
shipping_method_name	character varying(16)	not null	nazwa metody dostawy	
name	character varying(32)	not null	imię użytkownika	
surname	character varying(32)	not null	nazwisko użytkownika	
street	character varying(60)	not null	ulica	
street_number	character varying(10)	not null	numer ulicy	
flat_number	character varying(10)	null	numer mieszkania	
postal_code	character varying(10)	not null	kod pocztowy	
city	character varying(60)	not null	miasto	

country	character varying(60)	not null	kraj	
version	bigint	not null	numer wersji	

Indeksy:

**"order\_shipping\_pkey" PRIMARY KEY, btree (id)**

Wskazywany przez:

TABLE ""order"" CONSTRAINT "order\_shipping\_fk" FOREIGN KEY (shipping\_id) REFERENCES order\_shipping(id)

```
CREATE TABLE public.order_shipping
(
    id bigint NOT NULL,
    shipping_method_name character varying(16) COLLATE pg_catalog."default" NOT NULL,
    name character varying(32) COLLATE pg_catalog."default" NOT NULL,
    surname character varying(32) COLLATE pg_catalog."default" NOT NULL,
    street character varying(60) COLLATE pg_catalog."default" NOT NULL,
    street_number character varying(10) COLLATE pg_catalog."default" NOT NULL,
    flat_number character varying(10) COLLATE pg_catalog."default",
    postal_code character varying(10) COLLATE pg_catalog."default" NOT NULL,
    city character varying(60) COLLATE pg_catalog."default" NOT NULL,
    country character varying(60) COLLATE pg_catalog."default" NOT NULL,
    version bigint NOT NULL,
    CONSTRAINT order_shipping_pkey PRIMARY KEY (id)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;
```

## Tabela: order\_status

Kolumna	Typ	Nullowalne	Znaczenie	Domyślnie
id	bigint	not null	id statusu zamówienia	
name	character varying(16)	not null	nazwa statusu zamówienia	
version	bigint	not null	numer wersji	

Indeksy:

**"order\_status\_pk" PRIMARY KEY, btree (id)**

"order\_status\_unique" UNIQUE CONSTRAINT, btree (name)

Wskazywany przez:

TABLE ""order"" CONSTRAINT "order\_status\_fk" FOREIGN KEY (status\_id) REFERENCES order\_status(id)

### order\_status

```
CREATE TABLE public.order_status
(
    id bigint NOT NULL,
    name character varying(16) COLLATE pg_catalog."default" NOT NULL,
    version bigint NOT NULL,
```



```

        CONSTRAINT order_status_pk PRIMARY KEY (id),
        CONSTRAINT order_status_unique UNIQUE (name)
    )
    WITH (
        OIDS = FALSE
    )
    TABLESPACE pg_default;

```

**Tabela: shipping\_method**

Kolumna	Typ	Nullowalne	Znaczenie	Domyślnie
id	bigint	not null	id metody dostawy	
name	character varying(16)	not null	nazwa metody dostawy	
price	numeric(5,2)	not null	cena dostawy	
active	boolean	not null	czy metoda dostawy jest aktywna	true
version	bigint	not null	numer wersji	
created_by	bigint	not null	powiązanie z kontem autora metody wysyłki w tabeli account	

Indeksy:

**"shipping\_method\_pk" PRIMARY KEY, btree (id)**

"shipping\_method\_unique" UNIQUE CONSTRAINT, btree (name)

Ograniczenie kontrolne:

"shipping\_method\_price\_check" CHECK (price >= 0::numeric)

Ograniczenia kluczy obcych:

"account\_fk" FOREIGN KEY (created\_by) REFERENCES account(id)

#### shipping\_method

```

CREATE TABLE public.shipping_method
(
    id bigint NOT NULL,
    name character varying(16) COLLATE pg_catalog."default" NOT NULL,
    price numeric(5,2) NOT NULL,
    active boolean NOT NULL DEFAULT true,
    version bigint NOT NULL,
    created_by bigint,
    CONSTRAINT shipping_method_pk PRIMARY KEY (id),
    CONSTRAINT shipping_method_unique UNIQUE (name),
    CONSTRAINT account_fk FOREIGN KEY (created_by)
        REFERENCES public.account (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT shipping_method_price_check CHECK (price >= 0::numeric)
)
WITH (
    OIDS = FALSE

```

```
)
TABLESPACE pg_default;
```

## Dane inicjujące

Użytkownik w poziomie dostępu umożliwiającym tworzenie następnych kont (tabela account)

id	login	password	confirm	active	number_of_products	number_of_orders	number_of_logins	version
1	admin	9962271b738bc75844d698b931985fe829ca31e1ba79d5d447c1cce531d1c3fa	true	true	0	0	0	0
2	client	cca5e224cdaf36fd133708dab77a0c2aa33d44fd269aaef80d3e303715e7861	true	true	0	0	0	0
3	client2	d439bed4cdbfcd56d05f34d3978c397e2244f0a5c3094cefe66525f5acb269d5	true	false	0	0	0	0
4	manager	e22790af158f8039bcb00f9bd29804fca5d7a39fc93822104f8c4ca1c3df209a	true	true	0	0	0	0
5	manager2	b6b3fe424ed33eca b6b76cce9b0404a c9174d913086672 bc1035d46c8aa72bd	false	true	0	0	0	0

### Wycinek kopii danych bazy: użytkownik w poziomie dostępu umożliwiającym tworzenie innych kont

```
SELECT id, login, password, confirm, active, user_data_id,
number_of_products, number_of_orders, number_of_logins, version
FROM public.account;
```

Dane powyższych użytkowników (tabela user\_data)

id	name	surname	email	phone	version	street	street_number	flat_number	postal_code	city	country
1	imie1	nazwisko1	email1@p.pl	telefon1	0	ulica1	1		00-001	miasto1	kraj1
2	imie2	nazwisko2	email2@p.pl	telefon2	0	ulica2	2		00-002	miasto2	kraj2
3	imie3	nazwisko3	email3@p.pl	telefon3	0	ulica3	3		00-003	miasto3	kraj3
4	imie4	nazwisko4	email4@p.pl	telefon4	0	ulica4	4		00-004	miasto4	kraj4
5	imie5	nazwisko5	email5@p.pl	telefon5	0	ulica5	5		00-005	miasto5	kraj5

### Wycinek kopii danych bazy: użytkownik w poziomie dostępu umożliwiającym tworzenie innych kont

```
SELECT * FROM user_data;
```

Poziomy dostępów (tabela access\_level)

id	level	active	version
2	admin	t	0
3	level	t	0

4	manager	t	0
---	---------	---	---

### Wycinek kopii danych bazy: użytkownik w poziomie dostępu umożliwiającym tworzenie innych kont

```
SELECT * FROM access_level;
```

### Połączenie użytkowników z poziomami dostępu (tabela account\_alevel)

id_account	id_alevel	id	version
1	3	2	0
1	2	1	0
2	4	4	0
1	4	3	0
4	3	6	0
3	4	5	0
5	3	8	0
4	4	7	0
5	4	9	0

## Sekwencje

Dla większości tabel w bazie danych została utworzona sekwencja o schemacie nazw *nazwa\_tabeli\_id\_seq*, np. dla tabeli account odpowiadająca jej sekwencja nosi nazwę *account\_id\_seq*.

### Sekwencja: access\_level\_id\_seq

```
CREATE SEQUENCE public.access_level_id_seq
    INCREMENT 1
    START 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    CACHE 1;
```

### Sekwencja: account\_alevel\_id\_seq

```
CREATE SEQUENCE public.account_alevel_id_seq
    INCREMENT 1
    START 1
    MINVALUE 1
    MAXVALUE 9223372036854775807
    CACHE 1;
```

### Sekwencja: account\_id\_seq

```
CREATE SEQUENCE public.account_id_seq
    INCREMENT 1
```

```
START 5  
MINVALUE 1  
MAXVALUE 9223372036854775807  
CACHE 1;
```

**Sekwencja: archival\_password\_id\_seq**

```
CREATE SEQUENCE public.archival_password_id_seq  
INCREMENT 1  
START 57  
MINVALUE 1  
MAXVALUE 9223372036854775807  
CACHE 1;
```

**Sekwencja: category\_id\_seq**

```
CREATE SEQUENCE public.category_id_seq  
INCREMENT 1  
START 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
CACHE 1;
```

**Sekwencja: order\_id\_seq**

```
CREATE SEQUENCE public.order_id_seq  
INCREMENT 1  
START 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
CACHE 1;
```

**Sekwencja: order\_products\_id\_seq**

```
CREATE SEQUENCE public.order_products_id_seq  
INCREMENT 1  
START 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
CACHE 1;
```

**Sekwencja: order\_shipping\_id\_seq**

```
CREATE SEQUENCE public.order_shipping_id_seq
  INCREMENT 1
  START 1
  MINVALUE 1
  MAXVALUE 9223372036854775807
  CACHE 1;
```

#### Sekwencja: product\_id\_seq

```
CREATE SEQUENCE public.product_id_seq
  INCREMENT 1
  START 1
  MINVALUE 1
  MAXVALUE 9223372036854775807
  CACHE 1;
```

#### Sekwencja: shipping\_method\_id\_seq

```
CREATE SEQUENCE public.shipping_method_id_seq
  INCREMENT 1
  START 1
  MINVALUE 1
  MAXVALUE 9223372036854775807
  CACHE 1;
```

#### Sekwencja: user\_data\_id\_seq

```
CREATE SEQUENCE public.user_data_id_seq
  INCREMENT 1
  START 1
  MINVALUE 1
  MAXVALUE 9223372036854775807
  CACHE 1;
```

## 4 Użytkownicy bazodanowi 2 pkt

### Lista użytkowników bazodanowych

Nazwa użytkownika	Odpowiedzialność
ssbd01glassfish	Logowanie użytkowników
ssbd01	Konto administracyjne
ssbd01admin	Konto administracyjne
ssbd01mok	Obsługa mok
ssbd01mop	Obsługa mop

ssbd01moz	Obsługa moz
-----------	-------------

**Tabela uprawnień użytkowników bazodanowych**

Nazwa użytkownika	Uprawnienia
ssbd01glassfish	glassfish_auth_view = r
ssbd01	allTables = arwdDxt, allSequence = rwU
ssbd01admin	allTables = arwdDxt, allSequence = rwU
ssbd01mok	access_level = rwa, access_level_id_seq = rU, account = rwa, account_id_seq = rU, account_alevel = rwad, account_alevel_id_seq = rU, archival_password = ra, archival_password_id_seq = rU, user_data = rwa, user_data_id_seq = rU, verification_token = rwa
ssbd01mop	product = rwad, category = rwa, unit = r, account = rw, account_alevel = r, access_level = r, category_id_seq = rU, product_id_seq = rU
ssbd01moz	order = rwa, order_products = rwa, order_shipping = rwa, order_status = r, shipping_method = rwa, account = rw, account_alevel = r, access_level = r, order_id_seq = rU, order_products_id_seq = rU, order_shipping_id_seq = rU, shipping_method_id_seq = rU, product = rw, category = r, unit = r

Uprawnienia zostały zapisane w formacie: tabela = listaUprawnie

**Legenda:**

1. allTables - wszystkie tabele (gdy dla danego konta zostały wymienione szczegółowe uprawnienia do tabel, mają one pierwszeństwo nad oznaczeniem allTables)
2. allSequence - wszystkie sekwencje
3. all - brak ograniczeń uprawnień
4. r - uprawnienie do dokonywania SELECT
5. w - uprawnienie do dokonywania UPDATE
6. a - uprawnienie do dokonywania INSERT
7. d - uprawnienie do dokonywania DELETE
8. D - uprawnienie do dokonywania TRUNCATE
9. x - uprawnienie do dokonywania REFERENCES

10. t - uprawnienie do dokonywania TRIGGER
11. U - uprawnienie do używania (USAGE )

## 5 Identyfikacja obiektow encji 2 pkt

Zestawienie klas encji

Tabela1. **MOK.** Mapowanie encji na tabele.

Klasa encyjna	Tabela	Klucz główny	Typ klucza głównego
AccessLevel	access_level	id	bigint
Account	account	id	bigint
	user_data	id	bigint
	verification_token	id	bigint
AccountAlevel	account_alevel	id	bigint
ArchivalPassword	archival_password	id	bigint

Tabela2. **MOP.** Mapowanie encji na tabele.

Klasa encyjna	Tabela	Klucz główny	Typ klucza głównego
Category	category	id	bigint
Product	product	id	bigint
Unit	unit	id	bigint

Tabela3. **MOZ.** Mapowanie encji na tabele.

Klasa encyjna	Tabela	Klucz główny	Typ klucza głównego
Order1	order	id	bigint
OrderProducts	order_products	id	bigint
OrderShipping	order_shipping	id	bigint
OrderStatus	order_status	id	bigint
ShippingMethod	shipping_method	id	bigint

Związki między obiektami klas encji

Tabela4. Tabela krzyżowa związków encji.

		MOK				MOP			MOZ				
		AccessLevel	Account	AccountAlevel	ArchivalPassword	Category	Product	Unit	Order1	OrderProducts	OrderShipping	OrderStatus	ShippingMethod
MOK	AccessLevel			Accou ntAlev el	2								
				ALL	t ak								
	Account			Accou ntAlev el	2		Acco unt	2		Acco unt	2		Shippi ngMet hod
							-	tak		-	tak		1

																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					</
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	----

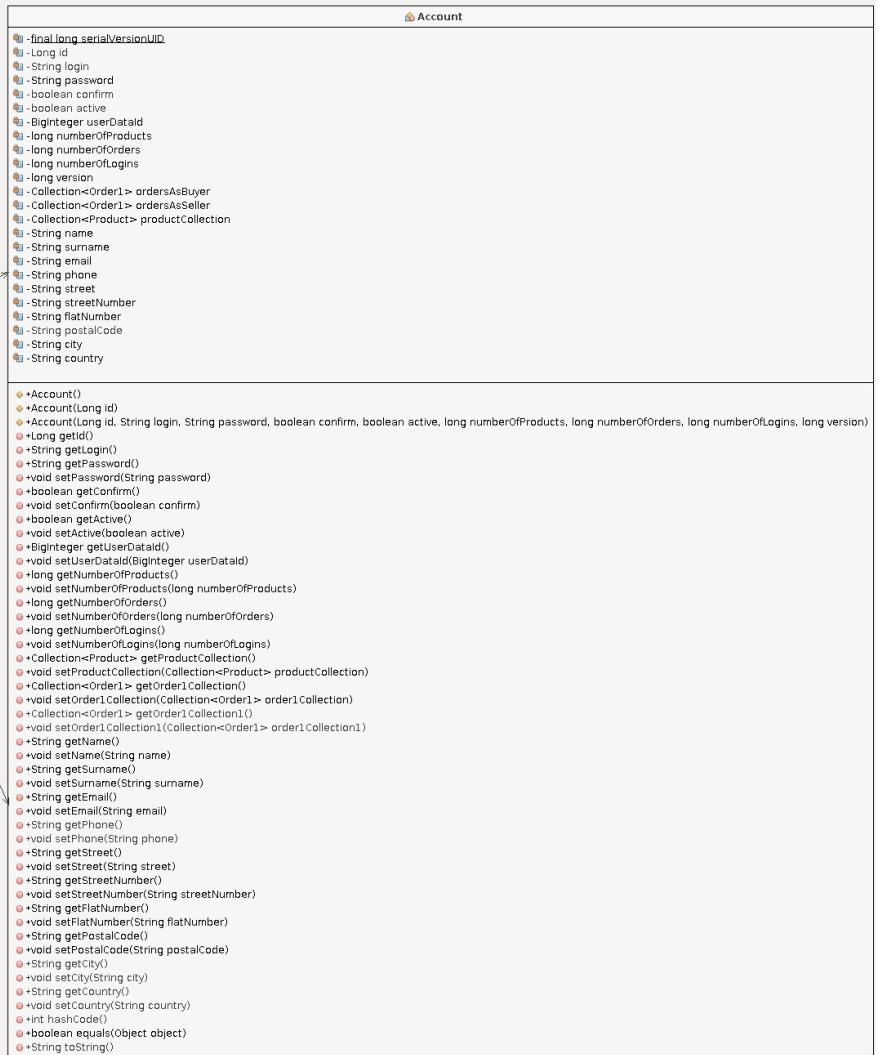
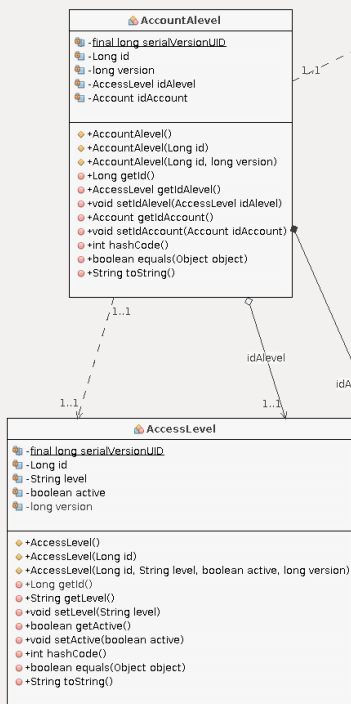
Tabela5. Legenda do Tabeli 4.

właściciel związku	kierunkowość związku(1-jednokierunkowy, 2-dwukierunkowy)
operacje kaskadowe w związku	mutowalność związku

6 Diagram klas encji 2 pkt

Diagram klas encji MOK:





## Diagram klas encji MOP:

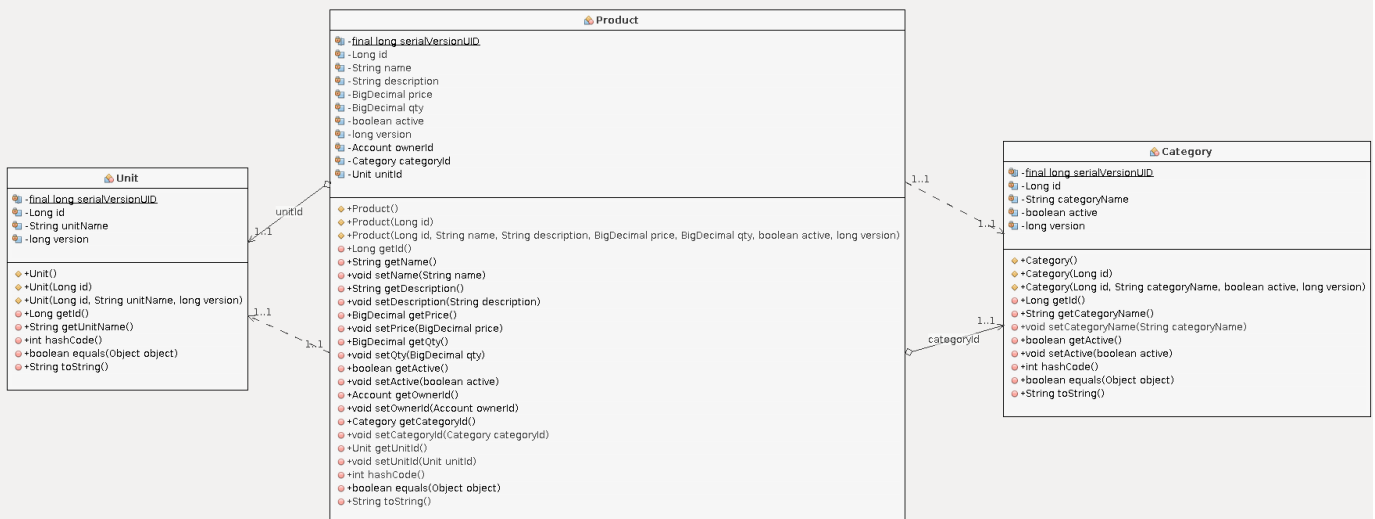
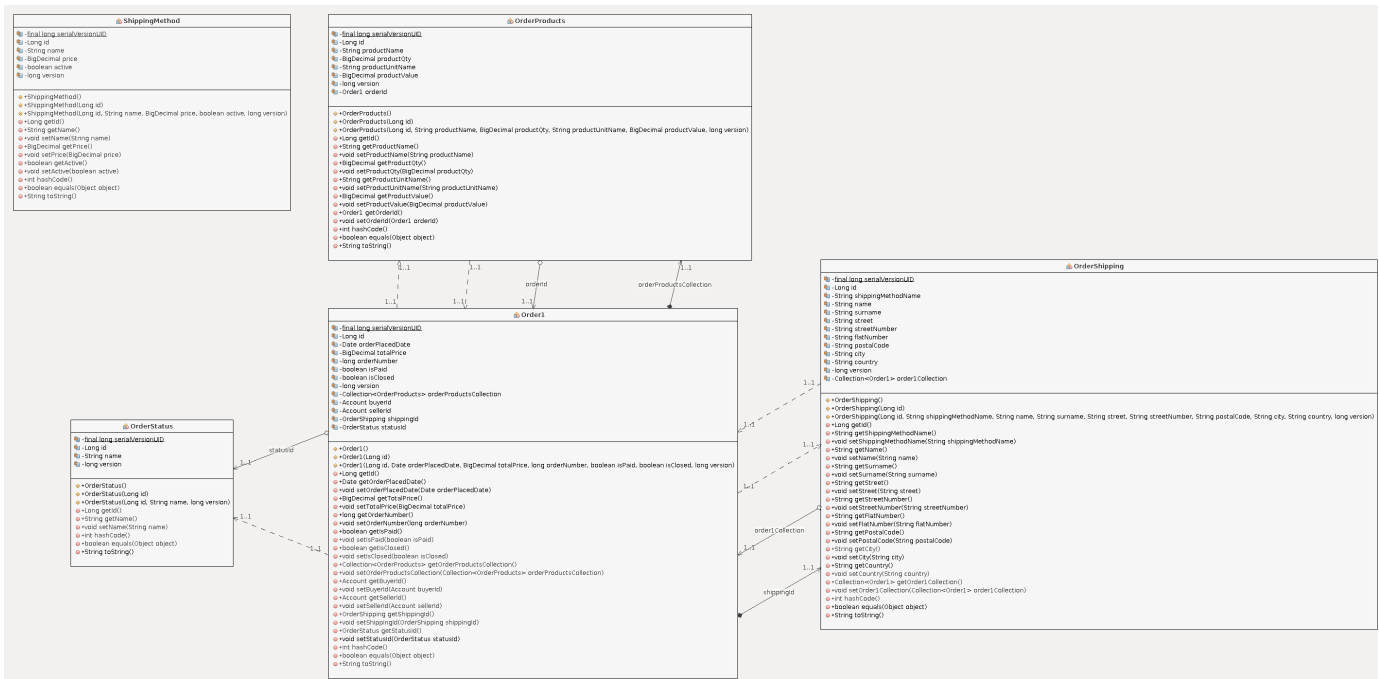


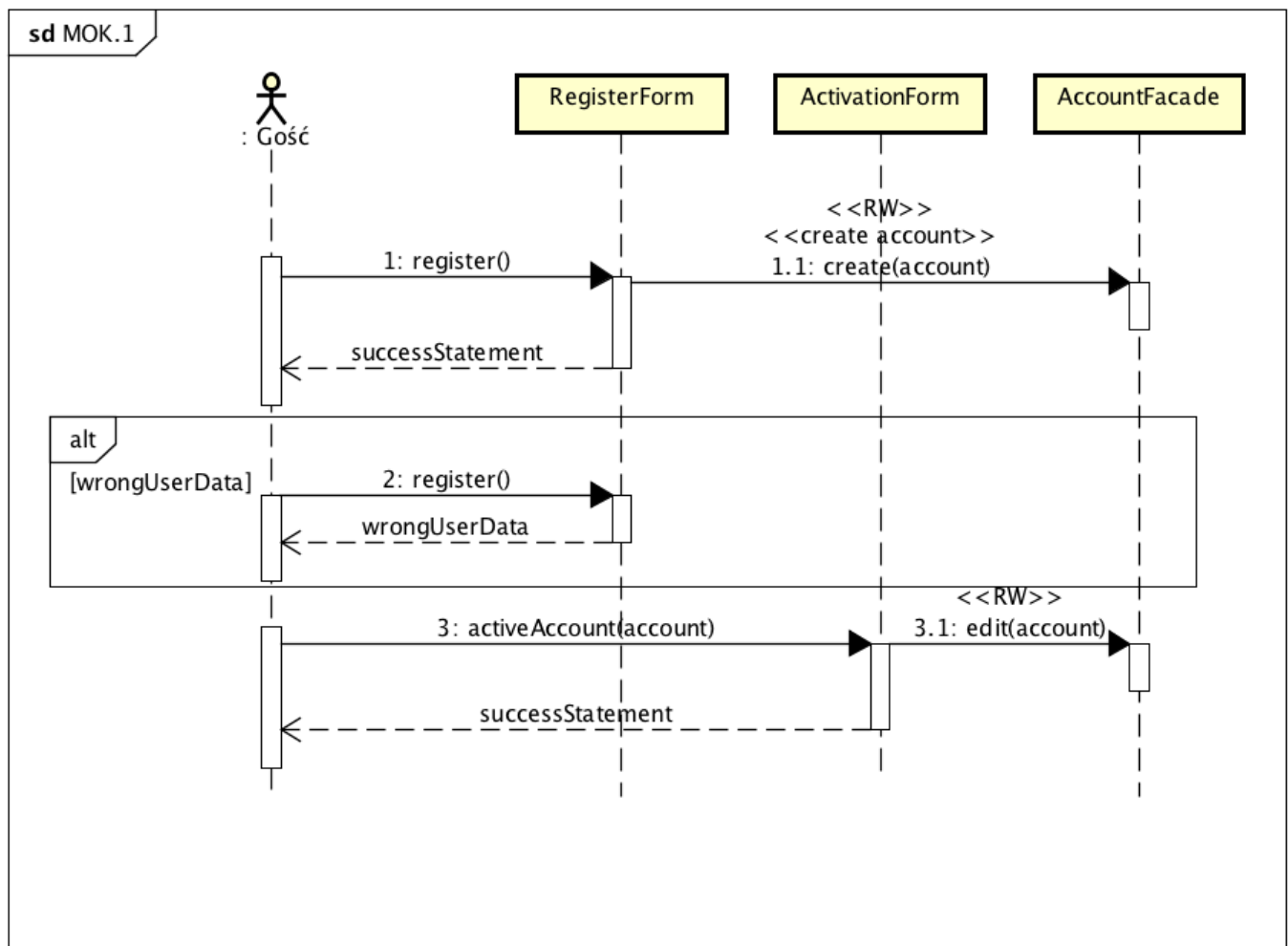
Diagram klas encji MOZ:



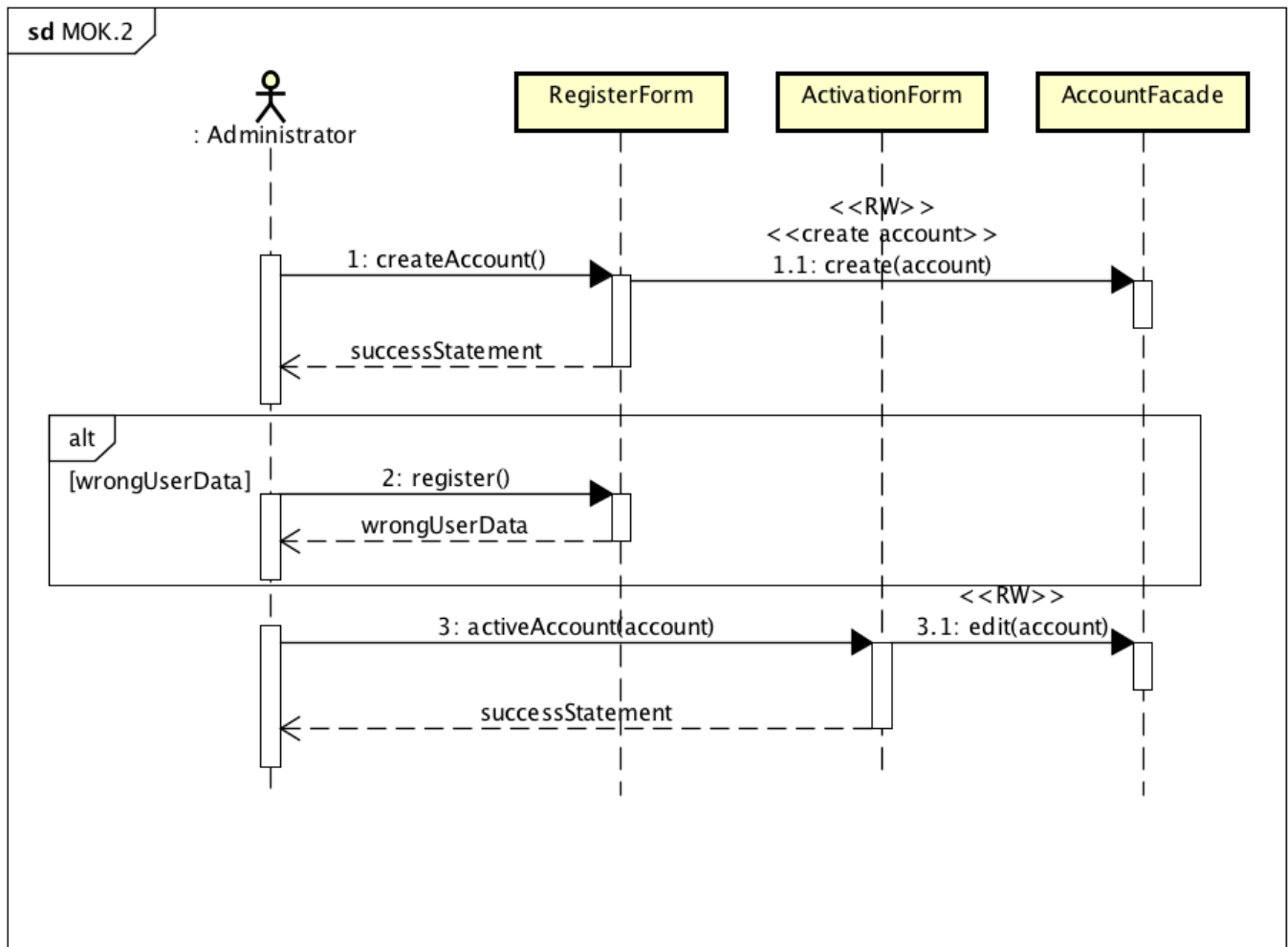
7 Diagramy sekwencji oraz identyfikacja transakcji bazodanowych 4 pkt

Diagramy MOK

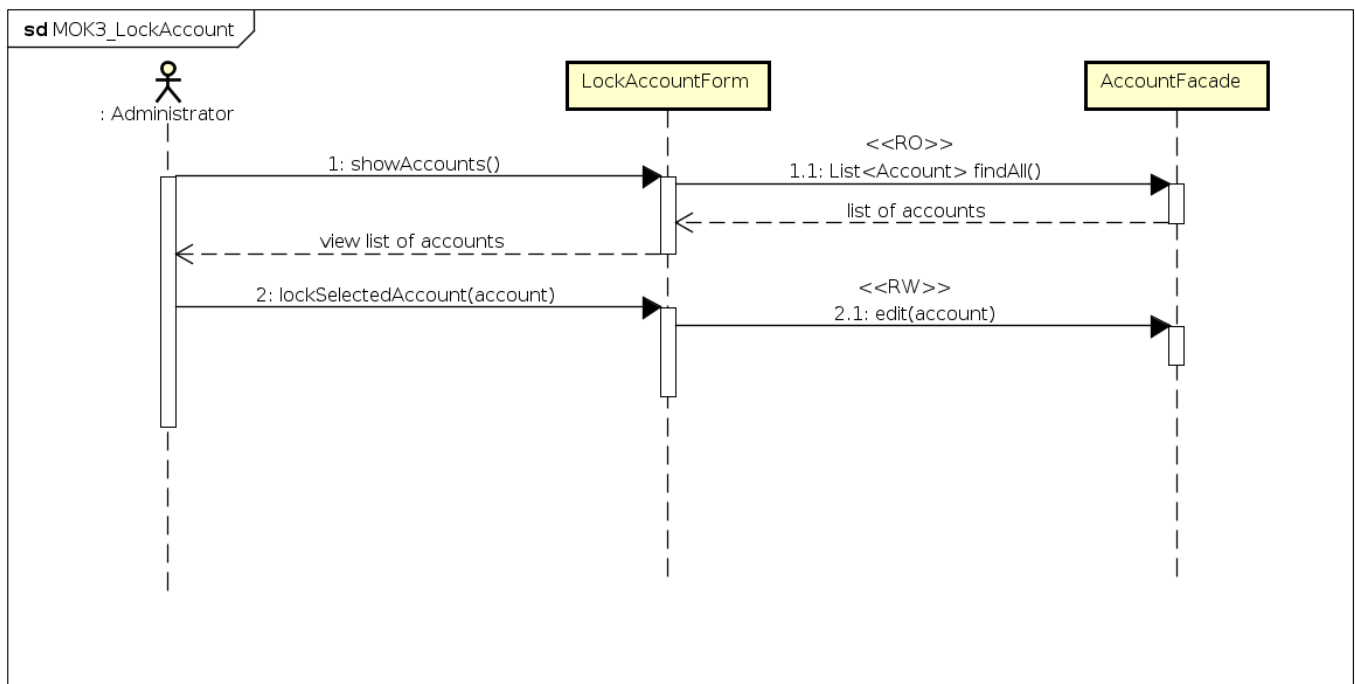
MOK1 - Zarejestruj (Michał Malec)



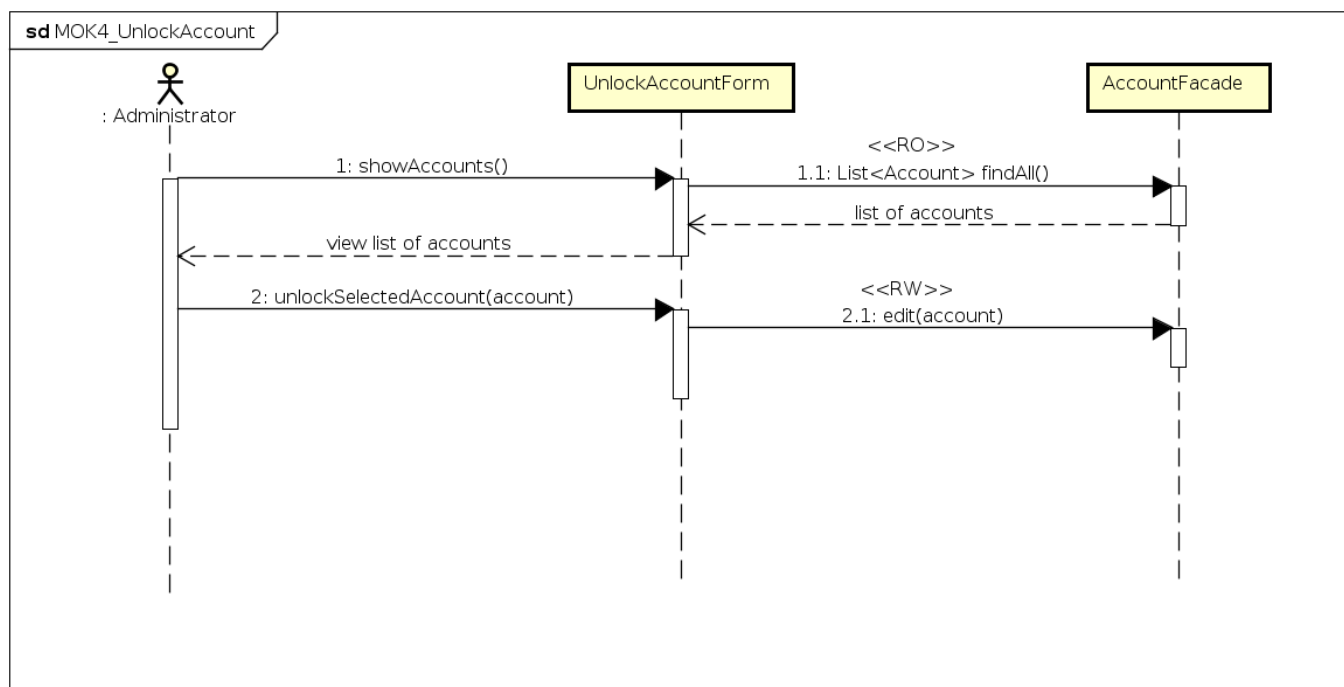
MOK2 - Utwórz konto (Michał Malec)



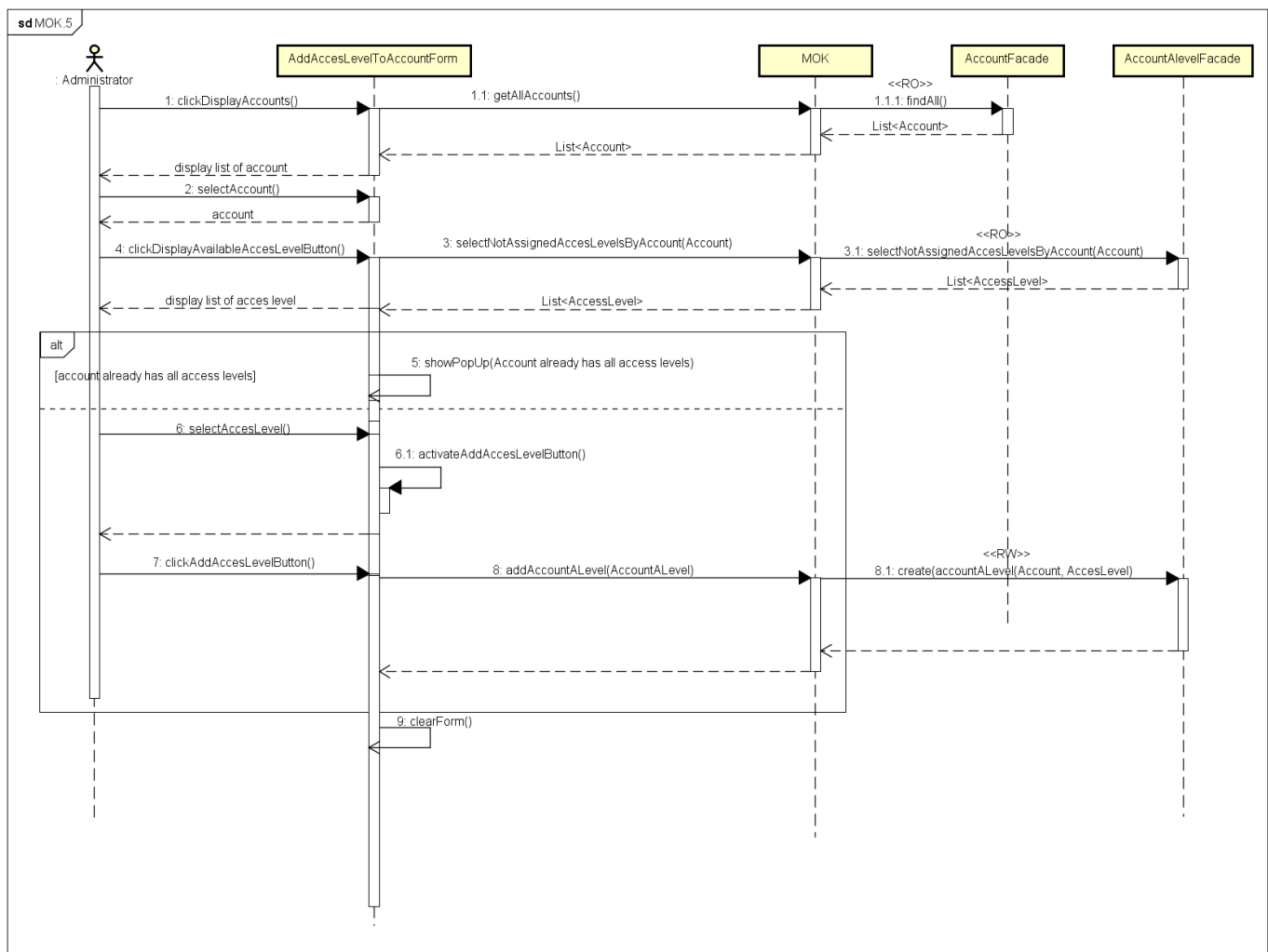
### MOK3 - Zablokuj konto (Piotr Biernacki)



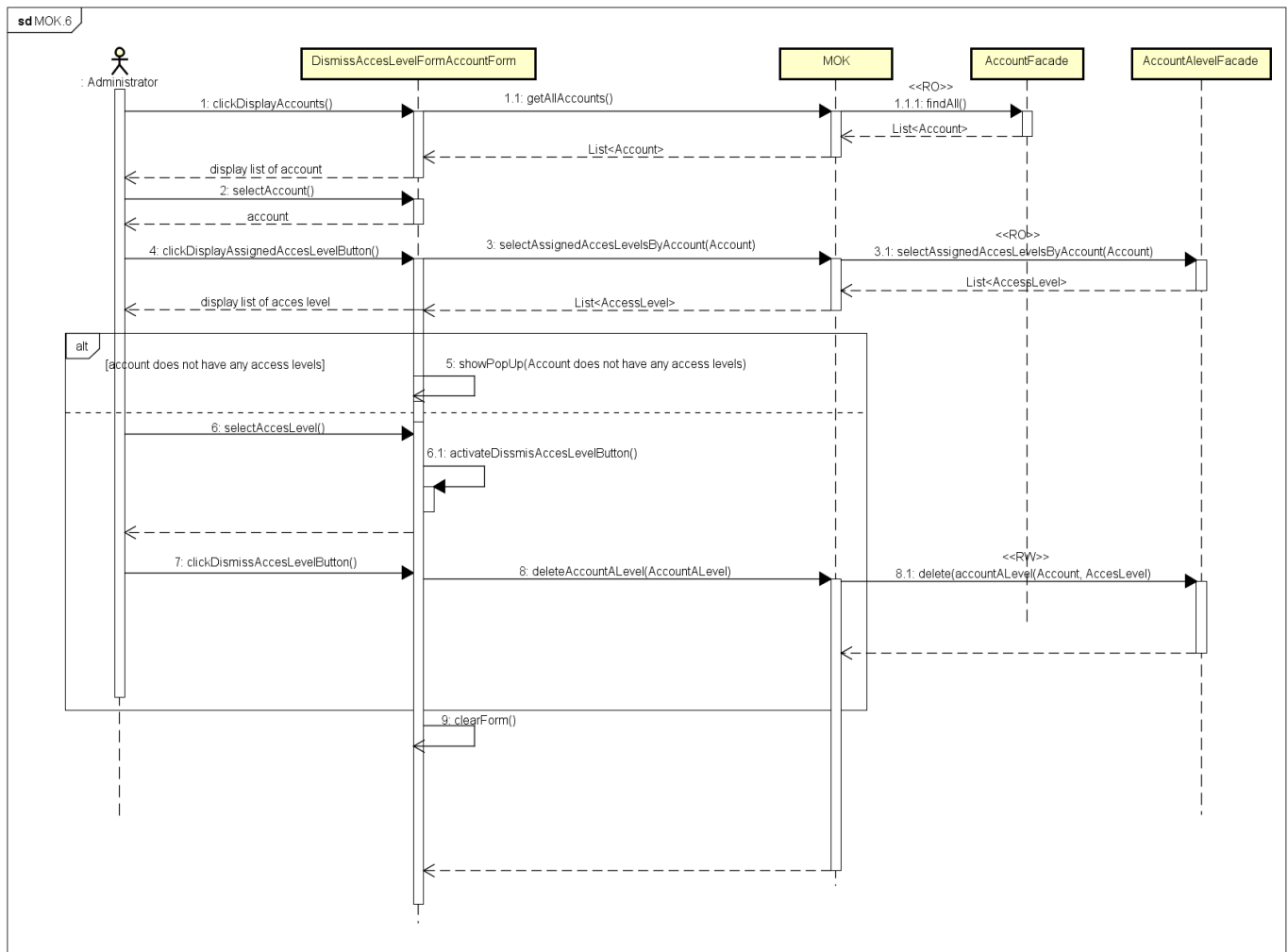
#### MOK4 - Odblokuj konto (Piotr Biernacki)



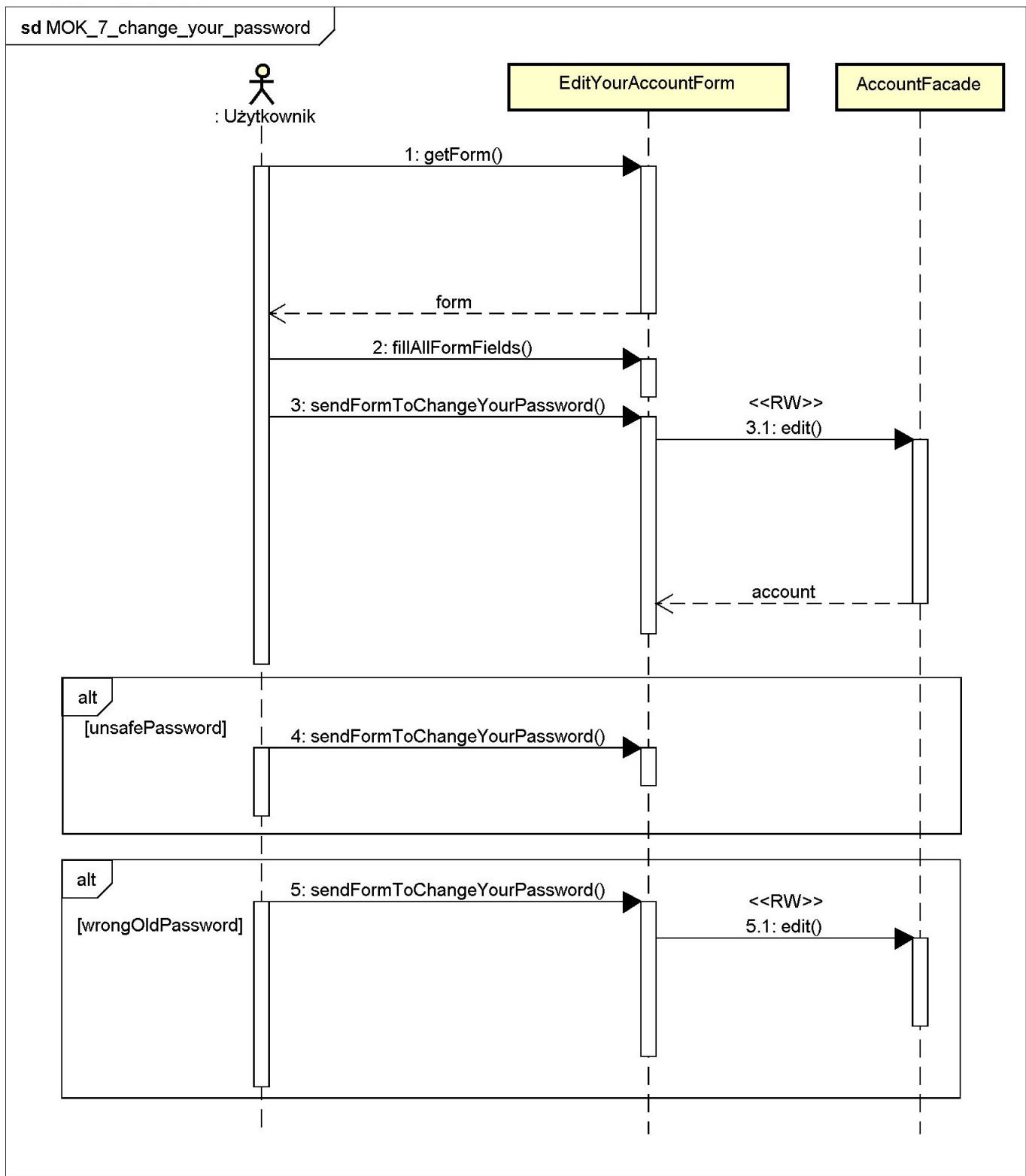
#### MOK5 - Dołącz poziom dostępu do konta (Filip Różański)



**MOK6 - Odłącz poziom dostępu od konta (Filip Różański)**

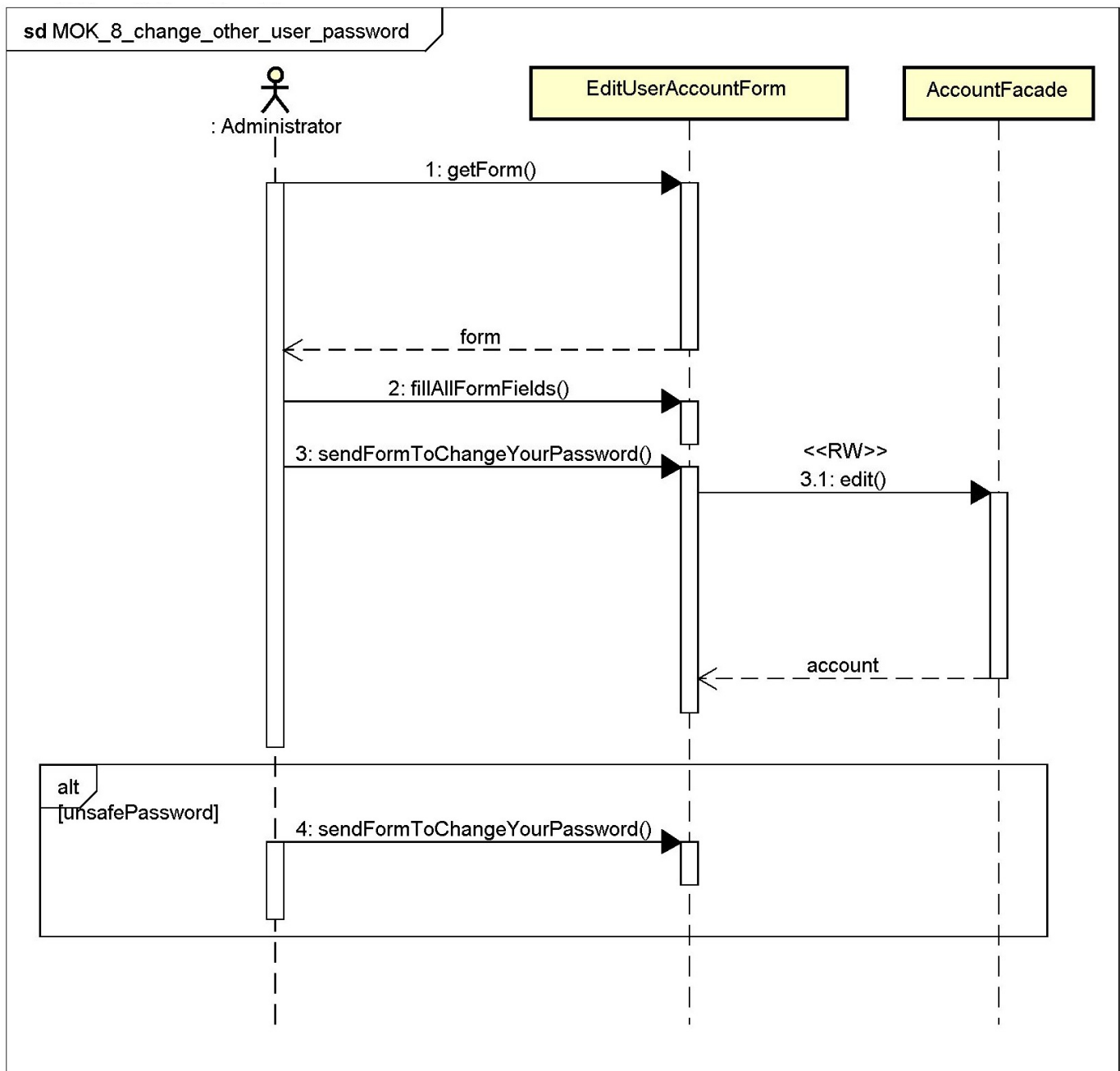


## MOK7 - Zmień własne hasło (Agata Kania)

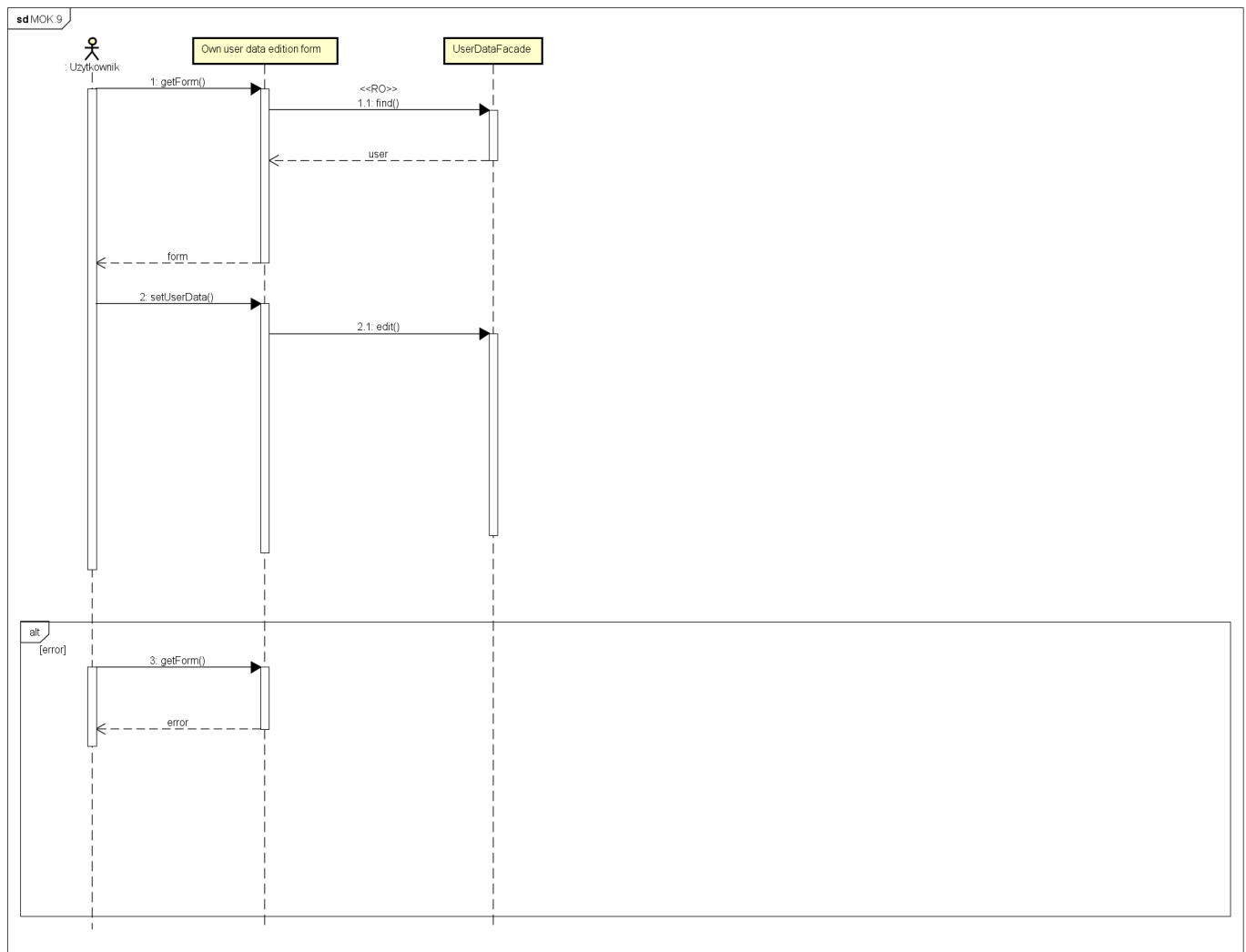


MOK8 - Zmień hasło innego użytkownika (Agata Kania)

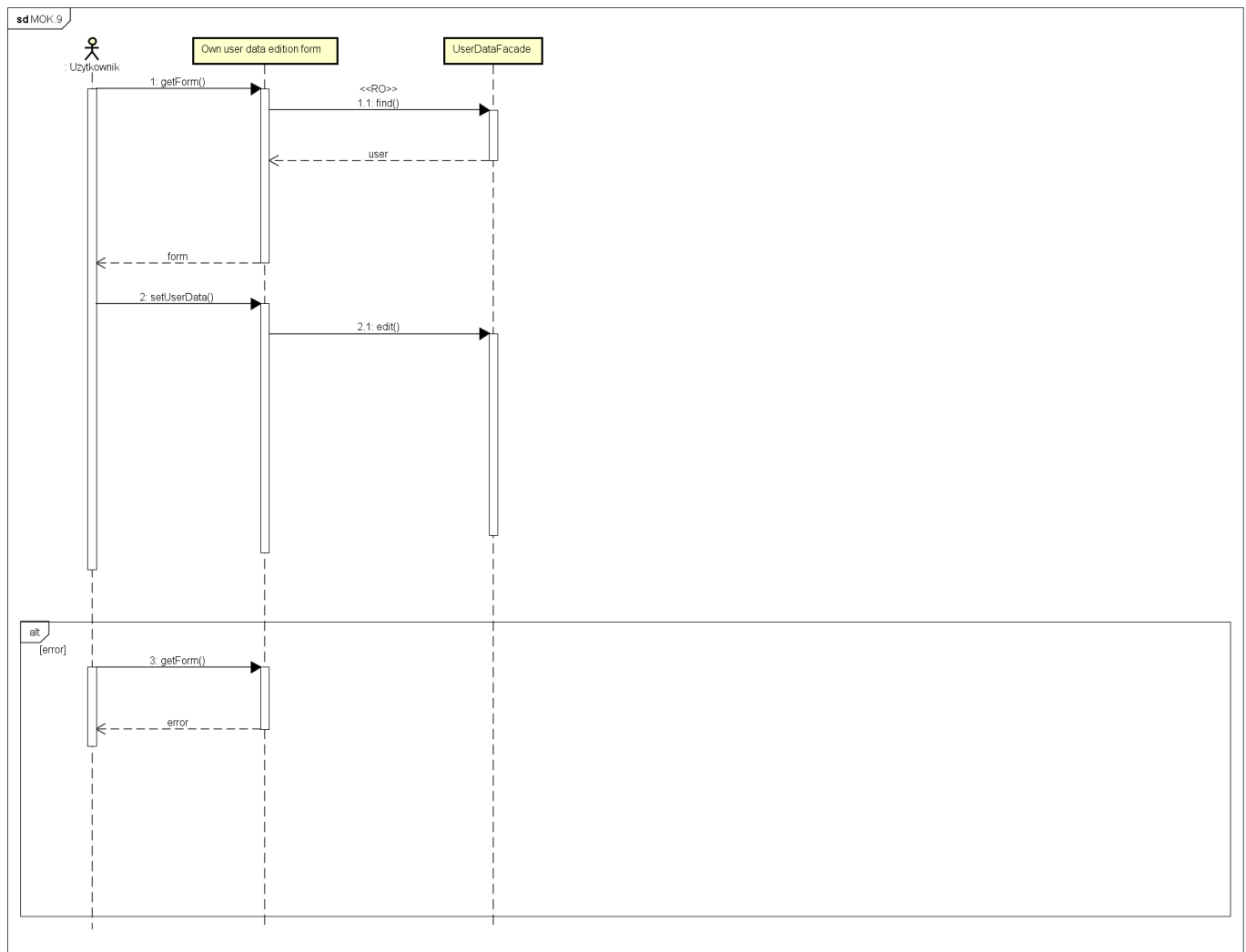




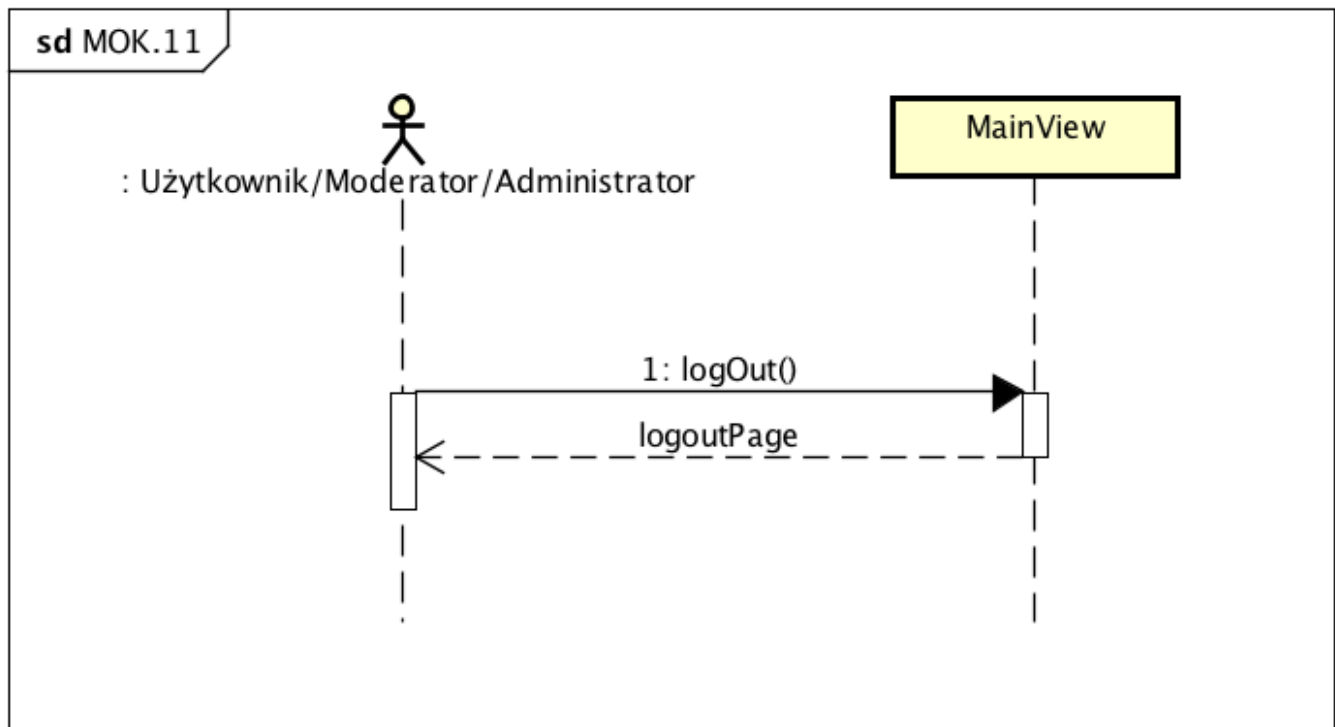
MOK9 - Edytuj dane własnego konta (Dominik Lange)



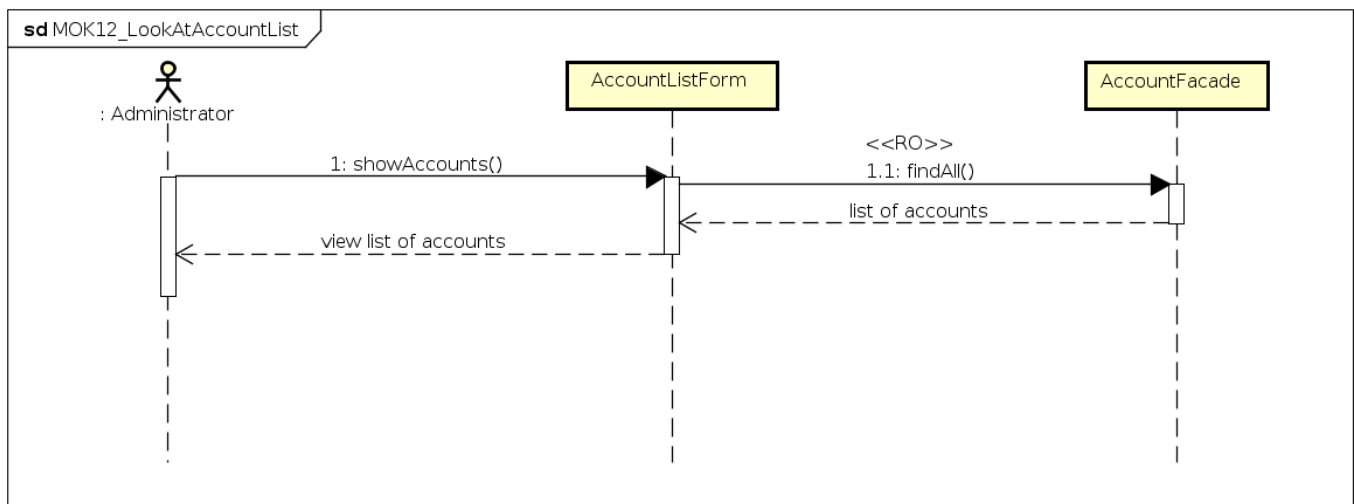
**MOK10 - Edytuj dane konta innego użytkownika (Dominik Lange)**



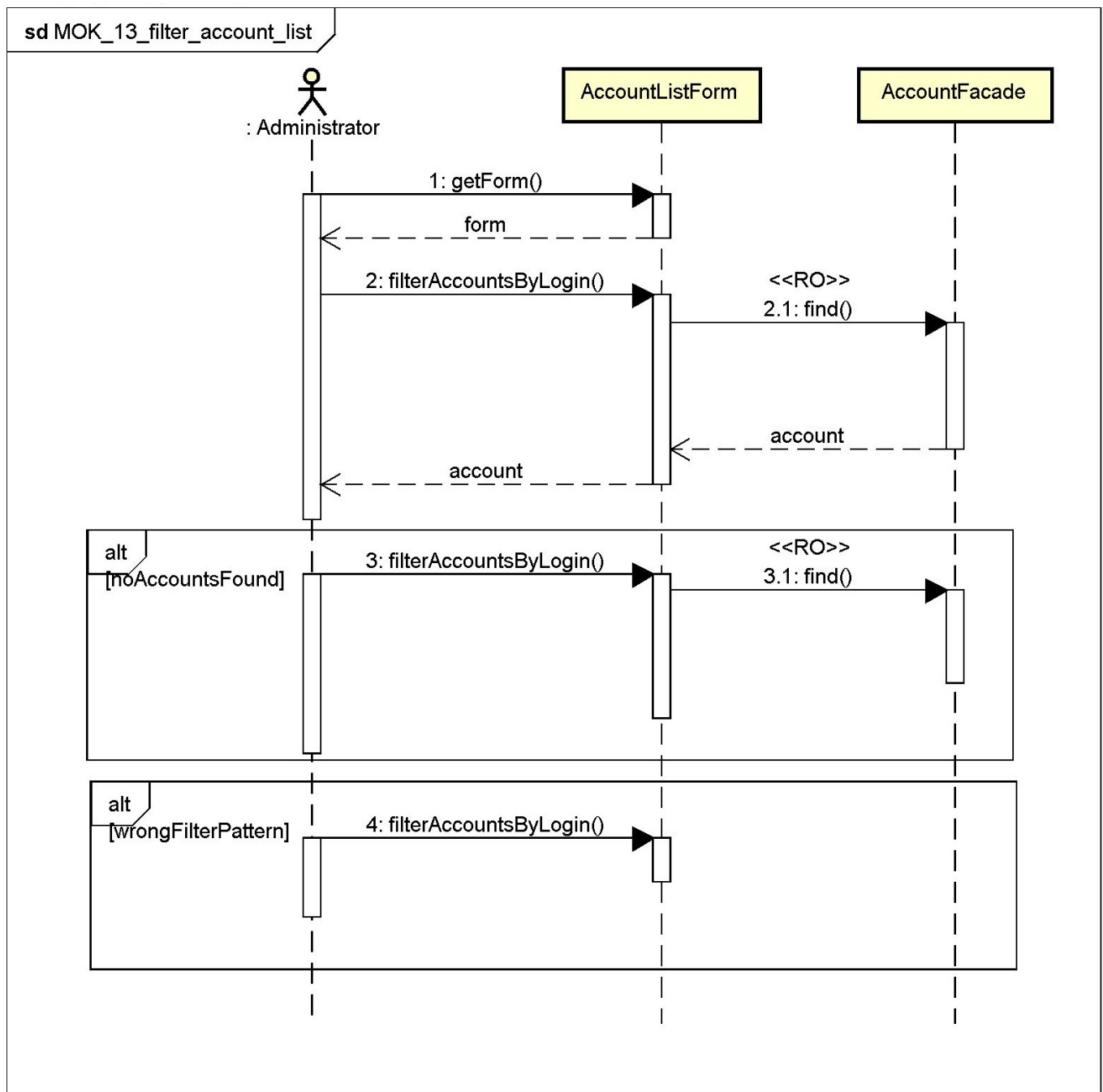
**MOK11 - Wyloguj (Michał Malec)**



**MOK12 - Przeglądaj listę kont (Piotr Biernacki)**

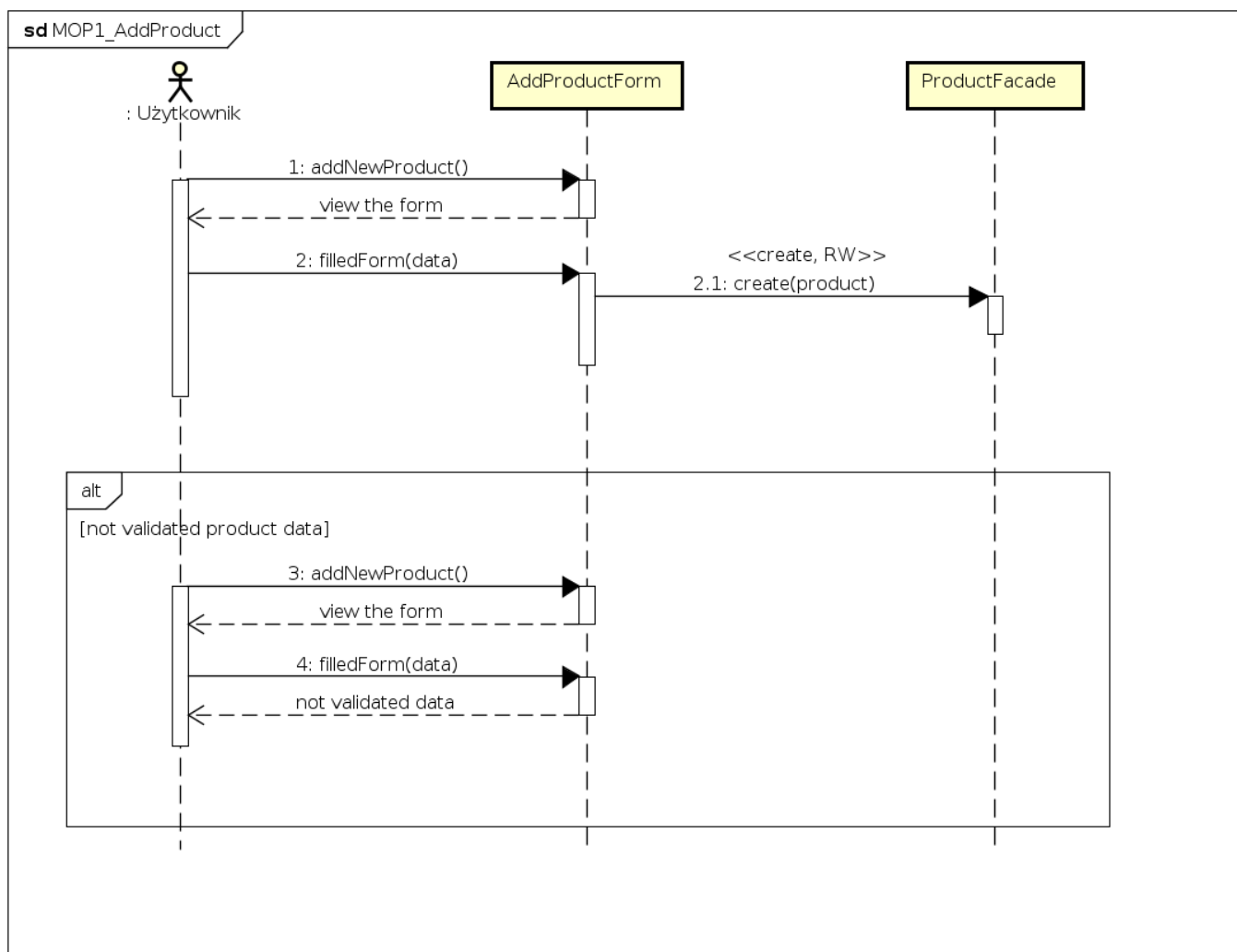


**MOK13 - Filtruj listę kont (Agata Kania)**

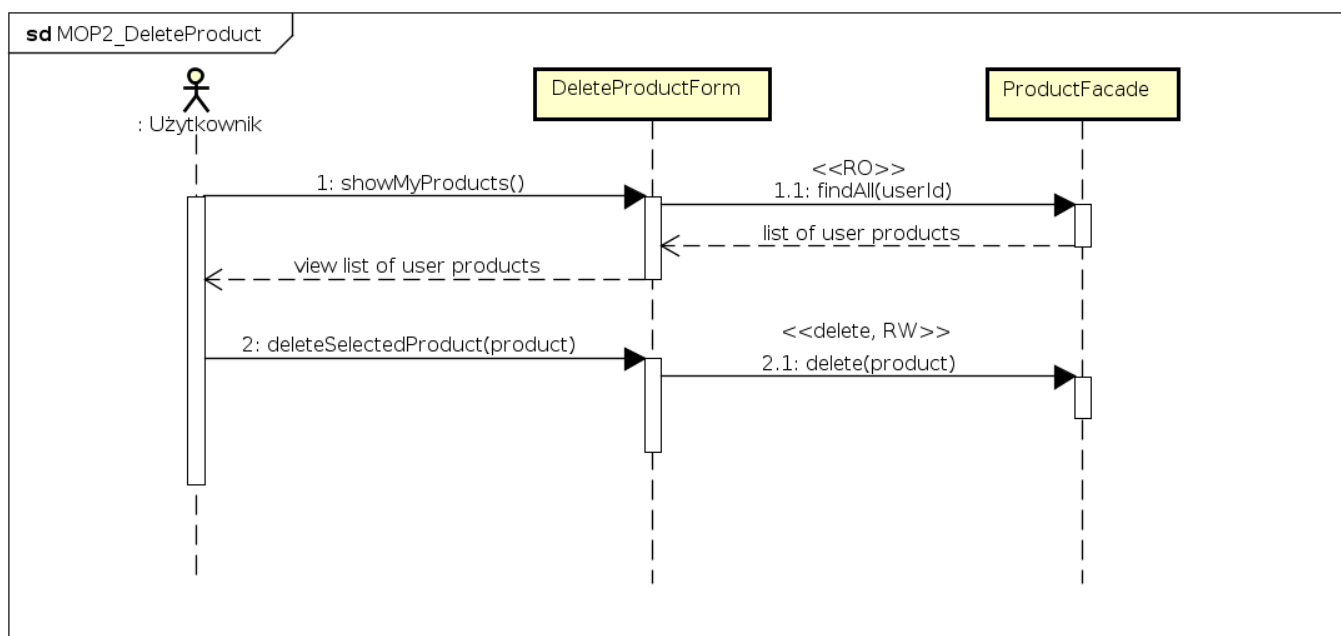


## Diagramy MOP

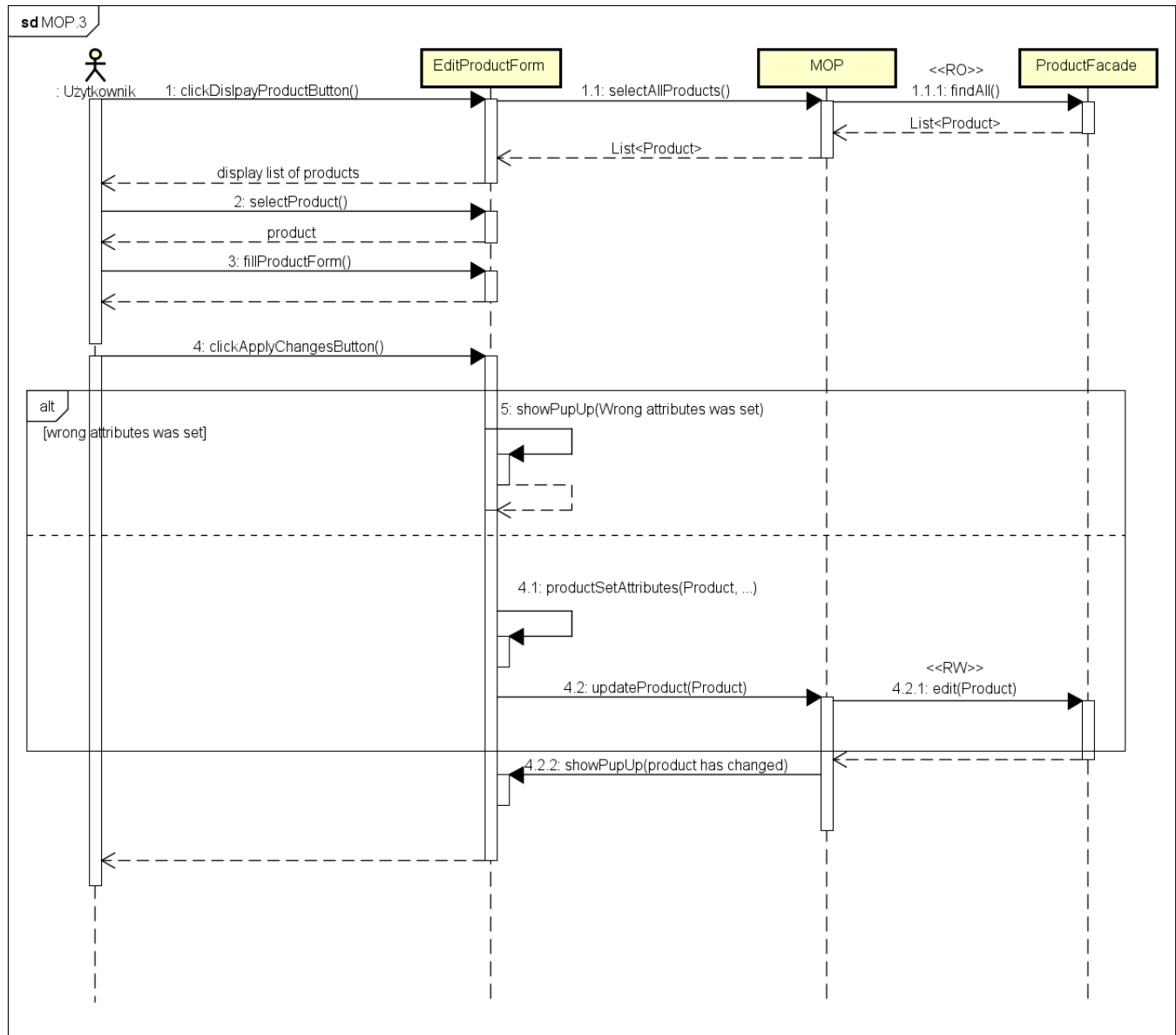
MOP1 - Dodaj produkt (Piotr Biernacki)



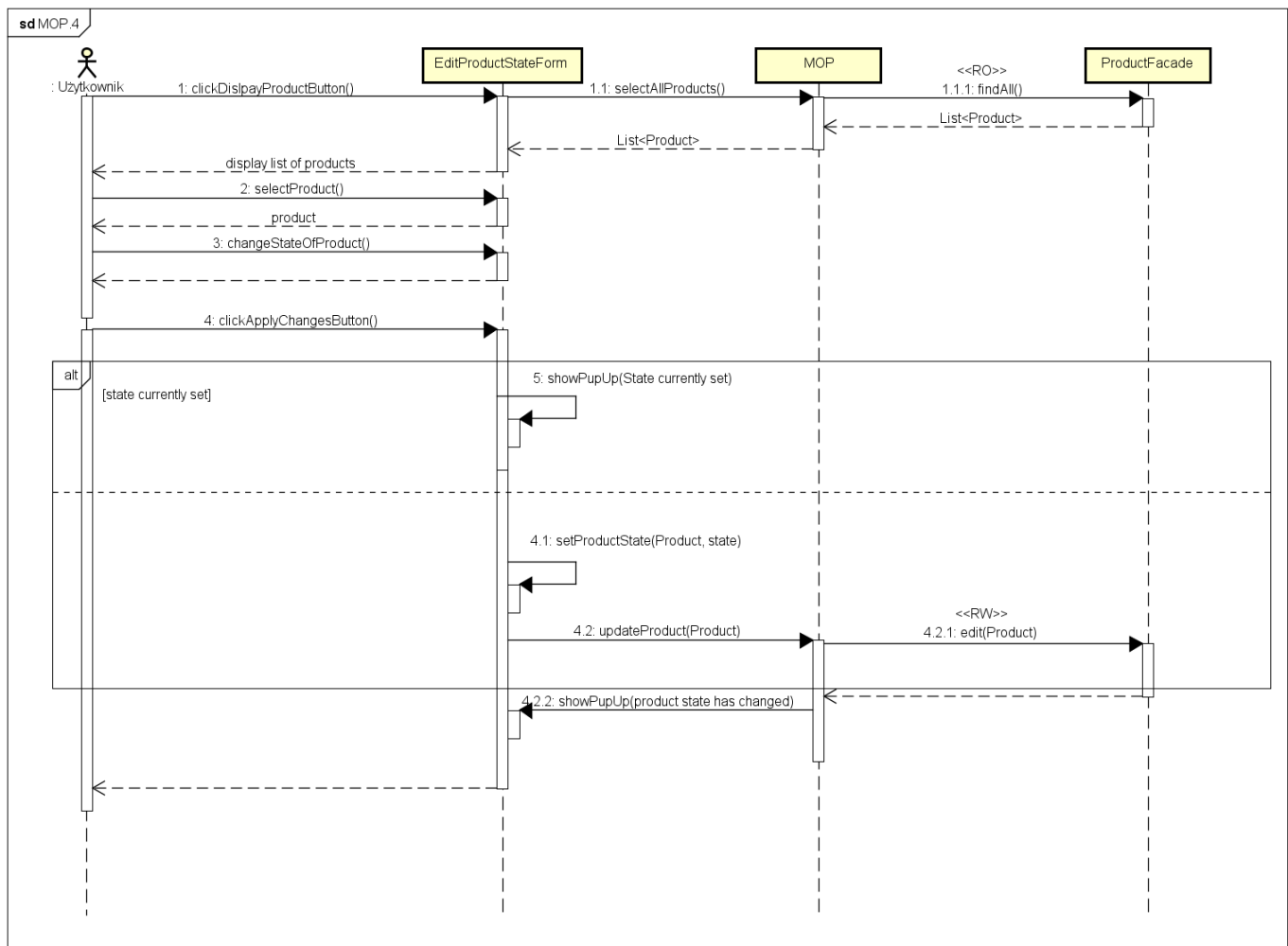
## MOP2 - Usuń produkt (Piotr Biernacki)



### MOP3 - Edytuj produkt (Filip Róžański)

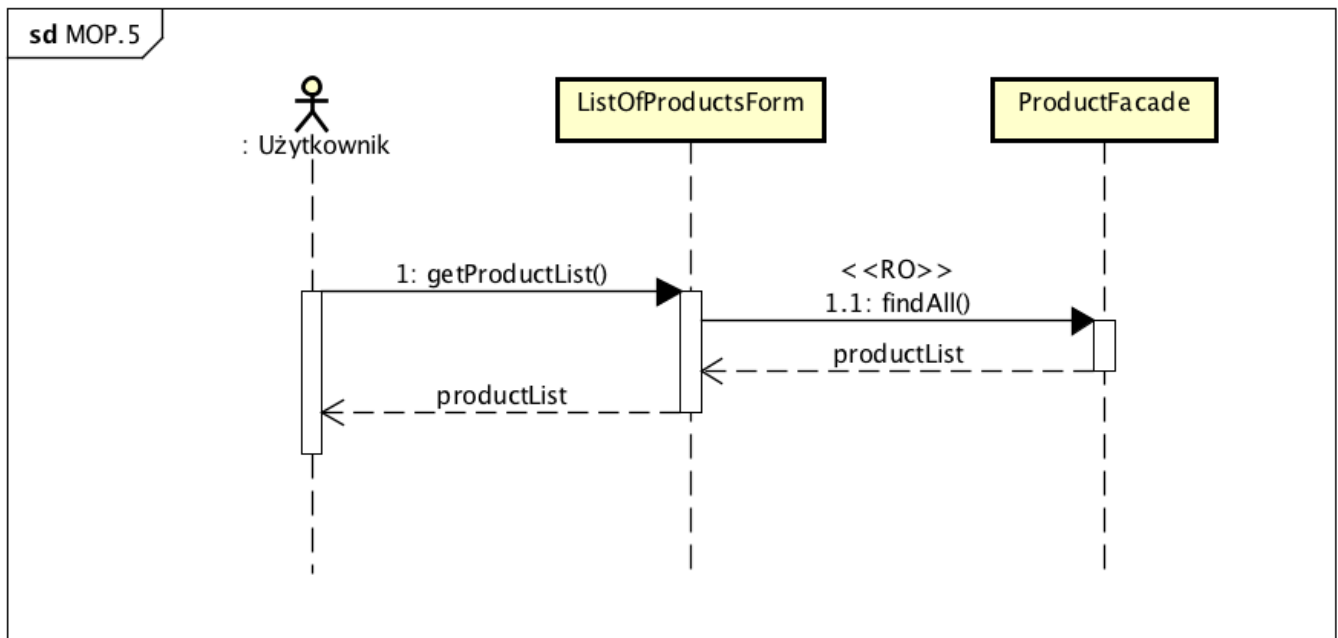


### MOP4 - Zmień stan produktu (aktywny, nieaktywny) (Filip Róžański)

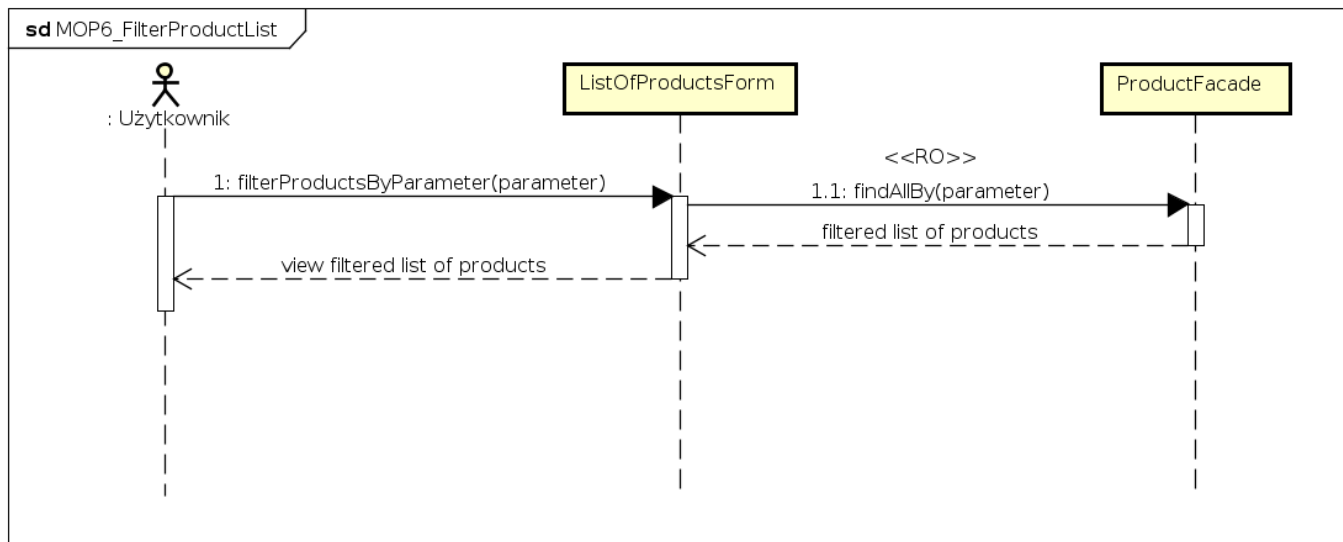


**MOP5 - Przeglądaj listę produktów (Michał Malec)**





#### MOP6 - Filtruj listę produktów (Piotr Biernacki)



#### MOP7 - Edytuj kategorie produktów (dodawanie nowych, aktywacja/dezaktywacja) (Dominik Lange)

sd MOP.7\_1

: Moderator

Category creation form

CategoryFacade

1: getForm()

form

2: addCategory()

<<RW>>  
2.1: create()

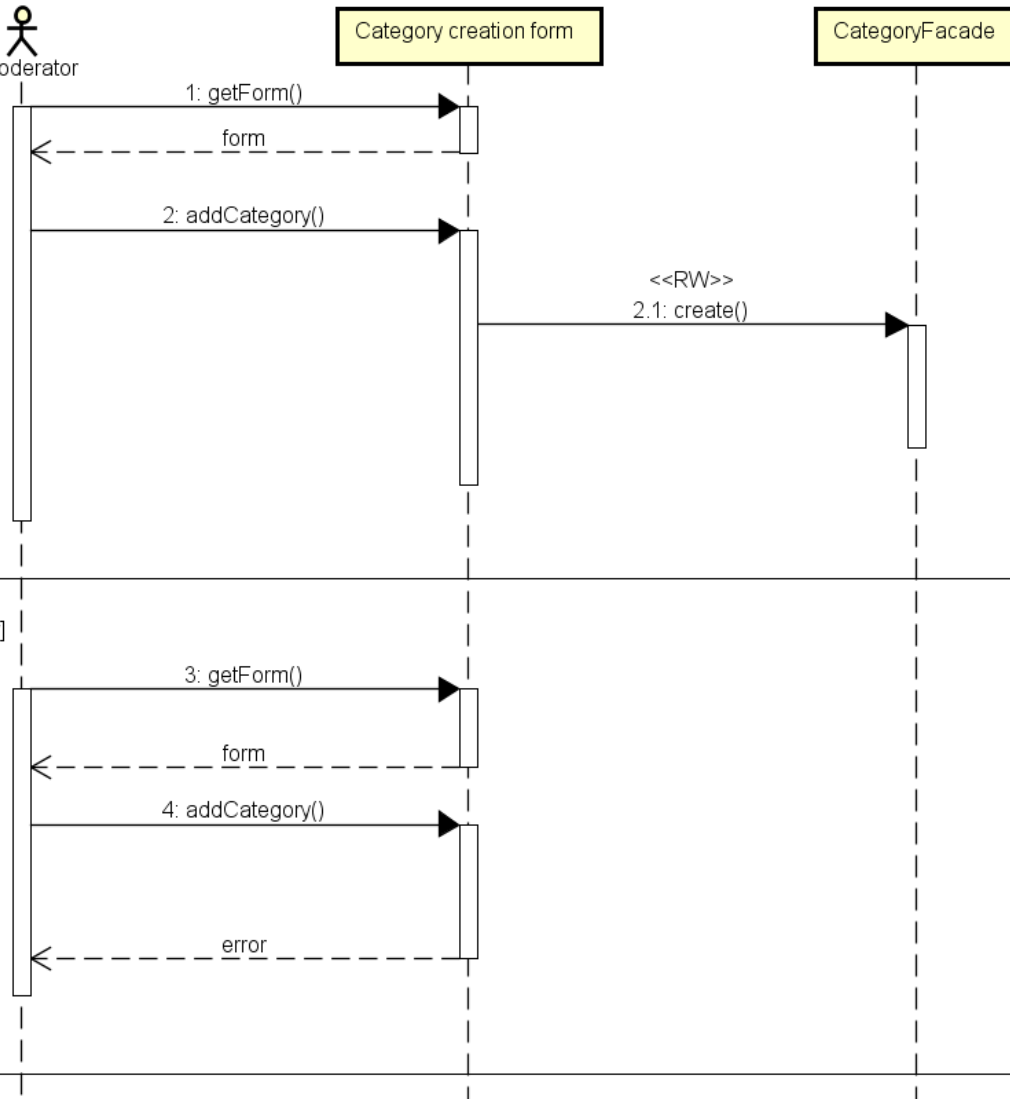
alt  
[error]

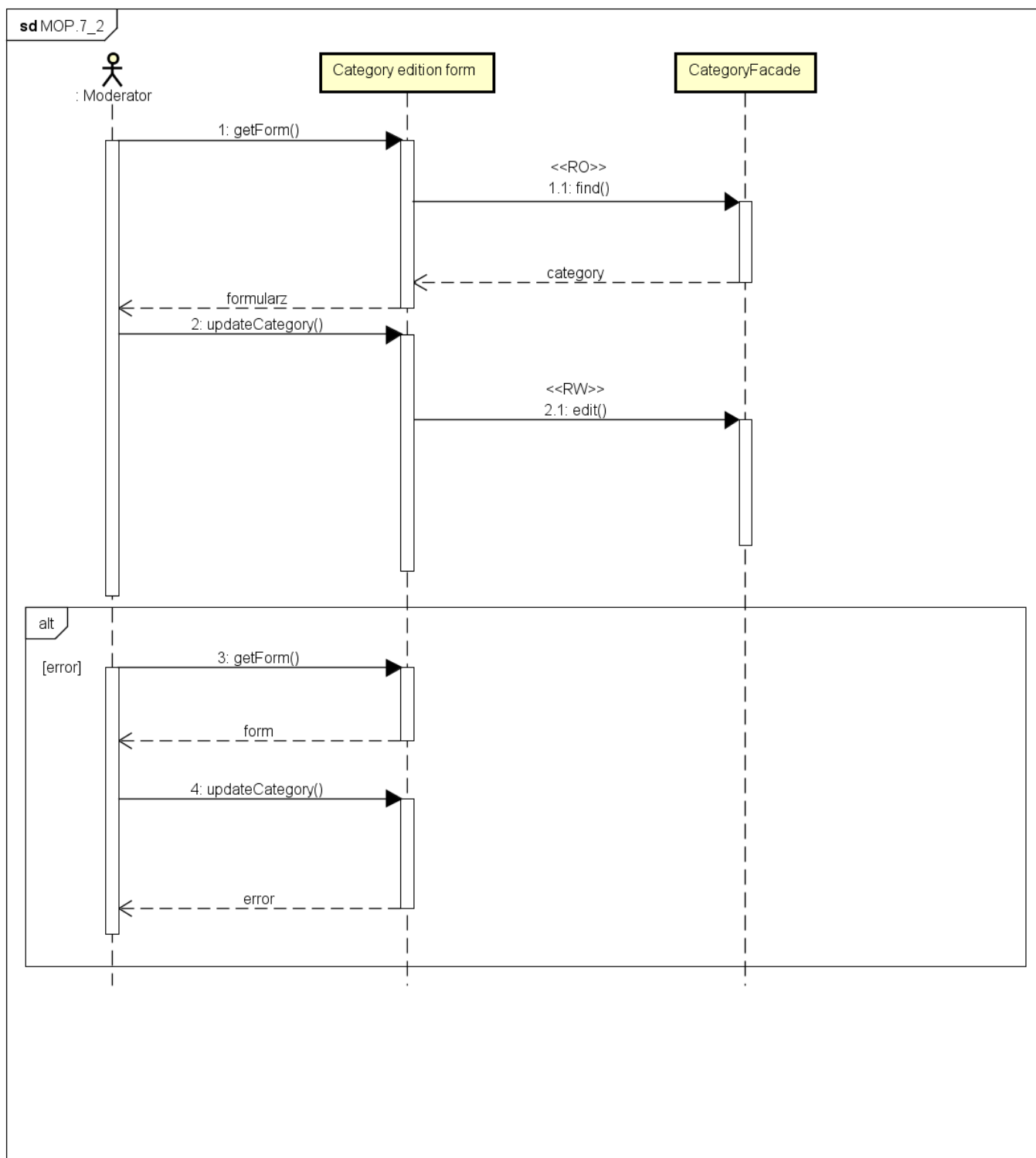
3: getForm()

form

4: addCategory()

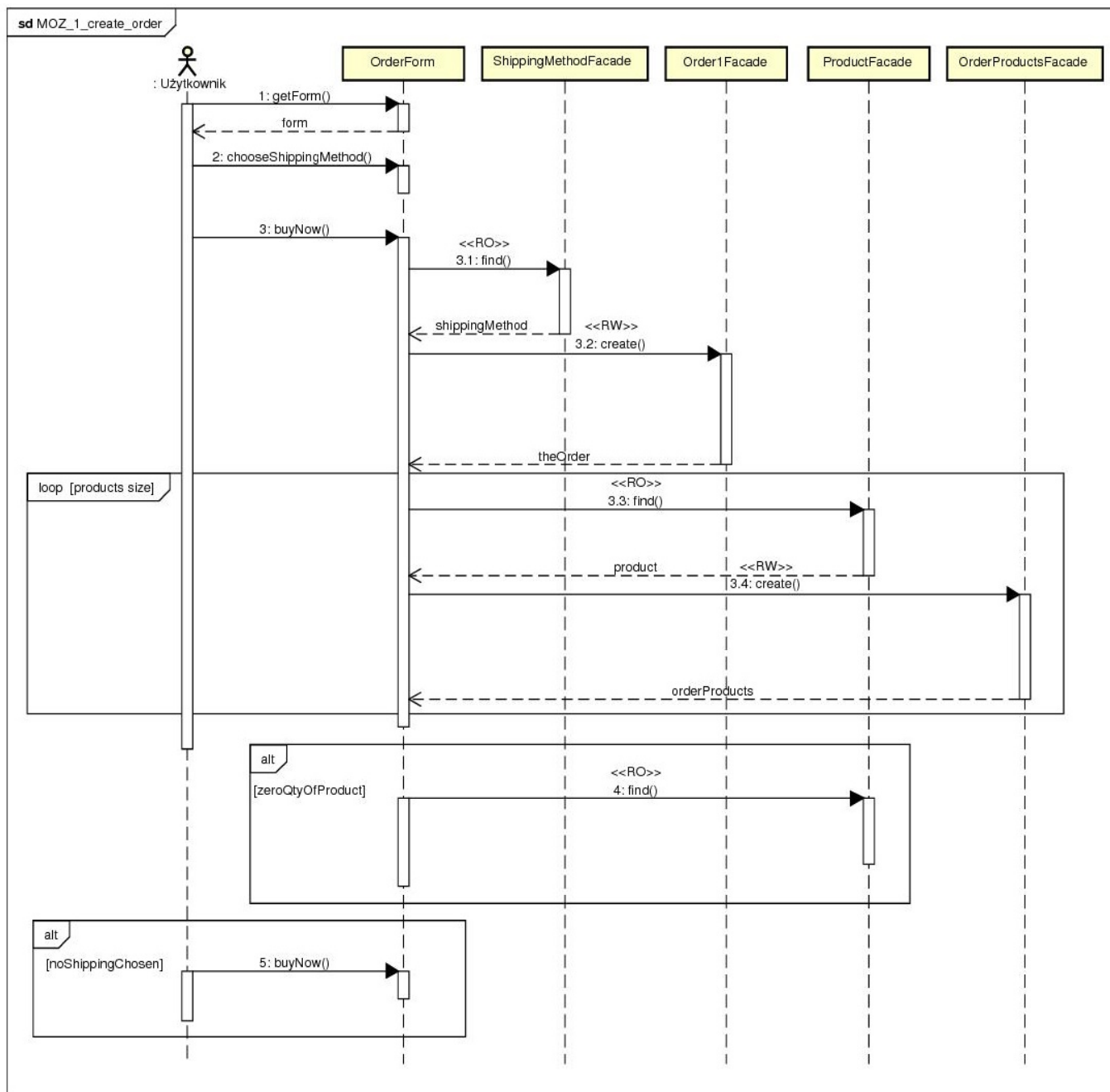
error



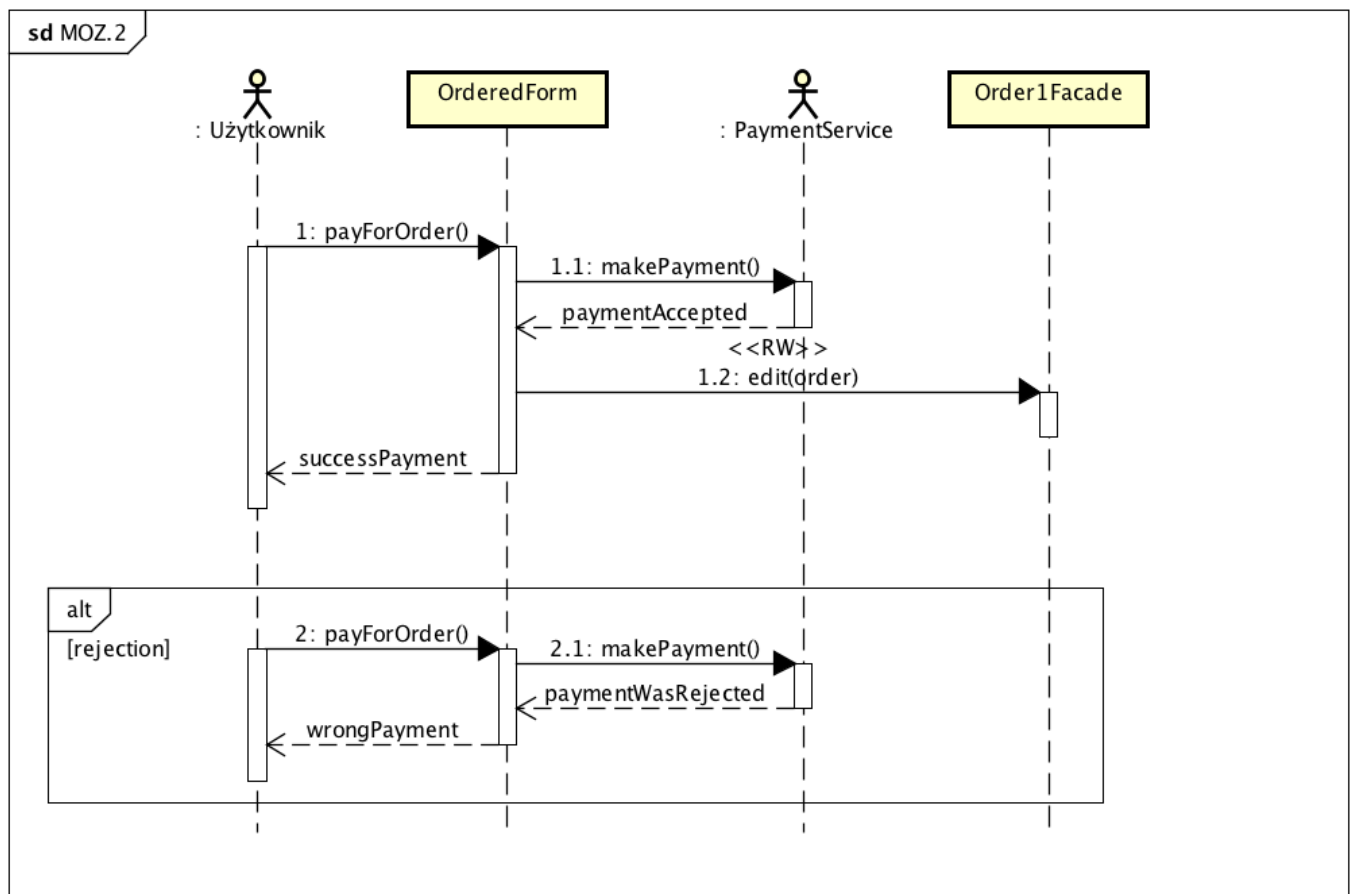


## Diagramy MOZ

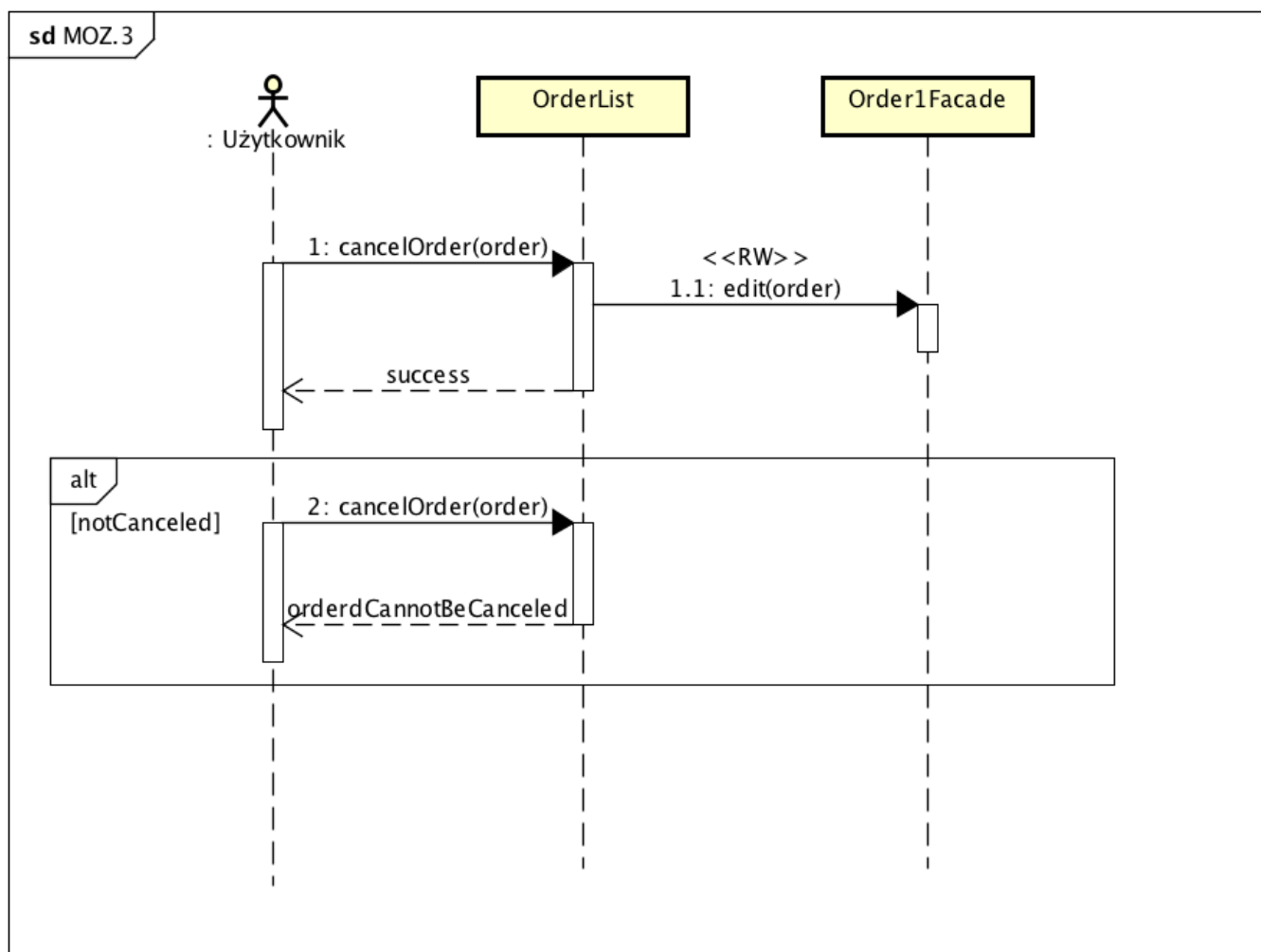
MOZ1 - Dokonaj zakupu (Agata Kania)



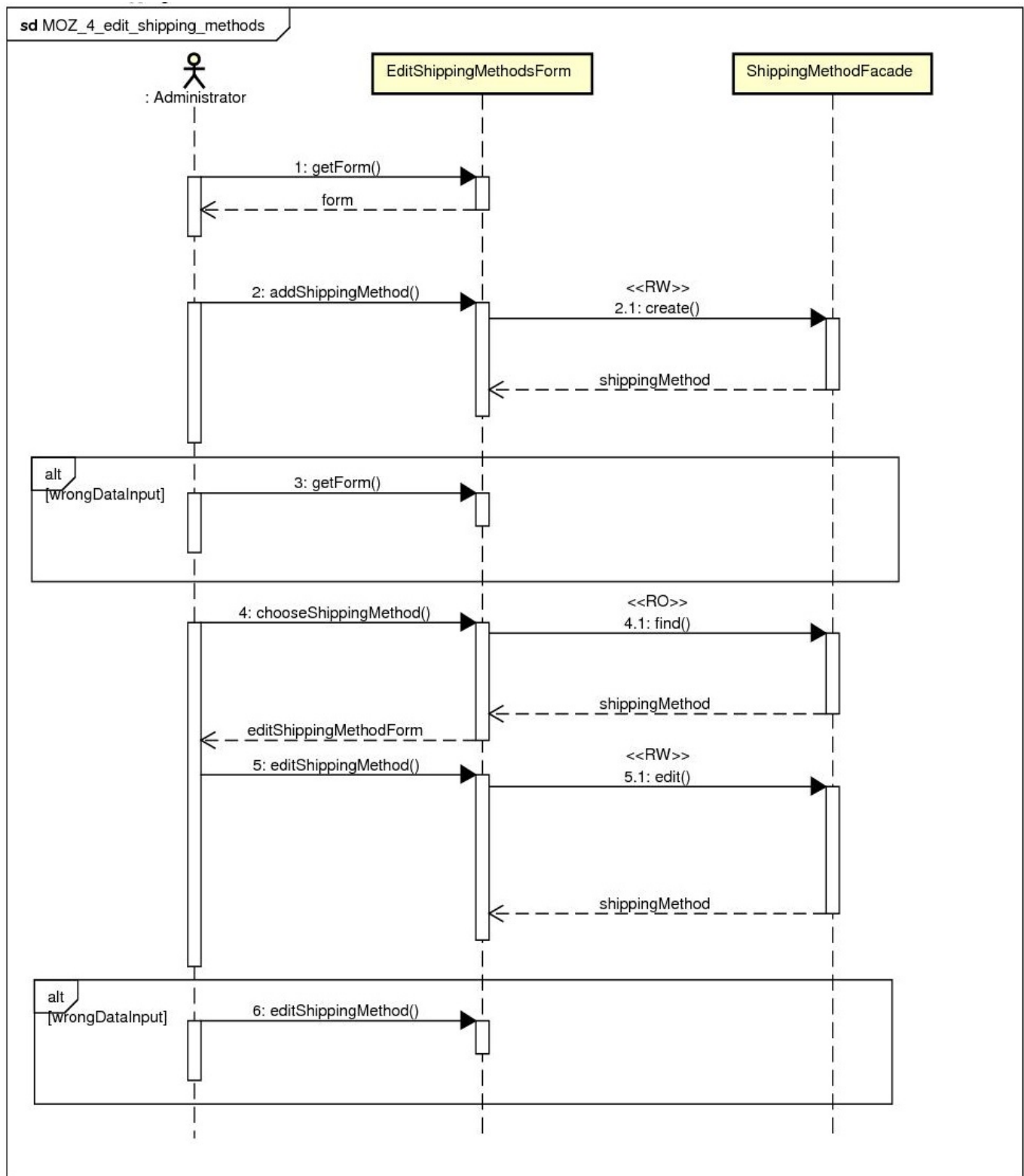
## MOZ 2 - Opłać zamówienie (Michał Malec)



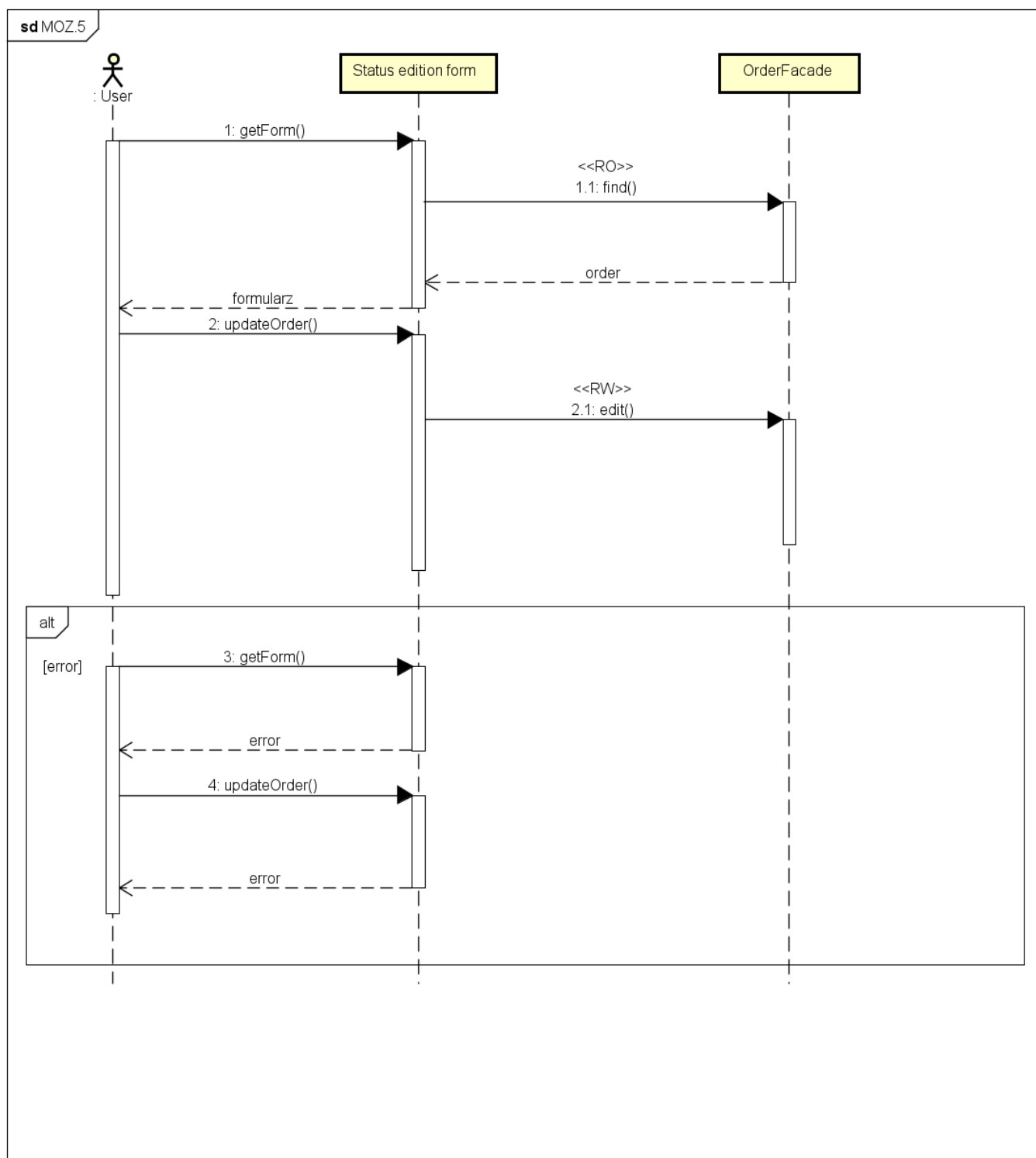
MOZ 3 - Anuluj zamówienie (Michał Malec)



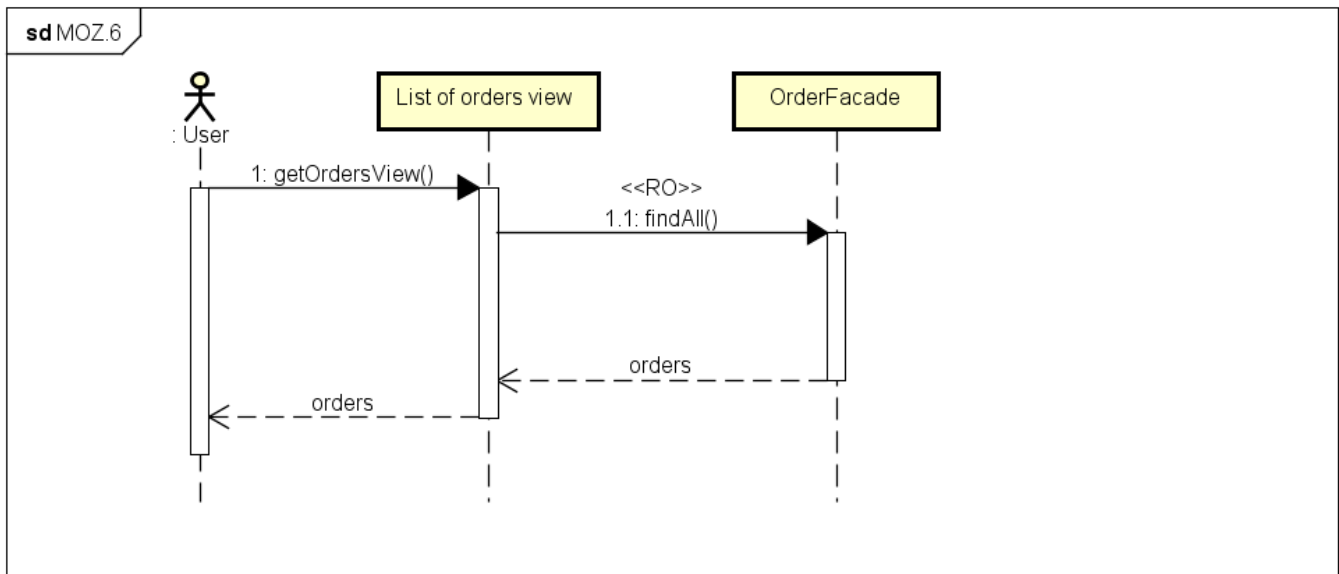
MOZ 4 - Edytuj metody wysyłki (dodawanie nowych, aktywacja/dezaktywacja) (Agata Kania)



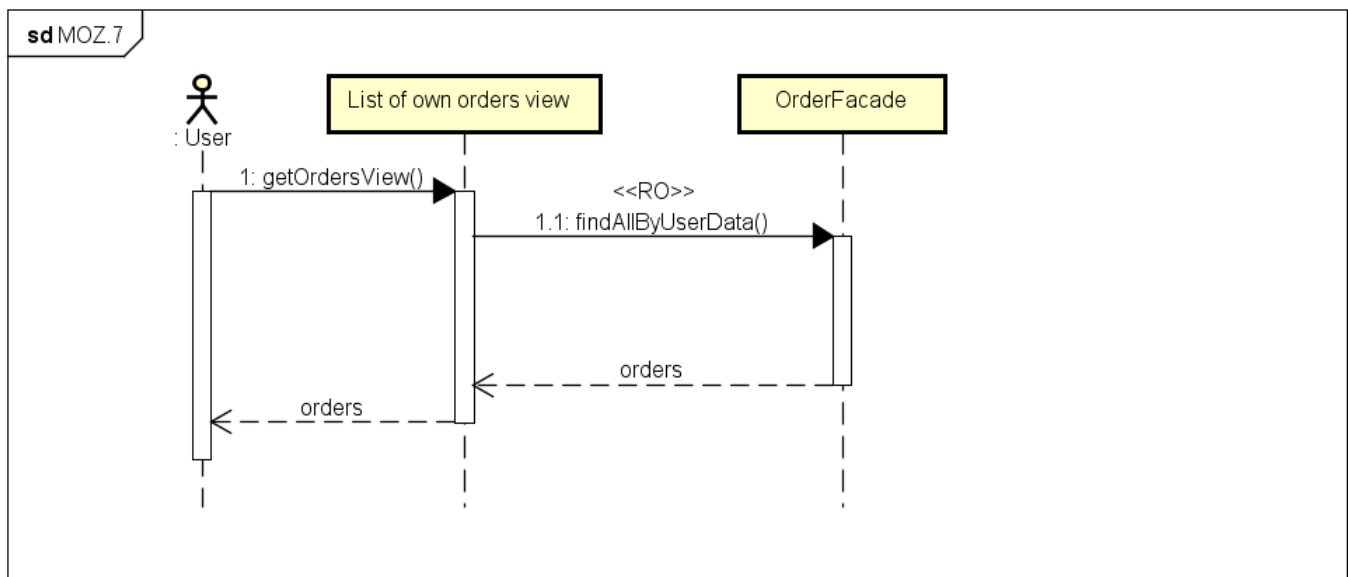
MOZ 5 - Zmień status zamówienia (Dominik Lange)



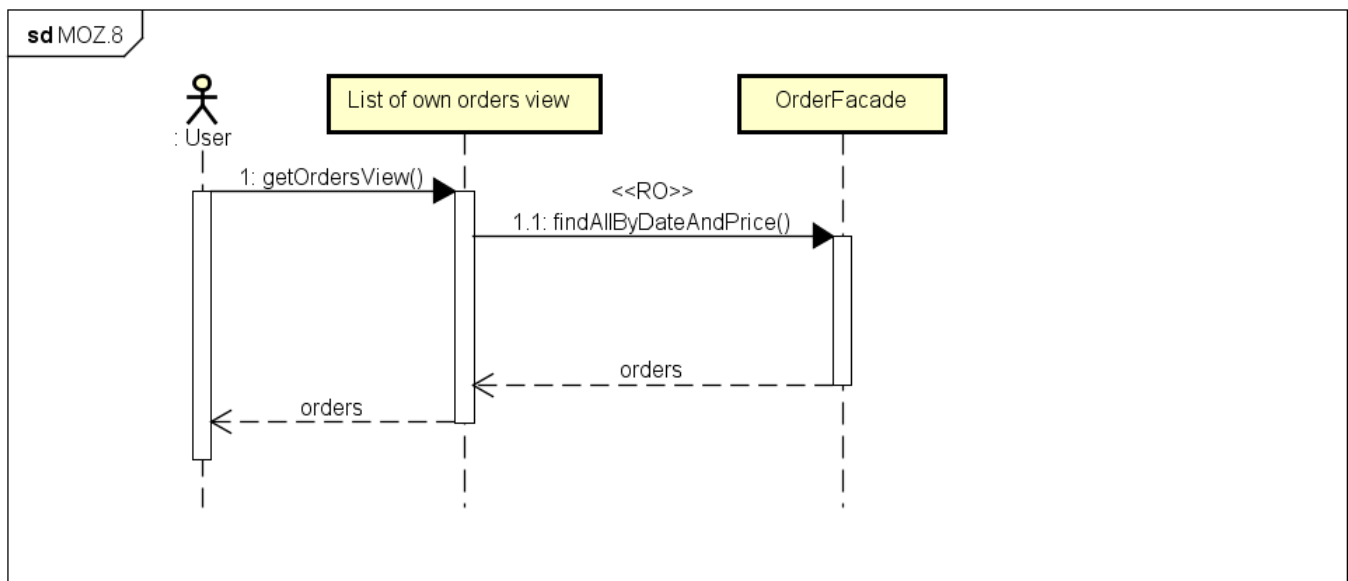




#### MOZ 7 - Przeglądaj listę własnych zamówień (Dominik Lange)



#### MOZ 8 - Filtruj zamówienia (Dominik Lange)



## 8 Konfiguracja uwierzytelniania w aplikacji 2 pkt

### Autoryzacja konta po rejestracji:

- Nowy użytkownik otrzyma wiadomość w wersji elektronicznej z linkiem aktywacyjnym prowadzącym do strony aktywującej konto. Aplikacja przechwyci tę operację i dokona zmiany stanu krotki *confirm* w tabeli *account* w bazie danych [SSBD01](#).

### Blokada konta:

- Blokowanie i odblokowywanie kont będzie odbywało się poprzez zmianę stanu właściwości *active* krotki w tabeli *account* w bazie danych [SSBD01](#). Widok *glassfish\_auth\_view* wyświetli jedynie aktywne konta.

### Poziomy dostęp:

- Przydzielanie i odbieranie poziomów dostępu danemu użytkownikowi będzie odbywało się na zasadzie dodawania i usuwania krotki z danym poziomem dostępu w tabeli *access\_alevel* w bazie danych [SSBD01](#). Dodatkowo jest możliwość wyłączenia poziomu dostępu dla wszystkich użytkowników jednocześnie. Widok *glassfish\_auth\_view* wyświetli konta z aktywnymi poziomami dostępu.

### Konfiguracja puli połączeń JDBC:

### Konfiguracija zasobu JDBC:

HomeAbout...

User: admin | Domain: ssbd01 | Server: studapp.it.p.lodz.pl

GlassFish™ Server Open Source Edition

Tree

Common Tasks

Domain

- server (Admin Server)

Clusters

Standalone Instances

Nodes

Applications

Lifecycle Modules

Monitoring Data

Resources

- Concurrent Resources
- Connectors
- JDBC
  - JDBC Resources
    - jdbc/\_\_TimerPool
    - jdbc/\_\_default
    - jdbc/ssbd01authDS
  - JDBC Connection Pools
- JMS Resources
- JNDI
- JavaMail Sessions
- Resource Adapter Configs

Configurations

- default-config
- server-config

Edit JDBC Resource

Edit an existing JDBC data source.

Load Defaults

JNDI Name:

jdbc/ssbd01authDS

Pool Name:

ssbd01authCP ▾

Use the [JDBC Connection Pools](#) page to create new pools

Deployment Order:

100

Specifies the loading order of the resource at server startup. Lower numbers are loaded first.

Description:

Status:

☒

Additional Properties (0)

Add PropertyDelete Properties

Select	Name	Value
No items found.		

Konfiguracja przestrzeni uwierzytelniania JDBC:

Home About...

User: admin Domain: ssbd01 Server: studapp.it.p.lodz.pl

GlassFish™ Server Open Source Edition

Tree

- Domain
  - server (Admin Server)
    - Clusters
    - Standalone Instances
    - Nodes
    - Applications
    - Lifecycle Modules
    - Monitoring Data
    - Resources
      - Concurrent Resources
      - Connectors
      - JDBC
      - JMS Resources
      - JNDI
      - JavaMail Sessions
      - Resource Adapter Configs
    - Configurations
      - default-config
      - server-config
        - Admin Service
        - Connector Service
        - EJB Container
        - HTTP Service
        - JVM Settings
        - Java Message Service
        - Logger Settings
        - Monitoring
        - Network Config
        - ORB
        - Security
          - Realms
            - admin-realms
            - certificate
            - file
            - ssbd01realm
        - Audit Modules
        - JACC Providers
        - Message Security
        - System Properties
        - Thread Pools

**Edit Realm**

Edit an existing security (authentication) realm.

Configuration Name: server-config

Realm Name: ssbd01realm

Class Name: com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm

Properties specific to this Class

JAAS Context: \* jdbcRealm  
Identifier for the login module to use for this realm

JNDI: \* jdbc/ssbd01authDS  
JNDI name of the JDBC resource used by this realm

User Table: \* glassfish\_auth\_view  
Name of the database table that contains the list of authorized users for this realm

User Name Column: \* login  
Name of the column in the user table that contains the list of user names

Password Column: \* password  
Name of the column in the user table that contains the user passwords

Group Table: \* glassfish\_auth\_view  
Name of the database table that contains the list of groups for this realm

Group Table User Name Column: \*  
Name of the column in the user group table that contains the list of groups for this realm

Group Name Column: \* level  
Name of the column in the group table that contains the list of group names

Password Encryption Algorithm: \* none  
This denotes the algorithm for encrypting the passwords in the database. It is a security risk to leave this field empty.

Assign Groups: \*  
Comma-separated list of group names

Database User: \*  
Specify the database user name in the realm instead of the JDBC connection pool

Database Password: \*  
Specify the database password in the realm instead of the JDBC connection pool

Digest Algorithm: \* SHA-256  
Digest algorithm (default is SHA-256); note that the default was MD5 in GlassFish versions prior to 3.1

Encoding: \*  
Encoding (allowed values are Hex and Base64)

Charset: \*  
Character set for the digest algorithm

## Sprawozdanie szczegółowe (25 pkt)

### Sprawozdanie szczegółowe

#### 9 Diagram UML klas komponentow EJB 4 pkt

Poprawny diagram UML komponentów EJB (klas i interfejsów), oddzielny dla każdego modułu funkcjonalnego.

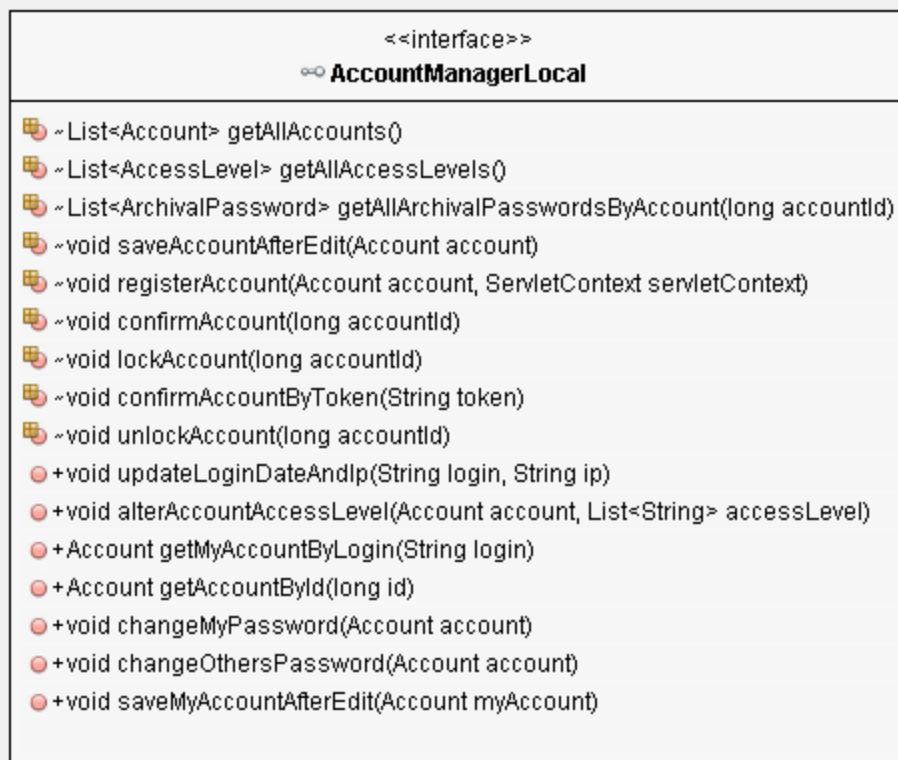
Diagram musi eksponować:

- relacje pomiędzy poszczególnymi klasami i interfejsami
- typ komponentu
- wyjątki weryfikowalne deklarowane przez poszczególne metody

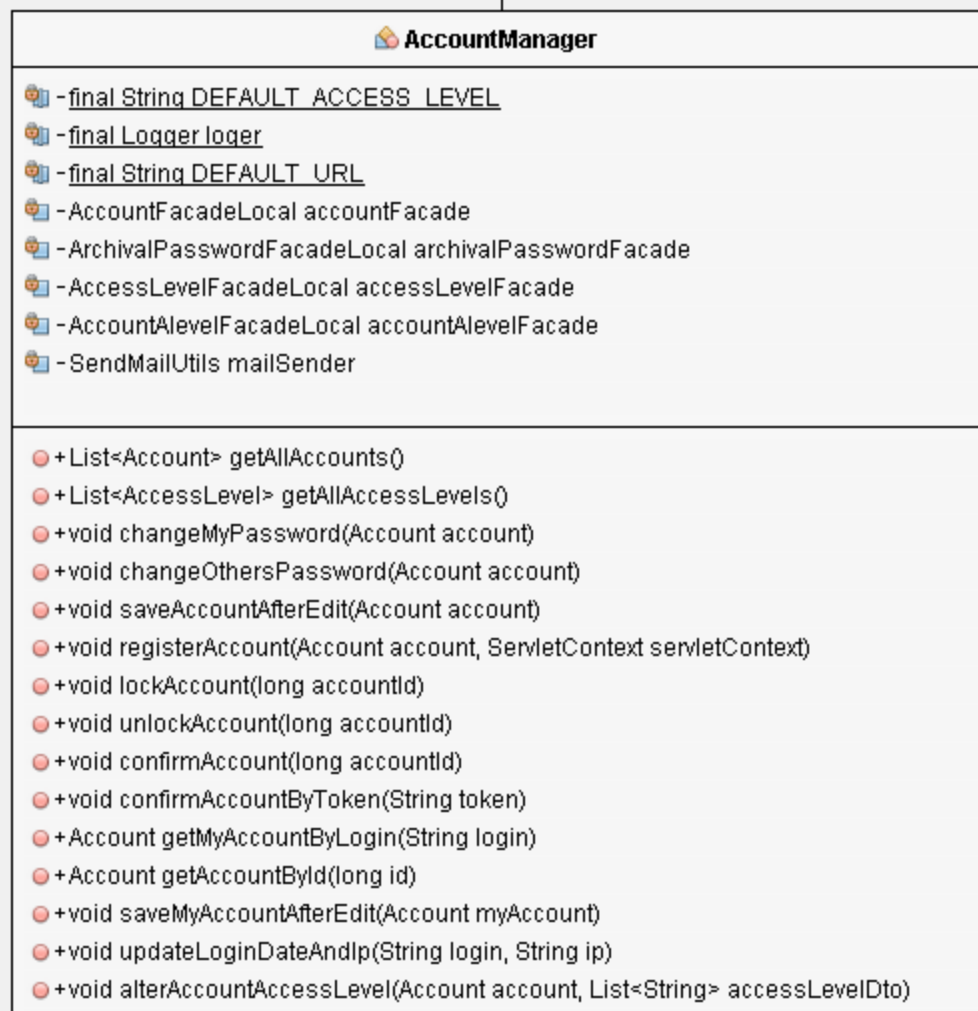
Wymagane jest także zrealizowanie i udostępnienie w repozytorium kodu źródłowego następujących składowych projektu:

- komponenty modułu MOK - pełna implementacja
- komponenty pozostałych modułów - zdefiniowane API, tzn. deklaracje klas i interfejsów, deklaracje metod w interfejsach, "puste" (np. rzucające wyjątek `UnsupportedOperationException`) implementacje metod w komponentach

*Diagram 1. Diagram UML komponentów EJB dla MOK.*



**<<implements>>**



```

-void addAccessLevelToAccount(Account account, AccessLevel accessLevel)
-void dismissAccessLevelFromAccount(Account account, AccessLevel accessLevel)
+List<ArchivalPassword> getAllArchivalPasswordsByAccount(long accountId)
-Date generateCurrentDate()
-void checkVerificationToken(Account account)
-void checkIfAccountConfirmed(Account account)
-void sendMailWithVerificationLink(String mail, String verificationLink)
-String createVerificationLink(Account account, ServletContext servletContext)

```

Tabela 1. Metody klasy AccountManager deklarujące wyjątki.

Metoda	Wyjątki
changeMyPassword	<ul style="list-style-type: none"> <li>AppBaseException</li> </ul>
changeOthersPassword	<ul style="list-style-type: none"> <li>AppBaseException</li> </ul>
saveAccountAfterEdit	<ul style="list-style-type: none"> <li>AppBaseException</li> </ul>
registerAccount	<ul style="list-style-type: none"> <li>AppBaseException</li> </ul>
lockAccount	<ul style="list-style-type: none"> <li>AppBaseException</li> </ul>
unlockAccount	<ul style="list-style-type: none"> <li>AppBaseException</li> </ul>
confirmAccount	<ul style="list-style-type: none"> <li>AppBaseException</li> </ul>
confirmAccountByToken	<ul style="list-style-type: none"> <li>AppBaseException</li> </ul>
getMyAccountByLogin	<ul style="list-style-type: none"> <li>AppBaseException</li> </ul>
getAccountById	<ul style="list-style-type: none"> <li>AppBaseException</li> </ul>
saveMyAccountAfterEdit	<ul style="list-style-type: none"> <li>AppBaseException</li> </ul>
updateLoginDateAndIp	<ul style="list-style-type: none"> <li>AppBaseException</li> </ul>
alterAccountAccessLevel	<ul style="list-style-type: none"> <li>AppBaseException</li> </ul>
addAccessLevelToAccount	<ul style="list-style-type: none"> <li>AppBaseException</li> </ul>



dismissAccessLevelFromAccount	<ul style="list-style-type: none"> <li>AppBaseException</li> </ul>
-------------------------------	--

Diagram 2. Diagram UML komponentów EJB dla MOP.

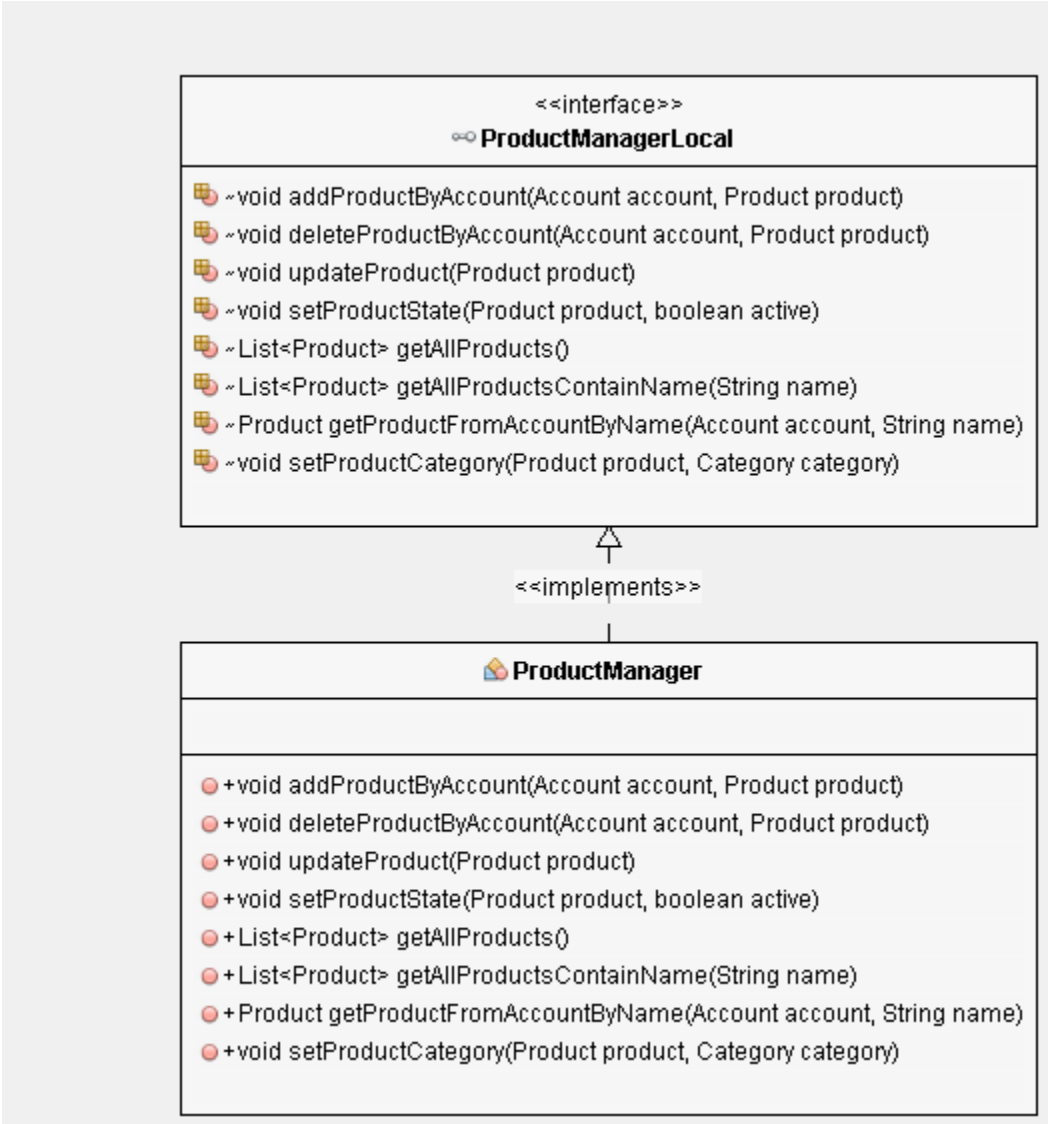


Tabela 2. Metody klasy ProductManager deklarujące wyjątki.

Metoda	Wyjątki
addProductByAccount	<ul style="list-style-type: none"> <li>NotImplementedException</li> </ul>
deleteProductByAccount	<ul style="list-style-type: none"> <li>NotImplementedException</li> </ul>
updateProduct	<ul style="list-style-type: none"> <li>NotImplementedException</li> </ul>

setProductState	<ul style="list-style-type: none"> <li>• NotImplementedException</li> </ul>
getAllProducts	<ul style="list-style-type: none"> <li>• NotImplementedException</li> </ul>
getAllProductsContainName	<ul style="list-style-type: none"> <li>• NotImplementedException</li> </ul>
getProductFromAccountByName	<ul style="list-style-type: none"> <li>• NotImplementedException</li> </ul>
setProductCategory	<ul style="list-style-type: none"> <li>• NotImplementedException</li> </ul>

Diagram 3. Diagram UML komponentów EJB dla MOZ.



Tabela 3. Metody klasy OrderManager deklarujące wyjątki.

Metoda	Wyjątki
makeOrder	<ul style="list-style-type: none"> <li>NotImplementedException</li> </ul>
makeOrderPayment	<ul style="list-style-type: none"> <li>NotImplementedException</li> </ul>
cancelOrder	<ul style="list-style-type: none"> <li>NotImplementedException</li> </ul>
addShippingMethod	<ul style="list-style-type: none"> <li>NotImplementedException</li> </ul>
setShippingMethodActive	<ul style="list-style-type: none"> <li>NotImplementedException</li> </ul>
setOrderStatus	<ul style="list-style-type: none"> <li>NotImplementedException</li> </ul>
getAllOrders	<ul style="list-style-type: none"> <li>NotImplementedException</li> </ul>
getAllOrdersByAccountAsSeller	<ul style="list-style-type: none"> <li>NotImplementedException</li> </ul>
getAllOrdersByAccountAsBuyer	<ul style="list-style-type: none"> <li>NotImplementedException</li> </ul>
getAllOrdersByDateAndPrice	<ul style="list-style-type: none"> <li>NotImplementedException</li> </ul>

## 10 Schemat bezpieczeństwa komponentów EJB 5 pkt

### Schemat bezpieczeństwa komponentów EJB

Konfiguracja deklaratywnego modelu bezpieczeństwa (kontroli dostępu) metod komponentów EJB aplikacji. Kontroli takiej muszą podlegać wszystkie metody biznesowe (udostępniane przez interfejs komponentu) wszystkich komponentów EJB - niezbędne jest propagowanie odpowiednich ustawień od komponentów dostępowych do fasad klas encyjnych.

Forma - do wyboru:

- naniesienie odpowiednich wartości na diagramy klas komponentów EJB, lub
- tabela, w której każdy wiersz reprezentuje pojedynczy przypadek użycia, i zawiera nazwy wszystkich metod (wraz z nazwami komponentów np. `makeOrderpoint.create`) wykonywanych w ramach realizacji danego przypadku użycia, wraz ze stosownymi adnotacjami

Ponadto należy zaprezentować mapowanie grup oraz tożsamości użytkowników na role aplikacji. Informacje te należy zaprezentować w formie tabeli lub wycinka z odpowiedniego deskryptora aplikacji.

Należy też zaprezentować rozwiązanie zapewniające tworzenie dziennika zdarzeń, w którym dla każdego wywołania metody biznesowej każdego z komponentów EJB w aplikacji notowane będą co najmniej:

- tożsamość użytkownika, na rzecz którego wykonywany jest przypadek użycia
- nazwa wykonywanej metody i klasy, której jest ona członkiem

- wartości (reprezentacja tekstowa) parametrów i wartości zwracanej, o ile takowe istnieją
- w przypadku zakończenia metody przez rzucenie wyjątku - nazwa i przyczyna (cause) wyjątku

W ramach prezentacji rozwiązania należy zacytować kod klas i metod realizujących funkcjonalność dziennika zdarzeń oraz fragment kodu wybranego (przykładowego) komponentu EJB obrazujący zastosowanie rozwiązania.

*Realizacja: Wszystkie metody biznesowe wszystkich komponentów (we wszystkich modułach funkcjonalnych aplikacji) muszą być opatrzone stosownymi adnotacjami konfiguracyjnymi deklaratywny model bezpieczeństwa.*

*Deskryptor aplikacji musi zawierać wszystkie przedstawione mapowania.*

*Przedstawione rozwiązanie tworzenia dziennika zdarzeń musi być zaimplementowane i zastosowane do wszystkich klas / metod komponentów.*

Tabela 1. Metody komponentów ejb wykonywane dla przypadków użycia.

Przypadek użycia	Opis	Metody
MOK.1	Zarejestruj	<ul style="list-style-type: none"> <li>• AccountManager.registerAccount @Override, @PermitAll</li> <li>• AccountFacade.create @Override @PermitAll</li> <li>• AccessLevelFacade.findByLevel @Override, @PermitAll</li> <li>• AccountAlevelFacade.create @Override, @PermitAll</li> <li>• ArchivalPasswordFacadeLocal.create @PermitAll</li> </ul>
MOK.2	Utwórz konto	<ul style="list-style-type: none"> <li>• AccountManager.registerAccount @Override, @PermitAll</li> <li>• AccountFacade.create @Override @PermitAll</li> <li>• AccessLevelFacade.findByLevel @Override, @PermitAll</li> <li>• AccountAlevelFacade.create @Override, @PermitAll</li> <li>• ArchivalPasswordFacadeLocal.create @PermitAll</li> </ul>
MOK.3	Zablokuj konto	<ul style="list-style-type: none"> <li>• AccountManager.getAllAccounts @Override, @RolesAllowed("getAllAccounts")</li> <li>• AccountFacade.findAll @RolesAllowed("findAll")</li> <li>• AccountManager.lockAccount @Override, @RolesAllowed("lockAccount")</li> <li>• AccountFacade.find @Override, @RolesAllowed("findAccount")</li> <li>• AccountFacade.edit @Override, @RolesAllowed("editAccount") @Override, @PermitAll</li> </ul>
MOK.4	Odblokuj konto	<ul style="list-style-type: none"> <li>• AccountManager.getAllAccounts @Override, @RolesAllowed("getAllAccounts")</li> <li>• AccountFacade.findAll @RolesAllowed("findAll")</li> <li>• AccountManager.unlockAccount @Override, @RolesAllowed("unlockAccount")</li> <li>• AccountFacade.find @Override, @RolesAllowed("findAccount")</li> <li>• AccountFacade.edit @Override, @RolesAllowed("editAccount") @Override, @PermitAll</li> </ul>

MOK.5	Dołącz poziom dostępu do konta	<ul style="list-style-type: none"> <li>• AccountManager.getAllAccounts @Override, @RolesAllowed("getAllAccounts")</li> <li>• AccountFacade.findAll @RolesAllowed("findAll")</li> <li>• AccountManager.getAccountById @Override, @RolesAllowed("getAccountById")</li> <li>• AccountFacade.find @Override, @RolesAllowed("findAccount")</li> <li>• AccountManager.alterAccountAccessLevel @Override, @RolesAllowed("alterAccountAccessLevel")</li> <li>• AccessLevelFacade.findByLevel @Override, @PermitAll</li> </ul>
MOK.6	Odłącz poziom dostępu od konta	<ul style="list-style-type: none"> <li>• AccountManager.getAllAccounts @Override, @RolesAllowed("getAllAccounts")</li> <li>• AccountFacade.findAll @RolesAllowed("findAll")</li> <li>• AccountManager.getAccountById @Override, @RolesAllowed("getAccountById")</li> <li>• AccountFacade.find @Override, @RolesAllowed("findAccount")</li> <li>• AccountManager.alterAccountAccessLevel @Override, @RolesAllowed("alterAccountAccessLevel")</li> <li>• AccessLevelFacade.findByLevel @Override, @PermitAll</li> </ul>
MOK.7	Zmień własne hasło	<ul style="list-style-type: none"> <li>• AccountManager.getMyAccountByLogin @Override, @RolesAllowed("getMyAccountByLogin")</li> <li>• AccountFacade.findByLogin @Override, @RolesAllowed("findByLogin")</li> <li>• AccountFacade.changeMyPassword @Override, @RolesAllowed("changeMyPassword")</li> <li>• AccountFacade.edit @Override, @RolesAllowed("editAccount")</li> <li>• ArchivalPasswordFacade.create @Override, @PermitAll</li> </ul>
MOK.8	Zmień hasło innego użytkownika	<ul style="list-style-type: none"> <li>• AccountManager.getAccountById @Override, @RolesAllowed("getAccountById")</li> <li>• AccountFacade.findAll @RolesAllowed("findAll")</li> <li>• AccountManager.changeOthersPassword @Override, @RolesAllowed("changeOthersPassword")</li> <li>• AccountFacade.edit @Override, @RolesAllowed("editAccount")</li> <li>• ArchivalPasswordFacade.create @Override, @PermitAll</li> </ul>
MOK.9	Edytuj dane własnego konta	

		<ul style="list-style-type: none"> <li>• AccountManager.getMyAccountByLogin @Override, @RolesAllowed ("getMyAccountByLogin")</li> <li>• AccountFacade.findByLogin @Override, @RolesAllowed("findByLogin") @Override, @RolesAllowed ("findByLogin")</li> <li>• AccountManager.saveMyAccountAfterEdit @Override, @RolesAllowed ("saveMyAccountAfterEdit")</li> <li>• AccountFacade.edit @Override, @RolesAllowed("editAccount") @Override, @PermitAll</li> </ul>
MOK.10	Edytuj dane konta innego użytkownika	<ul style="list-style-type: none"> <li>• AccountManager.getAccountById @Override, @RolesAllowed("getAccountById")</li> <li>• AccountFacade.find @Override, @RolesAllowed("findAccount")</li> <li>• AccountManager.saveAccountAfterEdit @Override, @RolesAllowed ("saveAccountAfterEdit")</li> <li>• AccountFacade.edit @Override, @RolesAllowed("editAccount") @Override, @PermitAll</li> </ul>
MOK.11	Wyloguj	
MOK.12	Przeglądaj listę kont	<ul style="list-style-type: none"> <li>• AccountManager.getAllAccounts @Override, @RolesAllowed("getAllAccounts")</li> <li>• AccountFacade.findAll @RolesAllowed ("findAll")</li> </ul>
MOK.13	Filtruj listę kont	<ul style="list-style-type: none"> <li>• AccountManager.getAllAccounts @Override, @RolesAllowed("getAllAccounts")</li> <li>• AccountFacade.findAll @RolesAllowed ("findAll")</li> </ul>
MOK.14	Wyświetl statystyki konta użytkownika	<ul style="list-style-type: none"> <li>• AccountManager.getAllAccounts @Override, @RolesAllowed("getAllAccounts")</li> <li>• AccountFacade.findAll @RolesAllowed ("findAll")</li> </ul>
MOP.1	Dodaj produkt	<ul style="list-style-type: none"> <li>• ProductManager.addProductByAccountLogin @Override, @RolesAllowed ("addProductByAccountLogin")</li> <li>• CategoryManager.getCategoryById @Override, @RolesAllowed ("getCategoryById")</li> <li>• CategoryFacade.find @RolesAllowed ("find")</li> <li>• ProductFacade.find @Override, @RolesAllowed("findProduct")ByLogin @Override, @RolesAllowed ("findByLoginForProduct")</li> <li>• ProductFacade.create @PermitAll</li> </ul>
MOP.2	Usuń produkt	

		<ul style="list-style-type: none"> <li>• ProductManager.deleteProductByAccount @Override, @RolesAllowed("deleteProductByAccount")</li> <li>• ProductFacade.find @Override, @RolesAllowed("findProduct")</li> <li>• AccountFacade.findByLogin @Override, @RolesAllowed("findByLogin")</li> <li>• ProductFacade.remove @Override, @RolesAllowed("removeProduct")</li> <li>• AccountFacade.edit @Override, @RolesAllowed("editAccount")</li> </ul>
MOP.3	Edytuj produkt	<ul style="list-style-type: none"> <li>• ProductManager.updateProduct @Override, @RolesAllowed("updateProduct")</li> <li>• ProductFacade.find @Override, @RolesAllowed("findProduct")</li> <li>• UnitFacade.find @Override, @RolesAllowed("findUnit")</li> <li>• ProductFacade.edit @Override, @RolesAllowed("editProduct")</li> </ul>
MOP.4	Zmień stan produktu (aktywny, nieaktywny)	<ul style="list-style-type: none"> <li>• ProductManager.setProductState @Override, @RolesAllowed("setProductState")</li> <li>• ProductManager.activeProduct @Override, @RolesAllowed("activeProduct") <ul style="list-style-type: none"> <li>• ProductFacade.find @Override, @RolesAllowed("findProduct")</li> <li>• ProductFacade.edit @Override, @RolesAllowed("editProduct")</li> </ul> </li> <li>• ProductManager.setProductState @Override, @RolesAllowed("setProductState")</li> <li>• ProductManager.deactiveProduct @Override, @RolesAllowed("deactiveProduct") <ul style="list-style-type: none"> <li>• ProductFacade.find @Override, @RolesAllowed("findProduct")</li> <li>• ProductFacade.edit @Override, @RolesAllowed("editProduct")</li> </ul> </li> </ul>
MOP.5	Przeglądaj listę produktów	<ul style="list-style-type: none"> <li>• ProductManager.getAllProducts @Override, @RolesAllowed("getAllProducts")</li> <li>• ProductFacade.find @Override, @RolesAllowed("findProduct")</li> </ul>
MOP.6	Filtruj listę produktów	<ul style="list-style-type: none"> <li>• ProductManager.getAllProducts @Override, @RolesAllowed("getAllProducts")</li> <li>• ProductFacade.find @Override, @RolesAllowed("findProduct")</li> </ul>
MOP.7	Edytuj kategorie produktów (dodawanie nowych, aktywacja/dezaktywacja)	<ul style="list-style-type: none"> <li>• CategoryManager.addCategory @Override, @RolesAllowed("addCategory")</li> <li>• CategoryFacade.create @Override, @RolesAllowed("addCategory")</li> </ul>
MOP.8	Przeglądaj własną listę produktów	<ul style="list-style-type: none"> <li>• ProductManager.getMyProducts @Override, @RolesAllowed("getMyProducts")</li> </ul>

		<ul style="list-style-type: none"> <li>ProductFacade.find @Override, @RolesAllowed("findProduct") ByOwnerLogin</li> </ul>
MOP.9	Przeglądaj listę kategorii produktów	<ul style="list-style-type: none"> <li>CategoryManager.getAllCategories @Override, @RolesAllowed("getAllCategories")</li> <li>CategoryFacade.find @RolesAllowed("find") @RolesAllowed("find")</li> </ul>
MOP.10	Aktywacja/deaktywacja kategorii produktów	<ul style="list-style-type: none"> <li>CategoryManager.activateCategory @Override, @RolesAllowed("activateCategory")</li> <li>CategoryFacade.find @RolesAllowed("find")</li> <li>CategoryFacade.edit @Override, @RolesAllowed("editCategory")</li> <li>CategoryManager.deactivateCategory @Override, @RolesAllowed("deactivateCategory")</li> <li>CategoryFacade.find @RolesAllowed("find")</li> <li>CategoryFacade.edit @Override, @RolesAllowed("editCategory")</li> </ul>
MOZ.1	Dokonaj zakupu	<ul style="list-style-type: none"> <li>OrderManager.makeOrder @Override, @RolesAllowed("makeOrder")</li> <li>ProductFacade.find @Override, @RolesAllowed("findProduct")</li> <li>UnitFacade.find @Override, @RolesAllowed("findUnit")</li> <li>AccountFacade.findByLogin @Override, @RolesAllowed("findByLogin")</li> <li>ShippingMethodFacade.find @Override, @RolesAllowed("findShippingMethod")</li> <li>OrderStatusFacade.find @Override, @RolesAllowed("getOrderStatusById")</li> <li>OrderShippingFacade.create @Override, @RolesAllowed("addShippingMethod") @PermitAll</li> <li>OrderFacade.create @PermitAll</li> <li>OrderProductsFacade.create @PermitAll</li> <li>ProductFacade.edit @Override, @RolesAllowed("editProduct")</li> <li>AccountFacade.edit @Override, @RolesAllowed("editAccount")</li> </ul>
MOZ.2	Przeglądaj metody wysyłki	<ul style="list-style-type: none"> <li>ShippingManager.getAllMethods @Override, @RolesAllowed("getAllMethods")</li> <li>ShippingFacade.find @Override, @RolesAllowed("findShippingMethod") All @PermitAll</li> </ul>
MOZ.3	Dodaj metodę wysyłki	<ul style="list-style-type: none"> <li>ShippingManager.addShippingMethod @Override, @RolesAllowed("addShippingMethod")</li> <li>AccountFacade.findByLogin @Override, @RolesAllowed("findByLogin")</li> </ul>



		<ul style="list-style-type: none"> <li>ShippingFacade.create @Override, @RolesAllowed("addShippingMethod")</li> </ul>
MOZ.4	Aktywacja/deaktywacja metody wysyłki	<ul style="list-style-type: none"> <li>ShippingManager.activateShippingMethod @Override, @RolesAllowed("activateShippingMethod")</li> <li>ShippingFacade.find @Override, @RolesAllowed("findShippingMethod")</li> <li>ShippingFacade.edit @Override, @RolesAllowed("editShippingMethod")</li> <li>ShippingManager.deactivateShippingMethod @Override, @RolesAllowed("deactivateShippingMethod")</li> <li>ShippingFacade.find @Override, @RolesAllowed("findShippingMethod")</li> <li>ShippingFacade.edit @Override, @RolesAllowed("editShippingMethod")</li> </ul>
MOZ.5	Edytuj stan zamówienia	<ul style="list-style-type: none"> <li>OrderManager.getOrder1ById @Override, @RolesAllowed("getOrder1ById")</li> <li>OrderFacade.find @Override, @RolesAllowed("getOrder1ById")</li> <li>OrderManager.getOrderStatusById @Override, @RolesAllowed("getOrderStatusById")</li> <li>orderStatusFacade.find @Override, @RolesAllowed("getOrderStatusById")</li> <li>OrderManager.setOrderStatus @Override, @RolesAllowed("setOrderStatus")</li> <li>OrderFacade.edit @PermitAll</li> </ul>
MOZ.6	Przeglądaj wszystkie zamówienia	<ul style="list-style-type: none"> <li>OrderManager.getAllOrders @Override, @RolesAllowed("getAllOrders")</li> <li>OrderFacade.findAll @RolesAllowed("findAll")</li> </ul>
MOZ.7	Przeglądaj listę własnych zamówień	<ul style="list-style-type: none"> <li>OrderManager.getAllByBuyer @Override, @RolesAllowed("getAllByBuyer")</li> <li>OrderFacade.findByBuyerLogin @Override, @RolesAllowed("findOrderByBuyerLogin")</li> <li>OrderManager.getAllBySeller @Override, @RolesAllowed("getAllBySeller")</li> <li>OrderFacade.findBySellerLogin @Override, @RolesAllowed("findOrderBySellerLogin")</li> </ul>
MOZ.8	Filtruj listę z zamówieniami	<ul style="list-style-type: none"> <li>OrderManager.getAllByBuyer @Override, @RolesAllowed("getAllByBuyer")</li> <li>OrderFacade.findByBuyerLogin @Override, @RolesAllowed("findOrderByBuyerLogin")</li> <li>OrderManager.getAllBySeller @Override, @RolesAllowed("getAllBySeller")</li> <li>OrderFacade.findBySellerLogin @Override, @RolesAllowed("findOrderBySellerLogin")</li> </ul>

```
<security-role-mapping>
  <role-name>USER</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>MANAGER</role-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>ADMIN</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>VIRTUAL</role-name>
  <group-name>virtual</group-name>
</security-role-mapping>
<!--MOK-->
<security-role-mapping>
  <role-name>getAllAccounts</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAllAccessLevels</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>saveAccountAfterEdit</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>confirmAccount</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>lockAccount</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>confirmAccount</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>unlockAccount</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>addAccessLevelToAccount</role-name>
  <group-name>admin</group-name>
```

```
</security-role-mapping>
<security-role-mapping>
  <role-name>dismissAccessLevelFromAccount</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>updateLoginDateAndIp</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
  <group-name>virtual</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAccountAlevel</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>changeMyPassword</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>changeOthersPassword</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>saveMyAccountAfterEdit</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>alterAccountAccessLevel</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAllArchivalPasswordsByAccount</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getMyAccountByLogin</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
  <group-name>virtual</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAccountById</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
```

```
<security-role-mapping>
  <role-name>saveMyAccountAfterEdit</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAccessLevelById</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<!--MOP-->
<security-role-mapping>
  <role-name>addProductByAccountLogin</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findAllActive</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getActiveCategories</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAllUnits</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAllCategories</role-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getCategoryById</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>activateCategory</role-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>deactivateCategory</role-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getUnitById</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>deleteProductByAccount</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
```

```
<role-name>updateProduct</role-name>
<group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>setProductState</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAllProducts</role-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAllActiveProducts</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getMyProducts</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findProductByOwnerLogin</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAllProductsContainName</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getProductFromAccountByName</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>setProductCategory</role-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>activeProduct</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>deactiveProduct</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>deleteProductByAccount</role-name>
  <group-name>user</group-name>
</security-role-mapping>
```

```
<security-role-mapping>
  <role-name>addCategory</role-name>
  <group-name>manager</group-name>
</security-role-mapping>
<!--MOZ-->
<security-role-mapping>
  <role-name>getProductByIdAndLogin</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>editProduct</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>makeOrder</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>makeOrderPayment</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>cancelOrder</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>addShippingMethod</role-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>activateShippingMethod</role-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>deactivateShippingMethod</role-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>editShippingMethod</role-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findShippingMethod</role-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>setShippingMethodActive</role-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>setOrderStatus</role-name>
  <group-name>user</group-name>
```

```

</security-role-mapping>
<security-role-mapping>
  <role-name>getAllOrders</role-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAllOrdersByAccountAsSeller</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAllOrdersByAccountAsBuyer</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAllOrdersByDateAndPrice</role-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAllMethods</role-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAllOrders</role-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAllByBuyer</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAllBySeller</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAllOrderStatus</role-name>
  <group-name>user</group-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getOrder1ById</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getOrderStatusById</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findUnit</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<!--facades-->

```

```
<security-role-mapping>
  <role-name>findAll</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>edit</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findAccount</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findByLogin</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
  <group-name>virtual</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findByAccountAndAccessLevel</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>removeAccountAlevel</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findArchivalPasswordByAccountId</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findRange</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>count</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findProduct</role-name>
  <group-name>manager</group-name>
```



```
<group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>find</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findAccessLevel</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findOrderByBuyerLogin</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findByLoginForProduct</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findOrderBySellerLogin</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>createProduct</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>editProduct</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>removeProduct</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>editCategory</role-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>editAccount</role-name>
  <group-name>user</group-name>
  <group-name>manager</group-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>addCategory</role-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findUnit</role-name>
```

```

    <group-name>user</group-name>
    <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>getOrder1ById</role-name>
    <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>getOrderStatusById</role-name>
    <group-name>user</group-name>
</security-role-mapping>

```

Wycinek kodu 2. Kod klasy odpowiedzialnej za tworzenie dziennika zdarzeń.

```

package pl.lodz.p.it.ssbd2018.ssbd01.tools;

@Interceptor
public class LoggerInterceptor {

    @Resource
    private SessionContext sessionctx;

    private static final Logger logger = LoggerFactory.getLogger
(LoggerInterceptor.class);

    @AroundInvoke
    public Object logMethodInvocation(InvocationContext context) throws
Exception {

        String className = context.getTarget().getClass().
getCanonicalName();
        String methodName = context.getMethod().getName();
        String user = sessionctx.getCallerPrincipal().getName();

        StringBuilder parameters = new StringBuilder();
        if (context.getParameters() != null) {
            for (Object param : context.getParameters()) {
                if (parameters.length() > 0) {
                    parameters.append(", ");
                }
                parameters.append(param);
            }
        }

        logger.info("{}.{}}({}) zostaa wywoana przez uytkownika {}",
            className, methodName, parameters, user
        );

        Object result;
        try {

```

```

        result = context.proceed();
    } catch (Exception ex) {
        StringBuilder causes = new StringBuilder();

        Throwable cause = ex.getCause();
        while (cause != null) {
            if (causes.length() > 0) {
                causes.append(", ");
            }
            causes.append(cause);
            cause = cause.getCause();
        }

        logger.error("{}.{}}({}) wywołana przez użytkownika {} rzuciła następujący wyjątek {}: {}. Causes: [{}]",
            className, methodName, parameters, user, ex, ex.
            getLocalizedMessage(), causes
        );

        throw ex;
    }

    String resultString = context.getMethod().getReturnType().equals
(Void.TYPE) ? "void" : Objects.toString(result);
    logger.info("{}.{}}({}) wywołana przez użytkownika {} zwróciła {}",
        className, methodName, parameters, user, resultString
    );

    return result;
}
}

```

Wycinek kodu 3. Fragment kodu klasy realizujący funkcjonalność dziennika zdarzeń.

```

package pl.lodz.p.it.ssbd2018.ssbd01.mok.managers;

import javax.interceptor.Interceptors;

@TransactionAttribute(TransactionAttributeType.REQUIRES_NEW)
@Stateless
@Interceptors(LoggerInterceptor.class)
public class AccountManager implements AccountManagerLocal {
    ...
}

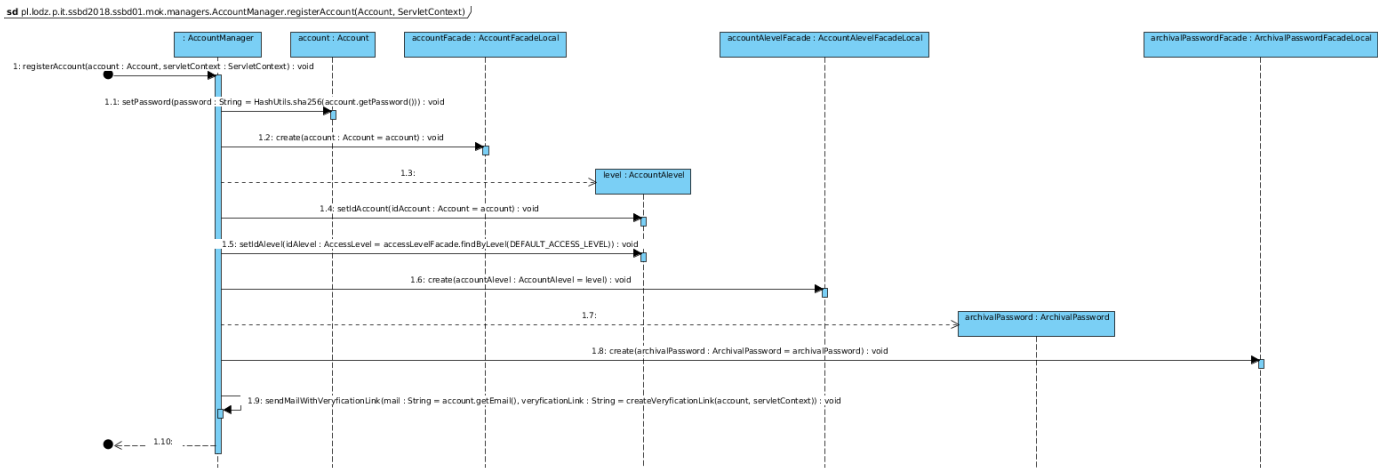
```

## 11 Identyfikacja transakcji aplikacyjnych 6 pkt

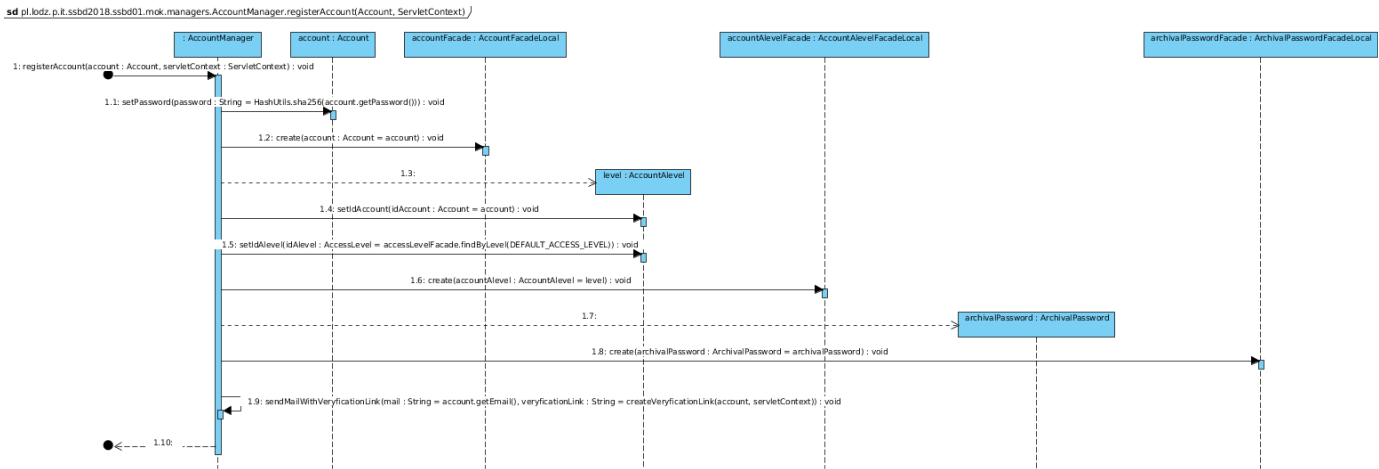
Z powodu zastosowanej architektury bezstanowej (@Stateless) nie mamy możliwości zapisu rozpoczęcia i zakończenia transakcji do dziennika zdarzeń.

Poniżej przedstawione są diagramy sekwencji poszczególnych przypadków użycia.

MOK.1 Zarejestruj:

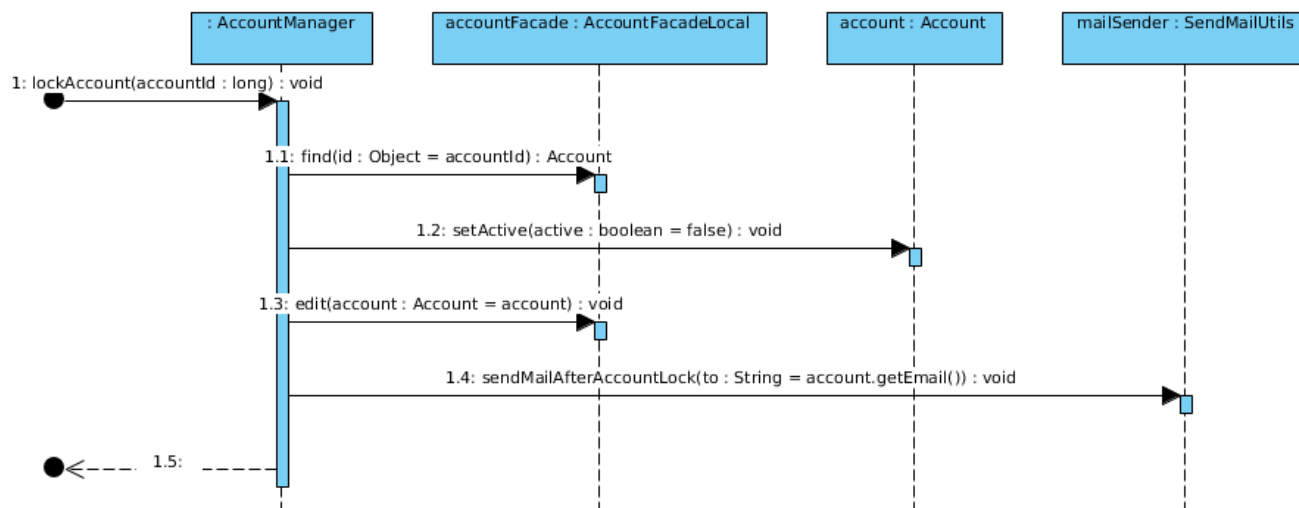


MOK.2 Utwórz konto:



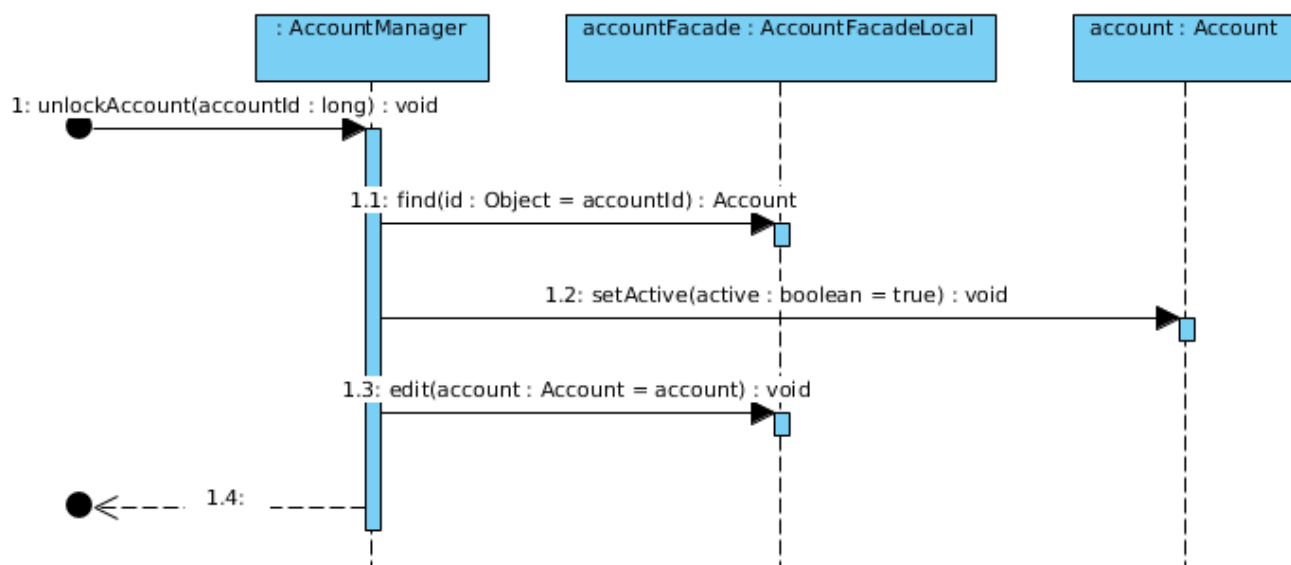
MOK.3 Zablokuj konto:

sd pl.lodz.p.it.ssb2018.ssb01.mok.managers.AccountManager.lockAccount(long)



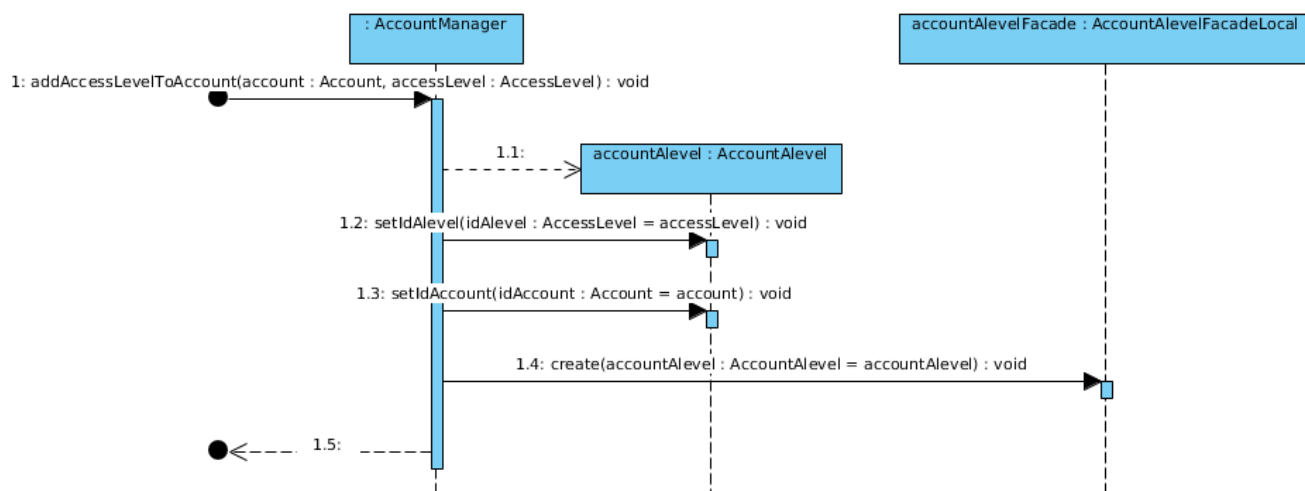
#### MOK.4 Odblokuj konto:

sd pl.lodz.p.it.ssb2018.ssb01.mok.managers.AccountManager.unlockAccount(long)



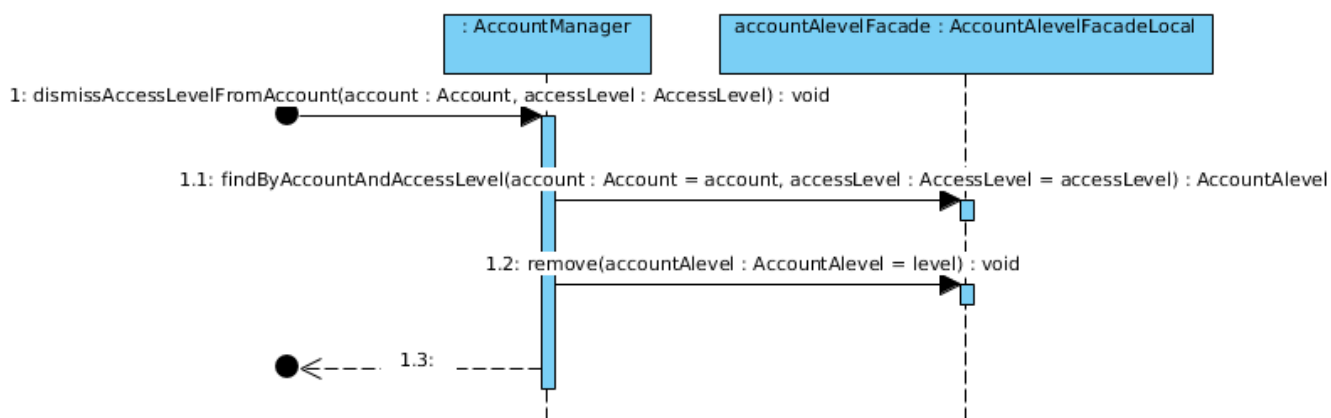
#### MOK.5 Dołącz poziom dostępu do konta:

sd pl.lodz.p.it.ssb2018.ssb01.mok.managers.AccountManager.addAccessLevelToAccount(Account, AccessLevel)



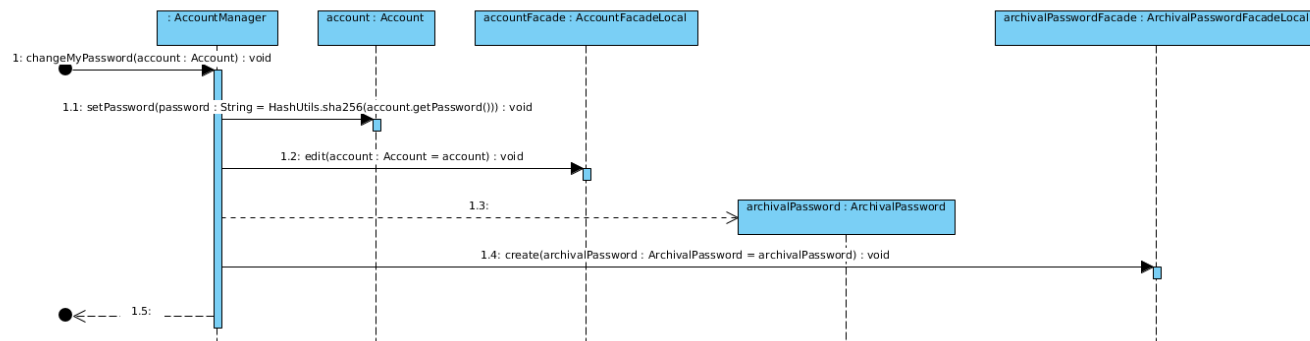
#### MOK.6 Odłącz poziom dostępu od konta:

sd pl.lodz.p.it.ssb2018.ssb01.mok.managers.AccountManager.dismissAccessLevelFromAccount(Account, AccessLevel)



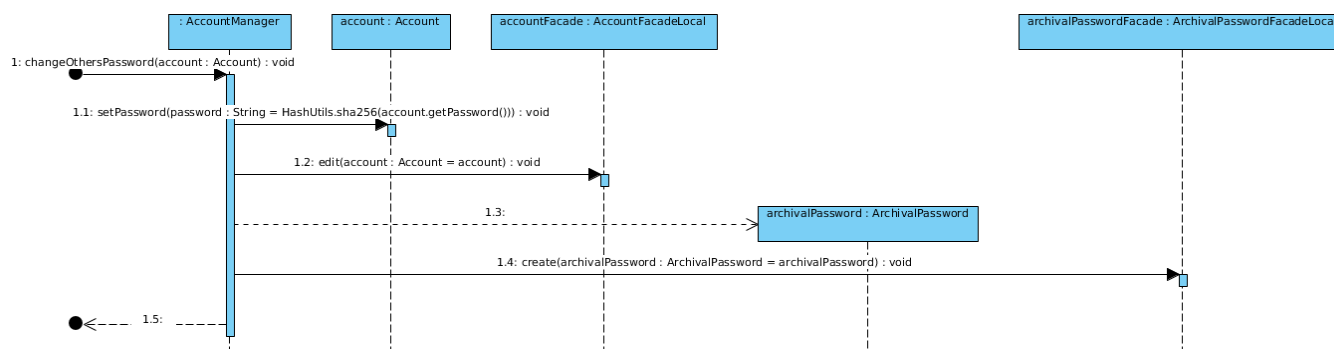
#### MOK.7 Zmień własne hasło:

sd pl.lodz.p.it.ssb2018.ssb01.mok.managers.AccountManager.changeMyPassword(Account)



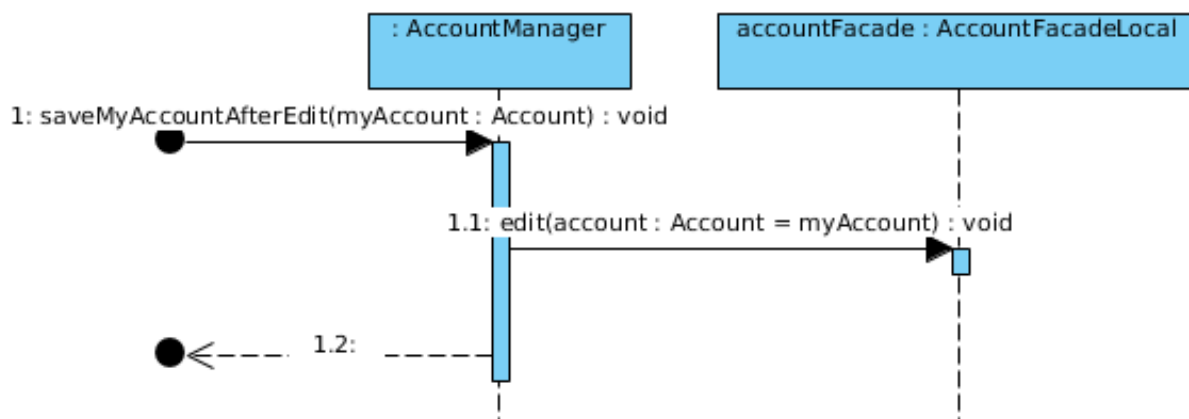
#### MOK.8 Zmień hasło innego użytkownika:

sd pl.lodz.p.it.ssb2018.ssb01.mok.managers.AccountManager.changeOthersPassword(Account)



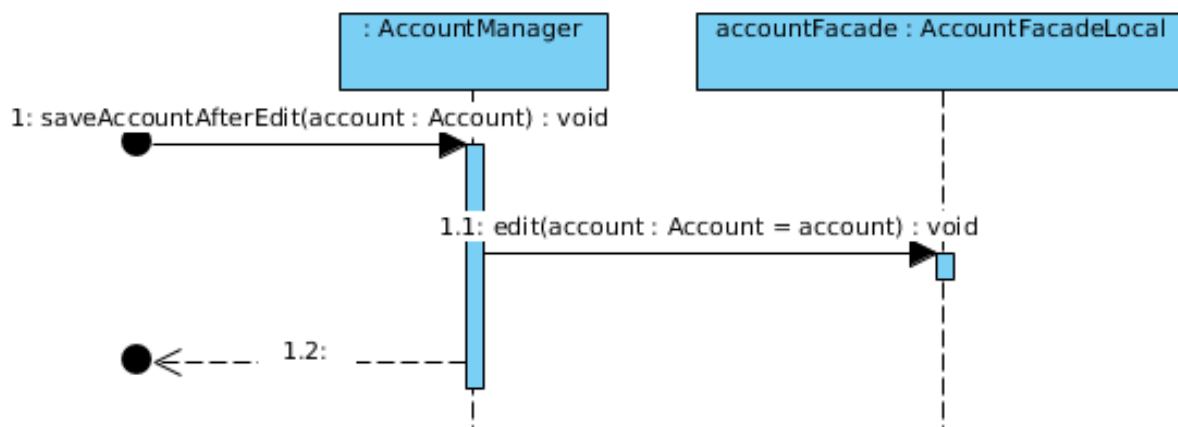
#### MOK.9 Edytuj dane własnego konta:

sd pl.lodz.p.it.ssb2018.ssb01.mok.managers.AccountManager.saveMyAccountAfterEdit(Account)



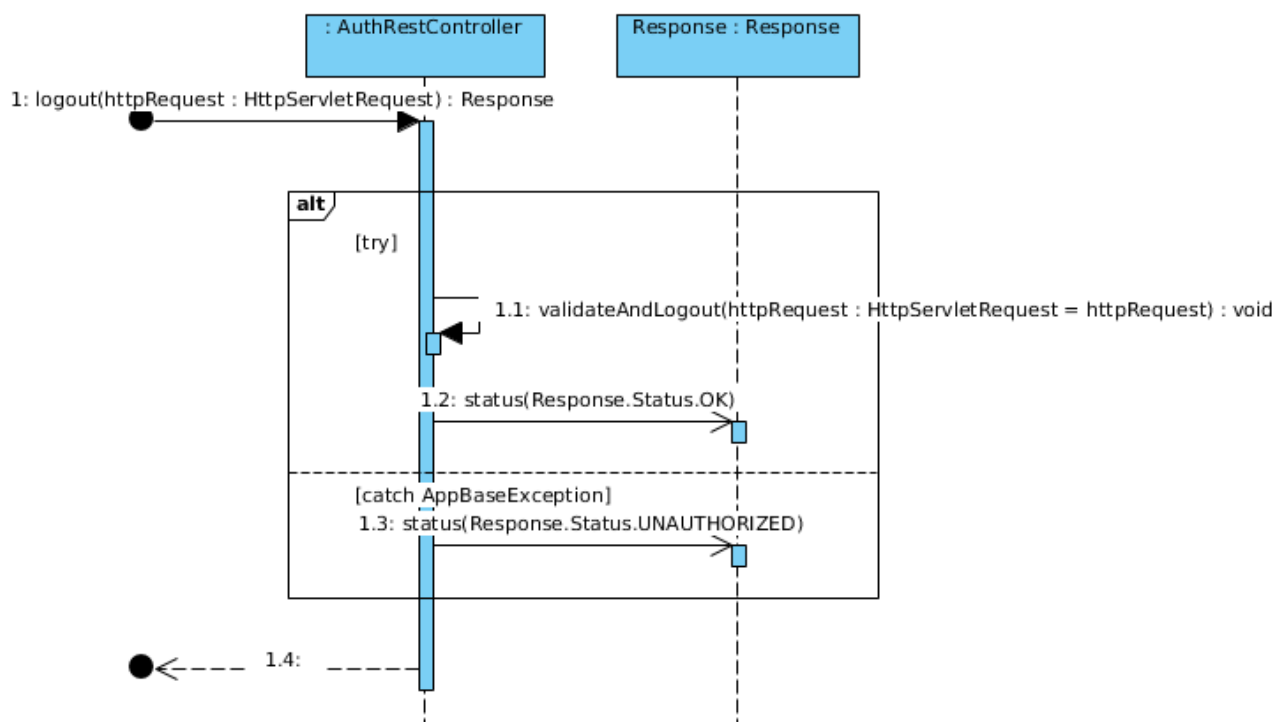
#### MOK.10 Edytuj dane konta innego użytkownika:

sd pl.lodz.p.it.ssb2018.ssb01.mok.managers.AccountManager.saveAccountAfterEdit(Account)



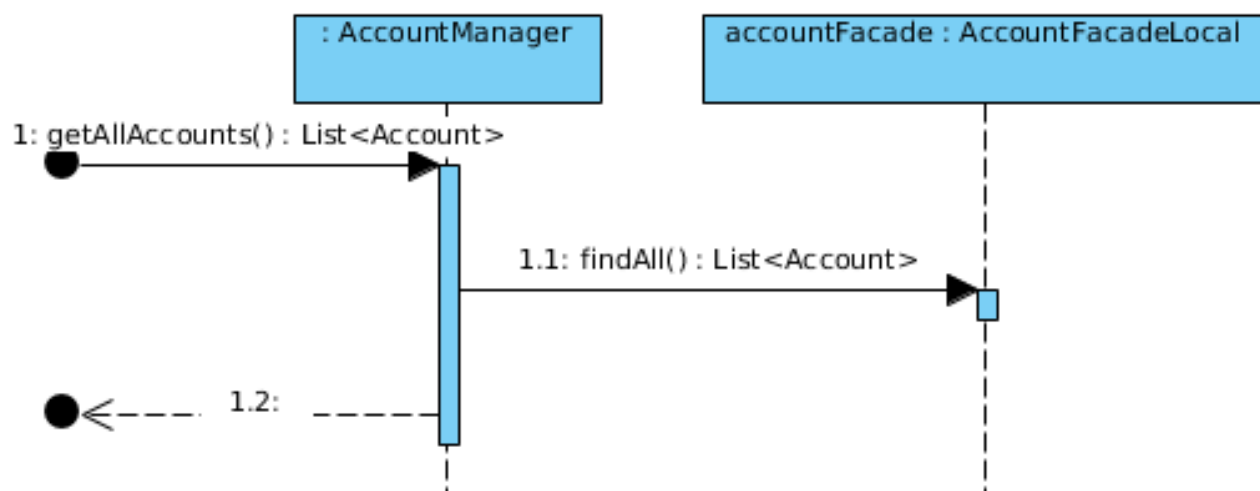
### MOK.11 Wyloguj:

**sd** pl.lodz.p.it.ssbd2018.ssbd01.mok.rest.AuthRestController.logout(HttpServletRequest)



### MOK.12 Przeglądaj listę kont:

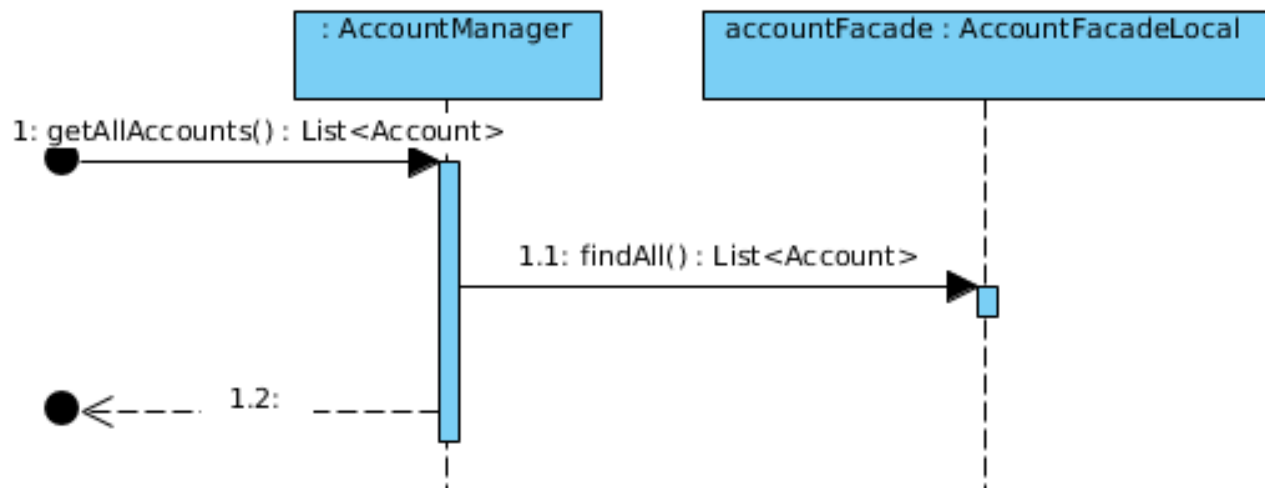
**sd** pl.lodz.p.it.ssbd2018.ssbd01.mok.managers.AccountManager.getAllAccounts()



MOK.13 oraz MOK.14 bazują na transakcji MOK.12, a ich dalsze przetwarzanie leży po stronie widoku.

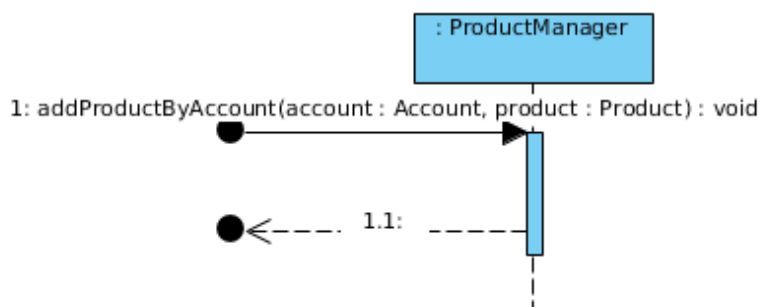


**sd** pl.lodz.p.it.ssbd2018.ssbd01.mok.managers.AccountManager.getAllAccounts()



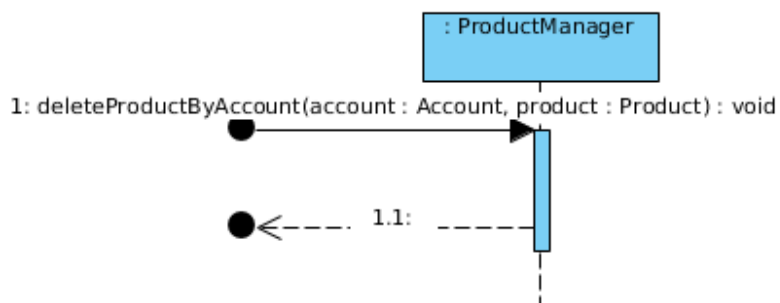
**MOP.1** Dodaj produkt:

**sd** pl.lodz.p.it.ssbd2018.ssbd01.mop.managers.ProductManager.addProductByAccount(Account, Product)



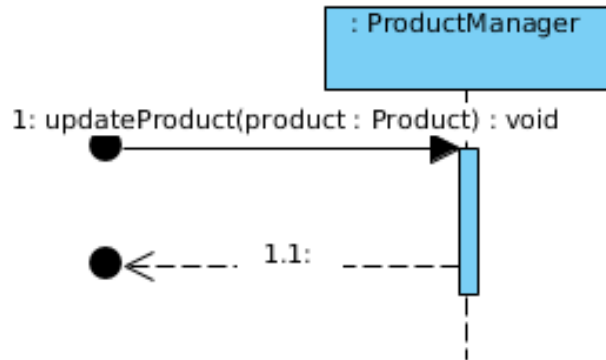
**MOP.2** Usuń produkt:

**sd** pl.lodz.p.it.ssbd2018.ssbd01.mop.managers.ProductManager.deleteProductByAccount(Account, Product)



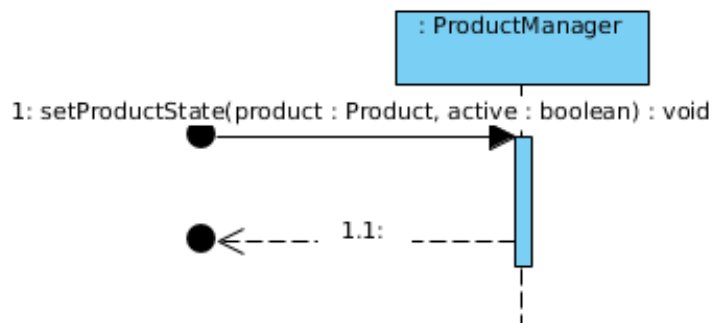
**MOP.3** Edytuj produkt:

**sd** pl.lodz.p.it.ssbd2018.ssbd01.mop.managers.ProductManager.updateProduct(Product)



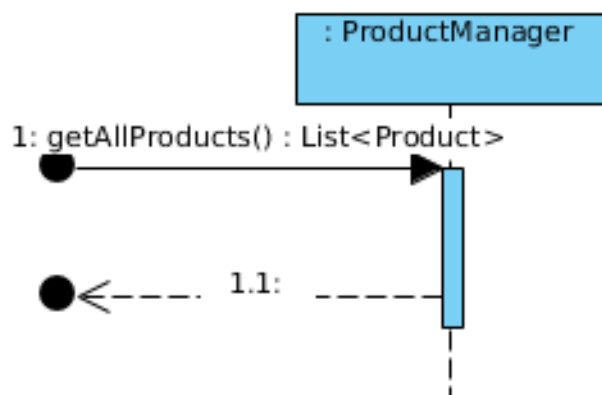
**MOP.4** Zmień stan produktu (aktywny, nieaktywny):

**sd** pl.lodz.p.it.ssbd2018.ssbd01.mop.managers.ProductManager.setProductState(Product, boolean)



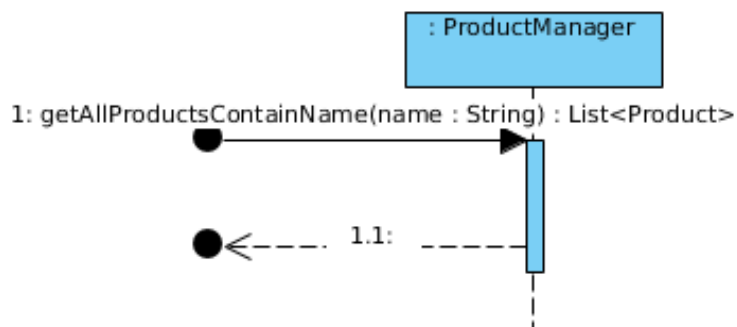
**MOP.5** Przeglądaj listę produktów:

**sd** pl.lodz.p.it.ssbd2018.ssbd01.mop.managers.ProductManager.getAllProducts()



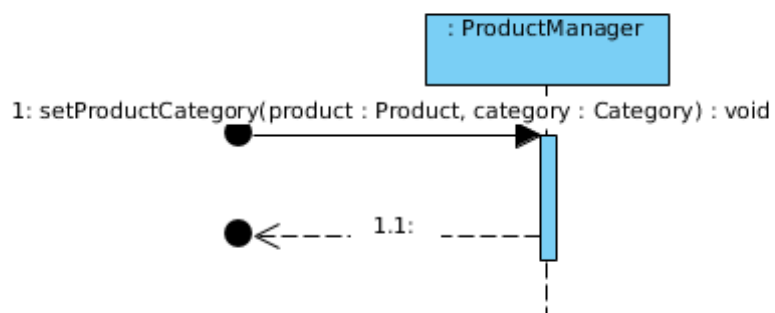
**MOP.6** Filtruj listę produktów:

**sd** pl.lodz.p.it.ssbd2018.ssbd01.mop.managers.ProductManager.getAllProductsContainName(String)



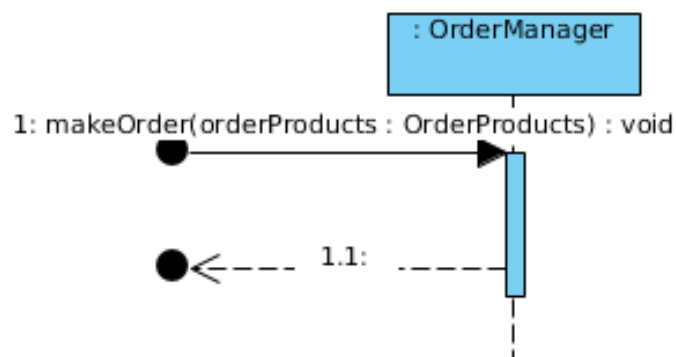
**MOP.7** Edytuj kategorie produktów (dodawanie nowych, aktywacja/dezaktywacja):

**sd** pl.lodz.p.it.ssbd2018.ssbd01.mop.managers.ProductManager.setProductCategory(Product, Category)



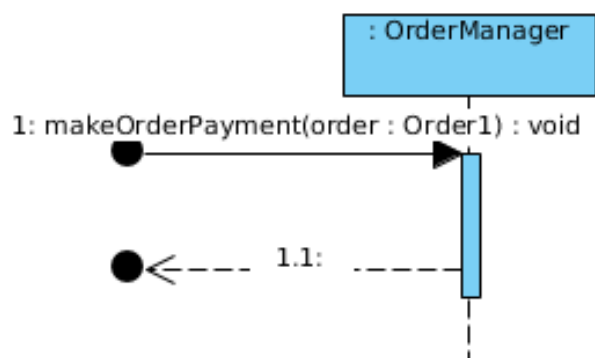
**MOZ.1** Dokonaj zakupu:

**sd** pl.lodz.p.it.ssbd2018.ssbd01.moz.managers.OrderManager.makeOrder(OrderProducts)



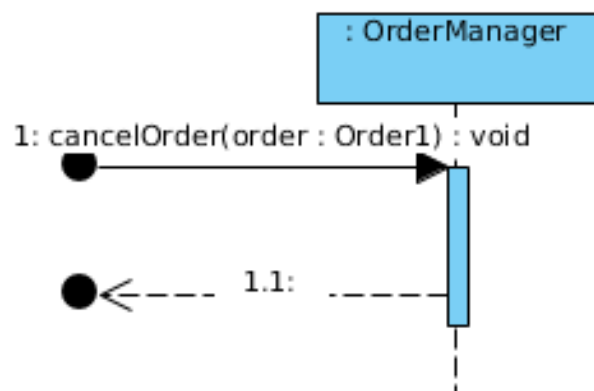
**MOZ.2** Opłać zamówienie:

**sd** pl.lodz.p.it.ssbd2018.ssbd01.moz.managers.OrderManager.makeOrderPayment(Order1)



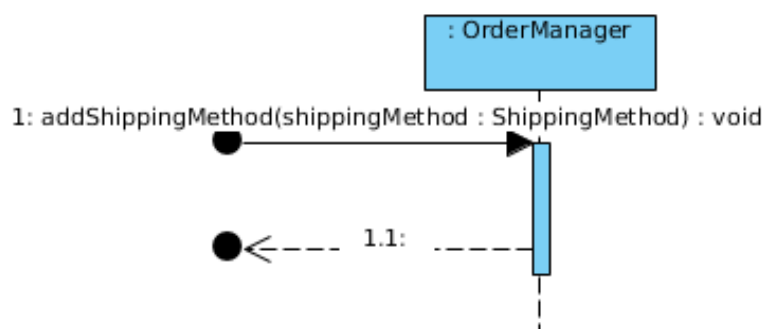
**MOZ.3** Anuluj zamówienie:

**sd** pl.lodz.p.it.ssbd2018.ssbd01.moz.managers.OrderManager.cancelOrder(Order1)



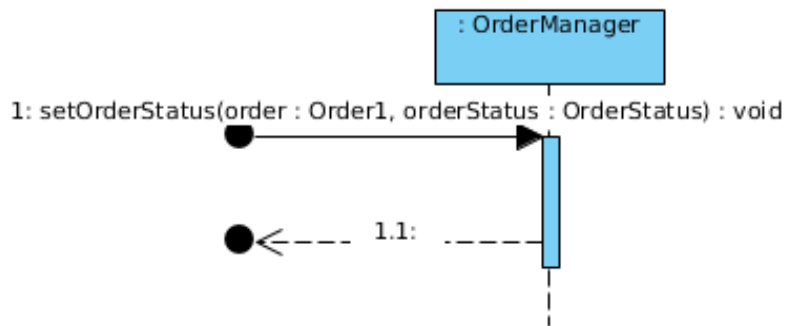
**MOZ.4** Edytuj metody wysyłki (dodawanie nowych, aktywacja/dezaktywacja):

**sd** pl.lodz.p.it.ssbd2018.ssbd01.moz.managers.OrderManager.addShippingMethod(ShippingMethod)



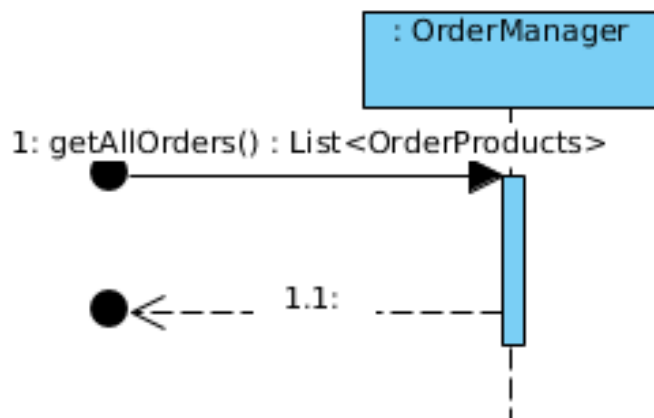
**MOZ.5** Zmień status zamówienia:

**sd** pl.lodz.p.it.ssbd2018.ssbd01.moz.managers.OrderManager.setOrderStatus(Order1, OrderStatus)



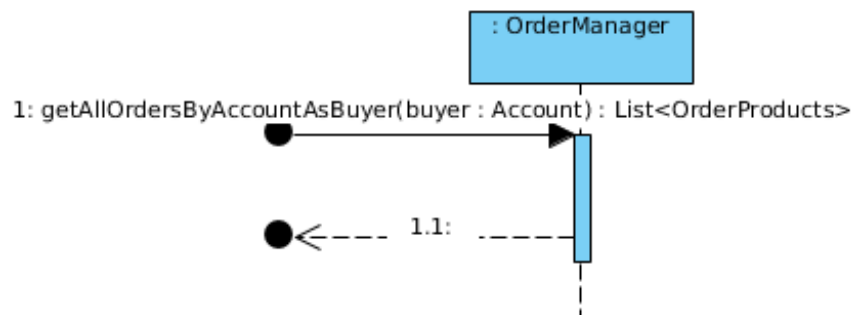
**MOZ.6** Przeglądaj wszystkie zamówienia:

**sd** pl.lodz.p.it.ssbd2018.ssbd01.moz.managers.OrderManager.getAllOrders()

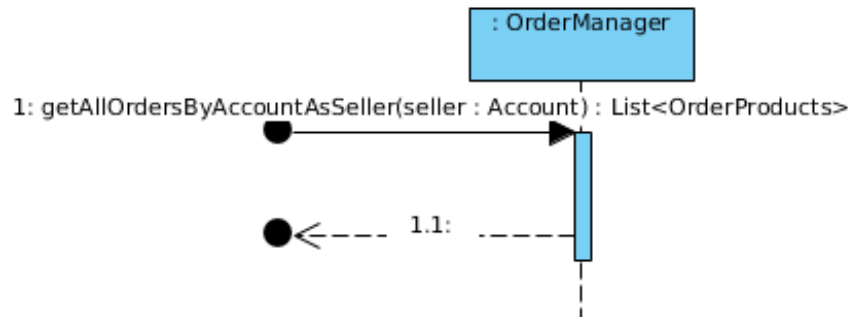


**MOZ.7** Przeglądaj listę własnych zamówień:

**sd** pl.lodz.p.it.ssbd2018.ssbd01.moz.managers.OrderManager.getAllOrdersByAccountAsBuyer(Account)

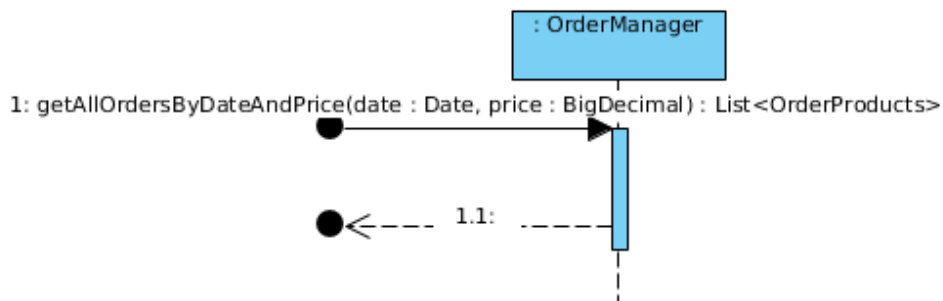


**sd** pl.lodz.p.it.ssbd2018.ssbd01.moz.managers.OrderManager.getAllOrdersByAccountAsSeller(Account)



**MOZ.8** Filtruj listę z zamówieniami:

**sd** pl.lodz.p.it.ssbd2018.ssbd01.moz.managers.OrderManager.getAllOrdersByDateAndPrice(Date, BigDecimal)



**Wczytanie danych konta do edycji:**

```
@Override
@RolesAllowed("getMyAccountByLogin")
public Account getMyAccountByLogin(String login) throws AppBaseException {
    return accountFacade.findByLogin(login);
}
```

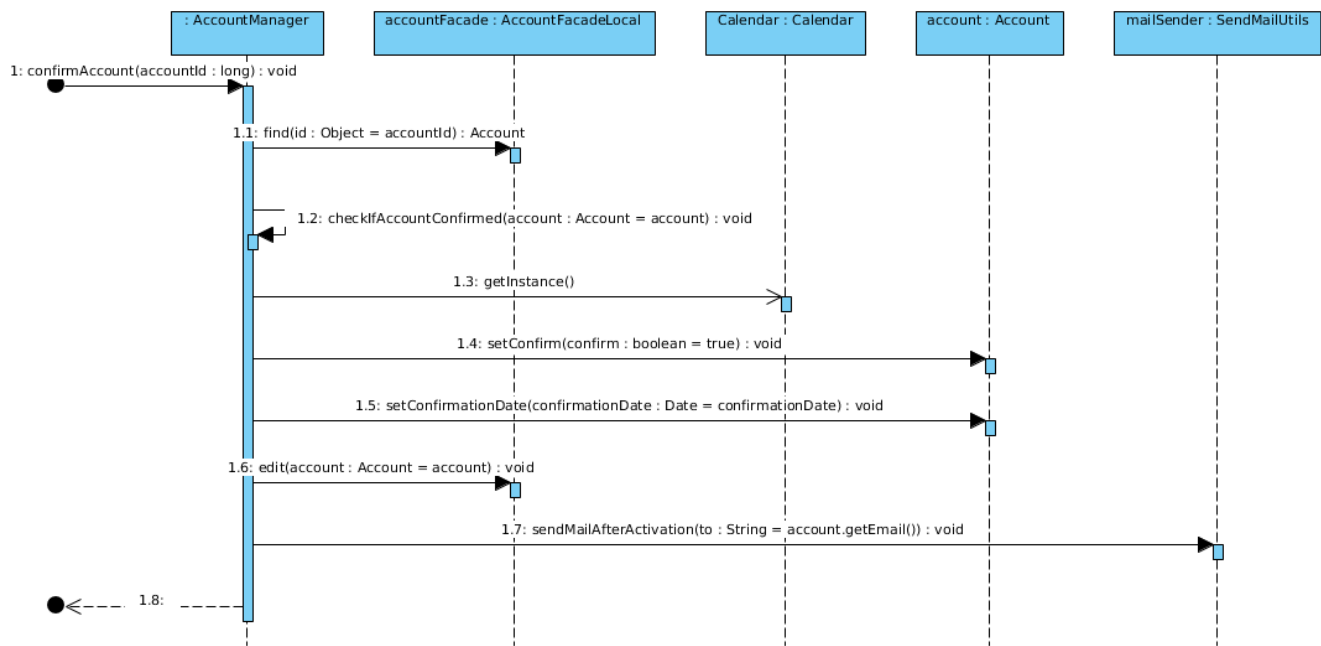
**Zapisanie konta po edycji:**

```
@Override
@RolesAllowed("saveAccountAfterEdit")
public void saveAccountAfterEdit(Account account) throws AppBaseException {
    accountFacade.edit(account);
}
```

**DODATKOWE DIAGRAMY SEKWENCJI:**

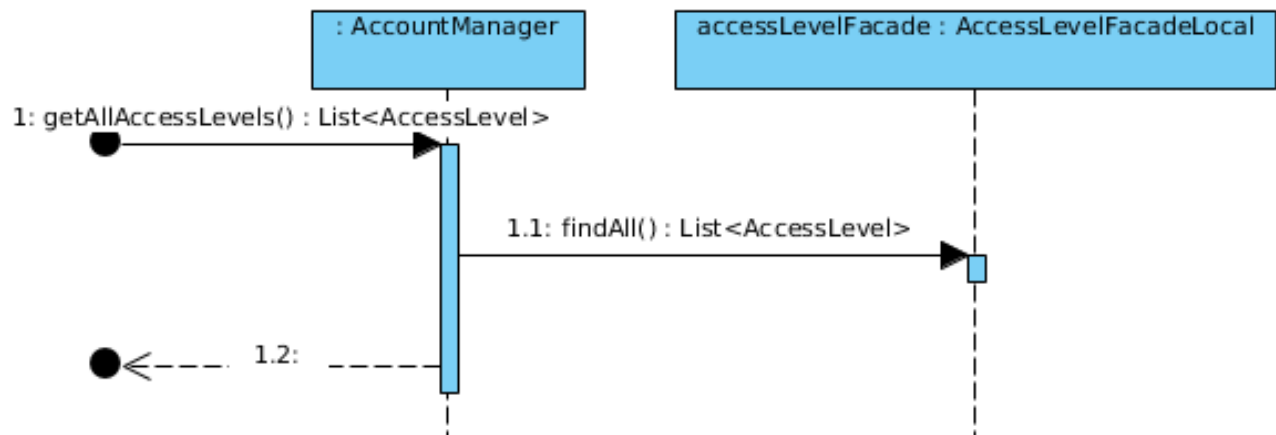
**Potwierdzenie konta (aktywacja):**

sd pl.lodz.p.it.ssb2018.ssb201.mok.managers.AccountManager.confirmAccount(long)



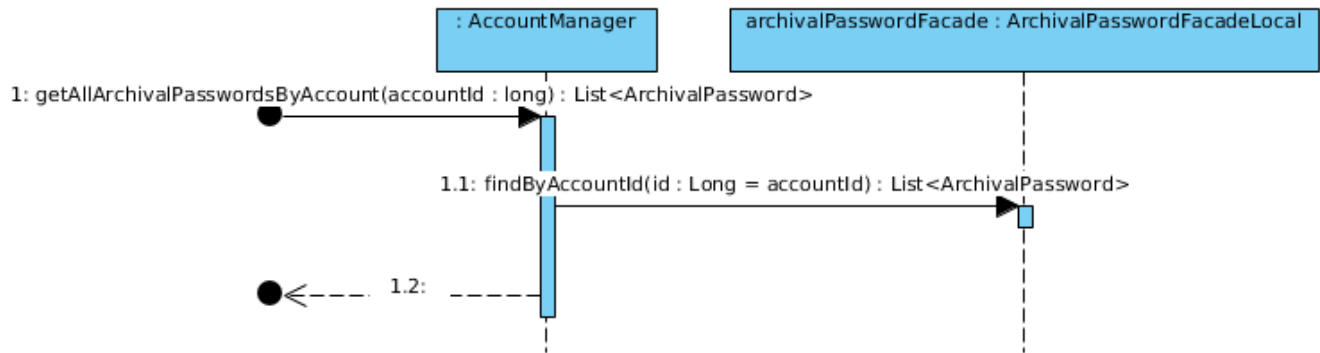
Pobranie wszystkich poziomów dostępu:

sd pl.lodz.p.it.ssb2018.ssb201.mok.managers.AccountManager.getAllAccessLevels()



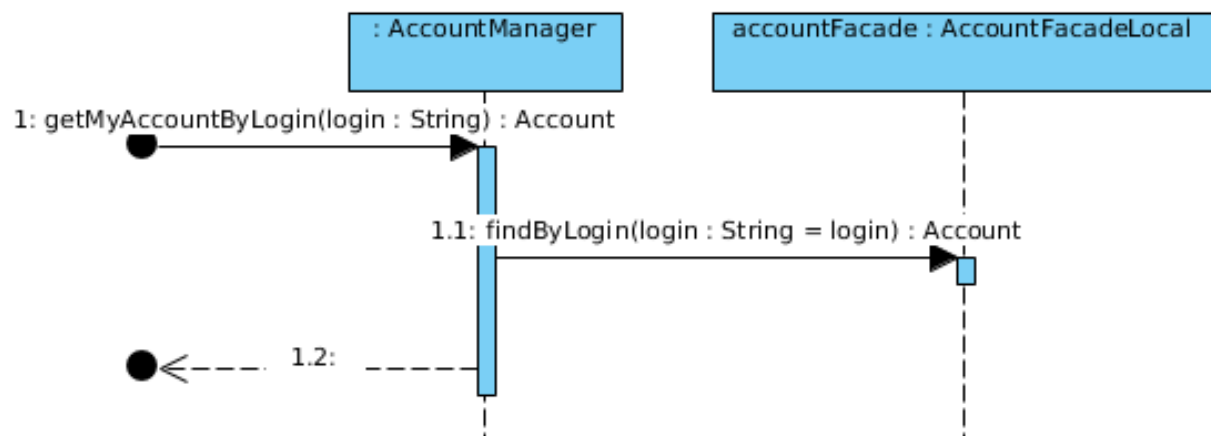
Pobranie wszystkich archiwalnych haseł konta:

**sd** pl.lodz.p.it.ssbd2018.ssbd01.mok.managers.AccountManager.getAllArchivalPasswordsByAccount(long)



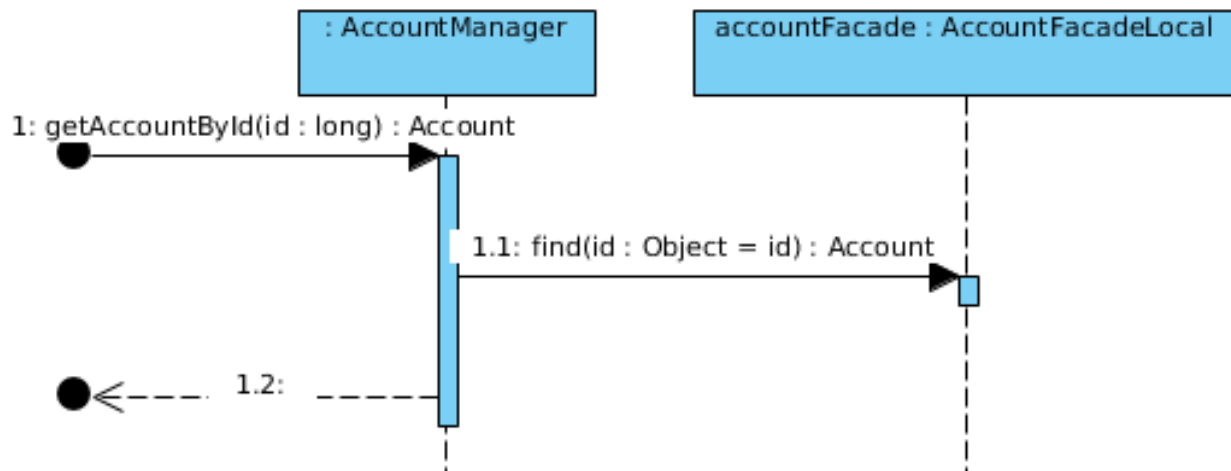
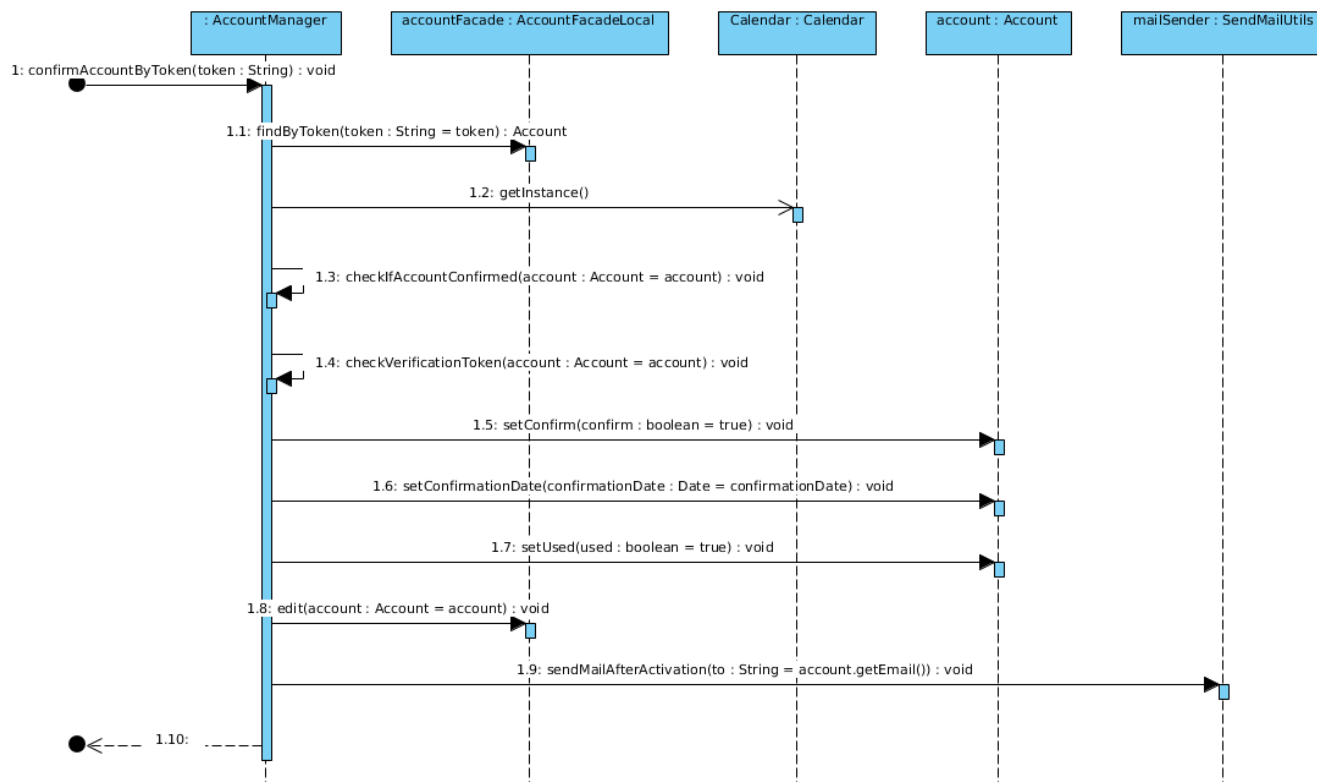
Pobranie konta za pomocą loginu:

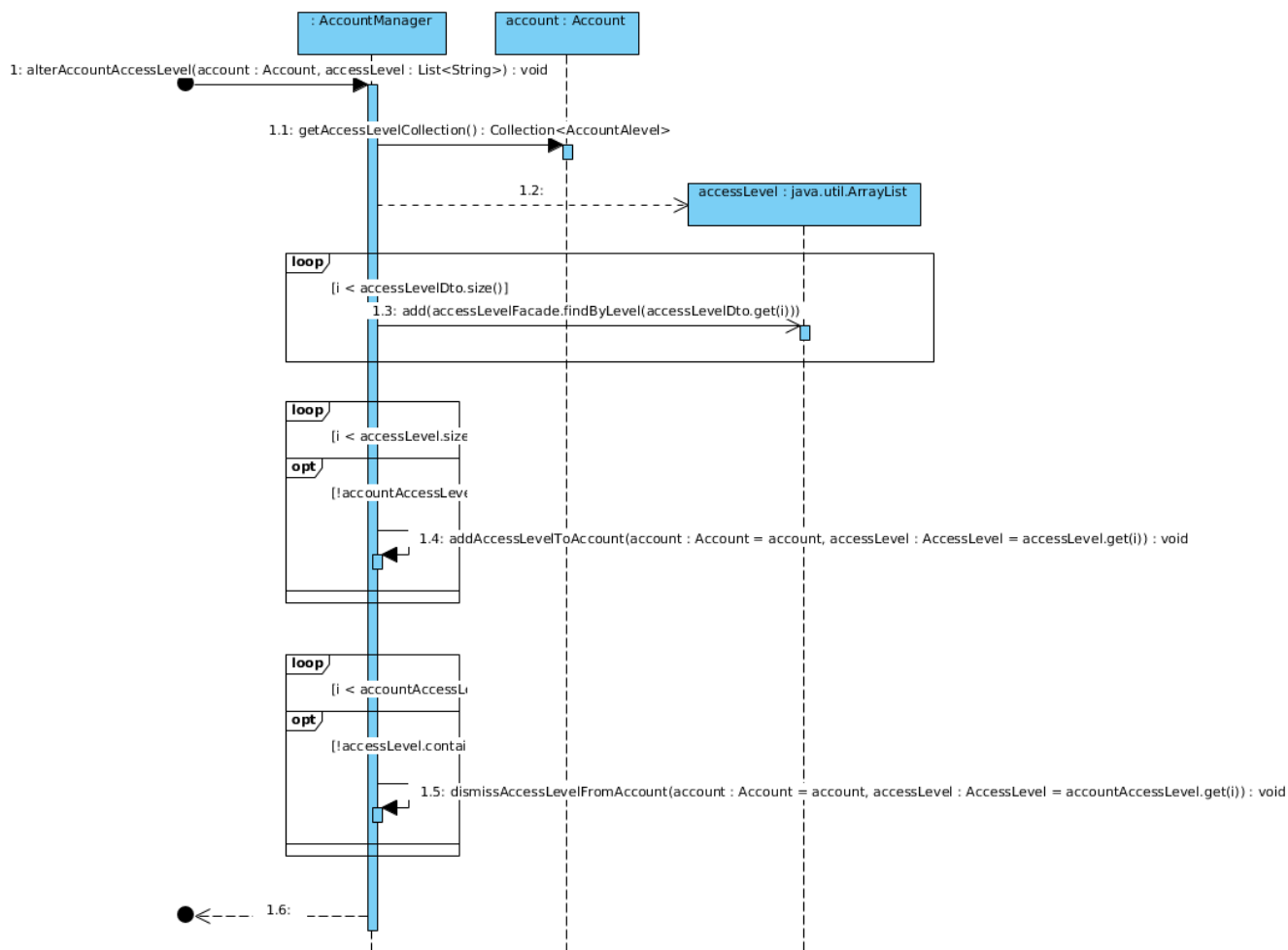
**sd** pl.lodz.p.it.ssbd2018.ssbd01.mok.managers.AccountManager.getMyAccountByLogin(String)



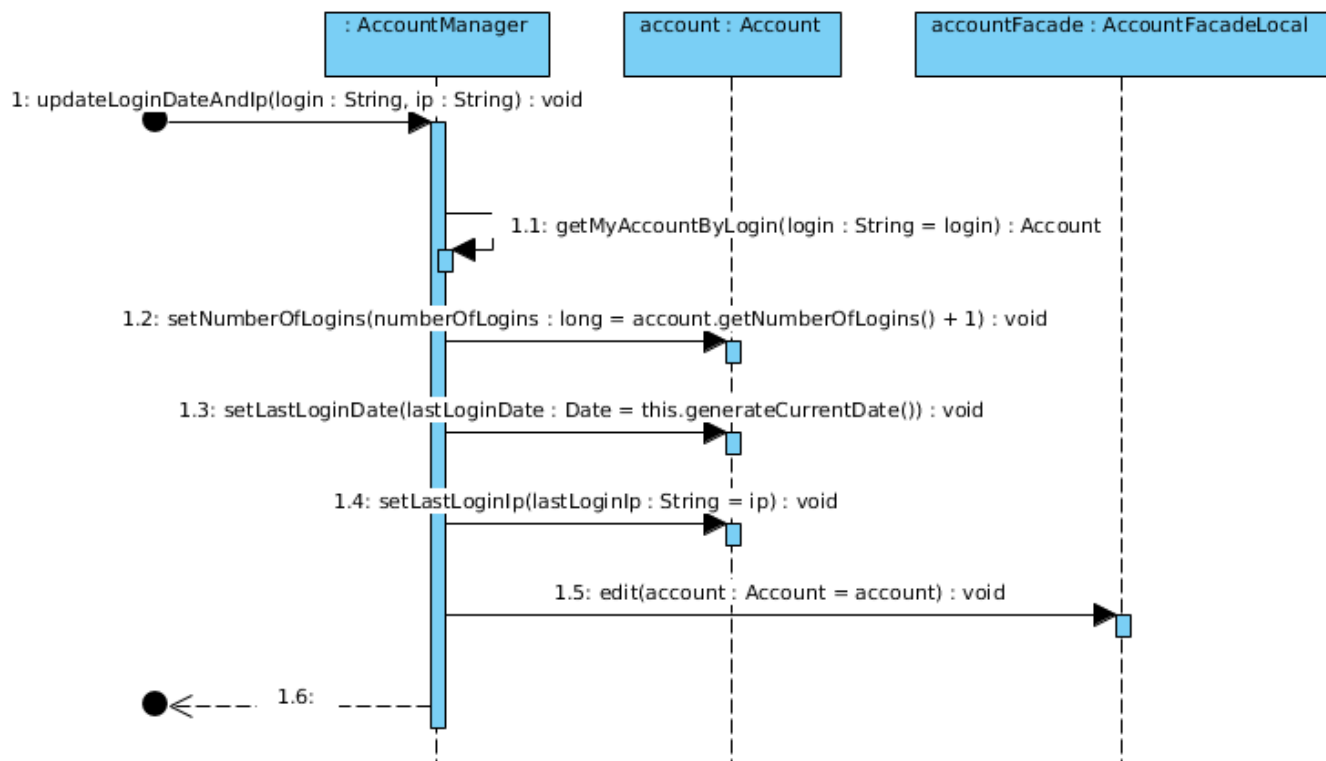
Pobranie konta za pomocą nr ID:



**Potwierdzenie konta poprzez token:****Zmiana poziomu dostępu konta:**



Zaktualizowanie daty ostatniego logowania i nr IP:



## 12 Zgłaszane wyjątki 4 pkt

Poniższe tabele przedstawiają zbiór wyjątków biznesowych oraz systemowych które zostały zaimplementowane w naszym systemie.

Wyjątki biznesowe					
Nazwa wyjątku	Miejsca zgłoszenia	Możliwe przyczyny wystąpienia	Obsługa w warstwie logiki	Obsługa w warstwie widoku	Czy prowadzi do odwołania transakcji
AppBaseException	Uogólnienie dla wszystkich poniższych wyjątków				
MokBaseException	Uogólnienie dla wyjątków:  <i>AccountAlevelException</i> , <i>AccountException</i> , <i>MailException</i>				
AccountAlevelException	<i>AccountAlevelFacade</i> : tworzenie poziomu dostępu	naruszenie ograniczeń podczas tworzenia poziomu dostępu	brak	<i>constraint_violation</i>	tak
AccountAlevelExistsException	Roszerza <i>AccountAlevelException</i>  <i>AccountAlevelFacade</i> : tworzenie poziomu dostępu	poziom dostępu już istnieje	brak	<i>access_level_exists_exception</i>	tak
AccessLevelExistsException	<i>AccessLevelFacade</i> : szukanie poziomu dostępu	podany poziom dostępu nie istnieje	brak	<i>access_level_not_found_exception</i>	nie
AccountException	<i>AccountFacade</i> :	zostały naruszone	brak	<i>constraint_violation</i>	tak

	tworzenie, edycja konta	ograniczenia nałożone na encję <i>Account</i> (obsługa wyjątku <i>ConstraintViolationException</i> )			
	<i>AccountManager</i> : zmiana własnego hasła, zmiana hasła innego konta	obsługa wyjątku <i>AppBaseException</i>	brak	<i>unknown_exception</i>	tak
	<i>AccountManager</i> : blokowanie konta, odblokowanie konta	obsługa wyjątku <i>AppBaseException</i>	brak	<i>unknow_error</i>	tak
AccountNotFoundException	<i>AccountManager</i> : blokowanie konta, odblokowanie konta	wprowadzenie jako parametru metody niesitniającego identyfikatora konta (obsługa wyjątku <i>NullPointerException</i> )	brak	<i>wrong_account_id_error</i>	tak
AccountOptimisticException	<i>Account</i> : ustalanie wersji encji	obsługa blokady optymistycznej	brak	<i>account_optimistic_lock_exception</i>	tak
	<i>AccountManager</i> : zapis konta po edycji, blokowanie konta, odblokowanie konta, potwierdzenie konta	obsługa wyjątku <i>OptimisticLockException</i>	brak	<i>account_optimistic_error</i>	tak
EmailNotUniqueException	<i>AccountFacade</i> : tworzenie, edycja, konta	podany adres email nie jest unikalny (obsługa wyjątku <i>PersistenceException</i> )	brak	<i>email_unique</i>	nie
AccountWasConfirmed	<i>AccountManager</i> : sprawdzenie czy konto jest potwierdzone	konto zostało już potwierdzone /aktywowane	brak	<i>account_was_confirmed_exception</i>	nie
AccountWithNoAccessLevelException	<i>AccountManager</i> : zmiana poziomów dostępu	brak zdefiniowanych poziomów dostępu do zmiany	brak	<i>no_access_level_error</i>	tak
ConstraintException	<i>AccountRestController</i> : walidacja ograniczeń	podane dane niezgodne z ograniczeniami	brak	<i>constraint_error</i>	tak
LoginNotUniqueException	<i>AccountFacade</i> : tworzenie, edycja, konta	podany login nie jest unikalny (obsługa wyjątku <i>PersistenceException</i> )	brak	<i>login_unique</i>	nie
PasswordDifferentException	<b>NIEUŻYWANY</b>				
PasswordNotMatch	<i>AccountManager</i> : zmiana własnego hasła	porównanie skrótów nieodwracalnych dwóch haseł daje negatywny wynik	brak	<i>password_not_match_error</i>	nie
PasswordSameAsArchivalPasswordException	<i>AccountRestController</i> : porównanie ze starymi hasłami danego konta	hasło było już wykorzystane wcześniej	brak	<i>password_same_as_archival_error</i>	tak
PasswordTooShortException	<i>AccountRestController</i> : sprawdzenie długości podanego hasła	podane hasło jest za krótkie - mniej niż 8 znaków	brak	<i>password_to_short</i>	tak
PhoneNotUniqueException	<i>AccountFacade</i> : tworzenie, edycja, konta	podany numer telefonu nie jest unikalny (obsługa wyjątku <i>PersistenceException</i> )	brak	<i>phone_unique</i>	nie
VerificationTokenException	<b>NIEUŻYWANY</b>				
TokenExpiredException	<i>AccountManager</i> : sprawdzenie ważności tokena	przyznany token jest już nieważny	brak	<i>token_expired_exception</i>	tak
TokenNotUniqueException	<i>AccountFacade</i> : sprawdzenie unikalności tokenu	token nieunikalny	brak	<i>token_unique</i>	tak

TokenUsedException	<i>AccountManager</i> : sprawdzenie czy token został już użyty	token użyty	brak	<i>token_used_exception</i>	tak
VerificationTokenNotFoundException	<b>NIEUŻYWANY</b>				
MailException	<b>NIEUŻYWANY</b>				
MailNotSentException	<b>NIEUŻYWANY</b>				
WrongMailAddressException	<b>NIEUŻYWANY</b>				
WrongTokenException	<i>AccountRestController</i> : walidacja tokenu	token niezgodny z oczekiwaniami	brak	<i>wrong_token</i>	tak

Wyjątki systemowe					
Nazwa wyjątku	Miejsca zgłoszenia	Możliwe przyczyny wystąpienia	Obsługa w warstwie logiki	Obsługa w warstwie widoku	Czy prowadzi do odwołania transakcji
EJBAccessExceptionHandler	<i>ApplicationConfig</i>	brak dostępu do bazy danych	brak	brak	nie dojdzie do jej inicjalizacji

## 13 Interfejs użytkownika 3 pkt

Interfejs użytkownika został utworzony przy pomocy platformy Angular.

Finalny kod szablonu widoku w aplikacji jest generowany przez przeglądarkę, a dane biznesowe są pobierane w odpowiedzi na poszczególne akcje użytkownika, w odróżnieniu od technologii JSF, gdzie dokument HTML jest w całości tworzony na serwerze i wysyłany do klienta.

Menu użytkownika to komponent istniejący przez cały czas działania aplikacji w przeglądarce użytkownika.

Jak każdy komponent platformy Angular składa się on z szablonu (pliku HTML, widoku) oraz klasy komponentu:

### Pełny kod szablonu HTML:

```
<div class="example-container mat-elevation-z8">

  <div class="example-header">
    <mat-form-field class='filter-form-field'>
      <input [(ngModel)]='filterValue' matInput (keyup)="
onFilterCriteriaChange($event.target.value)" placeholder="{{ 'MAT_TABLE.
FILTER' | translate }}">
    </mat-form-field>
  </div>

  <mat-table #table [dataSource]="dataSource" matSort >

    <ng-container matColumnDef="login">
      <mat-header-cell *matHeaderCellDef mat-sort-header> {{ 'ACCOUNT.
LOGIN' | translate }} </mat-header-cell>
      <mat-cell *matCellDef="let element"> {{element.login}} </mat-cell>
```

```

</ng-container>

<ng-container matColumnDef="lastLoginDate">
  <mat-header-cell *matHeaderCellDef mat-sort-header> {{ 'ACCOUNT.
LAST_LOGIN_DATE' | translate }} </mat-header-cell>
  <mat-cell *matCellDef="let element"> {{element.lastLoginDate}} </mat-
cell>
</ng-container>

<ng-container matColumnDef="lastLoginIp">
  <mat-header-cell *matHeaderCellDef mat-sort-header> {{ 'ACCOUNT.
LAST_LOGIN_IP' | translate }} </mat-header-cell>
  <mat-cell *matCellDef="let element"> {{element.lastLoginIp}} </mat-
cell>
</ng-container>

<ng-container matColumnDef="numberOfLogins">
  <mat-header-cell *matHeaderCellDef mat-sort-header> {{ 'ACCOUNT.
NUM_OF_LOGINS' | translate }} </mat-header-cell>
  <mat-cell *matCellDef="let element"> {{element.numberOfLogins}} <
/mat-cell>
</ng-container>

<ng-container matColumnDef="numberOfOrders">
  <mat-header-cell *matHeaderCellDef mat-sort-header> {{ 'ACCOUNT.
NUM_OF_ORDERS' | translate }} </mat-header-cell>
  <mat-cell *matCellDef="let element"> {{element.numberOfOrders}} <
/mat-cell>
</ng-container>

<ng-container matColumnDef="numberOfProducts">
  <mat-header-cell *matHeaderCellDef mat-sort-header> {{ 'ACCOUNT.
NUM_OF_PRODUCTS' | translate }} </mat-header-cell>
  <mat-cell *matCellDef="let element"> {{element.numberOfProducts}} <
/mat-cell>
</ng-container>

<ng-container matColumnDef="confirmAccount">
  <mat-header-cell class='button-cell' *matHeaderCellDef> {{ 'ACCOUNT.
CONFIRM' | translate }} </mat-header-cell>
  <mat-cell class='button-cell' *matCellDef="let element; let
rowIndex=index">
    <mat-checkbox (click)='onConfirmChange(element, rowIndex)'
[(ngModel)]= 'element.confirm' color='primary' [disabled]='element.
confirm'></mat-checkbox>
  </mat-cell>
</ng-container>

<ng-container matColumnDef="lockOrUnlockAccount">
  <mat-header-cell class='button-cell' *matHeaderCellDef> {{ 'ACCOUNT.
ACTIVE' | translate }} </mat-header-cell>
  <mat-cell class='button-cell' *matCellDef="let element; let
rowIndex=index">

```

```

        <mat-checkbox (change)='onActiveChange(element, rowIndex)'
[(ngModel)]= 'element.active' color='primary'></mat-checkbox>
    </mat-cell>
</ng-container>

    <ng-container matColumnDef="accessLevel">
        <mat-header-cell class='button-cell' *matHeaderCellDef> {{ 'ACCOUNT.
ACCESS_LEVELS' | translate }} </mat-header-cell>
        <mat-cell *matCellDef="let element; let rowIndex=index">
            <mat-select [(ngModel)]= 'element.accessLevels' placeholder="
Roles" (change)='onAccessLevelChange(element, rowIndex)' multiple>
                <mat-option *ngFor='let role of availableRoles' [value]
='role.value' >{{ role.displayedValue | translate }}</mat-option>
            </mat-select>
        </mat-cell>
    </ng-container>

    <mat-header-row *matHeaderRowDef="getDisplayedColumns()"></mat-header-
row>
    <mat-row *matRowDef="let row; let rowId=index; columns:
getDisplayedColumns();" [ngClass]='{ "highlited-row" : wasRowChanged
(rowId) }'></mat-row>
</mat-table>

    <mat-paginator #paginator
        [pageSize]="10"
        [pageSizeOptions]="[5, 10, 20]"
        [showFirstLastButtons]="true"
        (page)='onPaginatorPageChange()'>
    </mat-paginator>

</div>
<div class="buttons">
    <button mat-button color='primery'>{{ 'FORM.RETURN' | translate }} <
/button>
    <button mat-button mat-raised-button color='primary' [disabled]='!
wasSomeUserChanged()' (click)='submitAccounts()'>{{ 'FORM.SUBMIT' |
translate}}</button>
</div>

```

### Pełny kod klasy komponentu:

```

import { Component, ViewChild, OnInit, Output, EventEmitter } from
'@angular/core';
import { MatSidenav } from '@angular/material/sidenav';
import { Observable } from 'rxjs/Observable';
import { TranslateService } from '@ngx-translate/core';
import { SessionService } from '../../mok/common/session.service';
import { AuthUtilService } from '../../mok/common/auth-util.service';
import { AuthService } from '../../mok/common/auth.service';

```

```

import { AccountData } from '../../../mok/model/account-data';
import { NotificationService } from '../../../mok/common/notification.
service';
import { Route, Router } from '@angular/router';
import { LocationService } from '../../../mok/common/location.service';
import { Properties } from '../../../constsants';

@Component({
  selector: 'app-sidenav',
  templateUrl: './sidenav.component.html',
  styleUrls: ['./sidenav.component.css']
})
export class SidenavComponent implements OnInit {

  readonly AdminRole: string = Properties.AdminRole;
  readonly ManagerRole: string = Properties.ManagerRole;
  readonly UserRole: string = Properties.UserRole;
  readonly GuestRole: string = Properties.GuestRole;
  readonly VirtualRole: string = Properties.VirtualRole;

  userIdentity: AccountData = {};
  rolesStringified = '';

  displayedRouter: string;

  constructor(
    private sessionService: SessionService,
    private translateService: TranslateService,
    private authUtil: AuthUtilService,
    private authService: AuthService,
    private notificationService: NotificationService,
    private router: Router,
    private locationService: LocationService) {

  }

  ngOnInit() {
    this.sessionService.getMyIdentity().subscribe((data: AccountData) => {
      this.userIdentity = data;
    }, () => {
      this.userIdentity = {
        login: this.translateService.instant("GUEST"),
        roles: [this.GuestRole]
      };
    });
    this.locationService.currentRouter.subscribe(router => {
      if (router.length > 0) {
        this.translateService.get(router).subscribe(routerTranslation => {
          this.translateService.get('LOCATION.YOUR_LOCATION').subscribe
            (locationTranslation => {
              this.displayedRouter = locationTranslation + ': ' + routerTranslation;
            });
        });
      }
    });
  }

```



```

    } else {
    this.displayedRouter = '';
    }
    });
    this.updateLoginAndRoles();
    }

    onSidenavOpenedChange() {
    this.updateLoginAndRoles();
    }

    onLogoutClick() {
    this.authService.logout().subscribe(() => {
    this.updateLoginAndRoles();
    this.notificationService.displayTranslatedNotification('SUCCESS.LOGOUT');
    this.router.navigate(['/main']);
    });
    }

    updateLoginAndRoles() {
    this.sessionService.getMyIdentity().toPromise().then((data) => {
    this.userIdentity.login = data.login;
    this.userIdentity.roles = data.roles;
    this.rolesStringified = JSON.stringify(data.roles);
    }).catch((data) => {
    this.userIdentity.login = this.translateService.instant('GUEST');
    this.userIdentity.roles = [this.GuestRole]
    this.rolesStringified = JSON.stringify(this.userIdentity.roles);
    });
    }

    hasRole(role: string): boolean {
    return this.authUtil.hasRole(role, this.userIdentity);
    }

    isMainPage(router: string): boolean {
    if (router.length <= 0 || router === 'LOCATION.MAIN_PAGE') return true;
    else return false;
    }
    }

```

Mechanizm autoryzacji oparty jest o punkt dostępowy REST (ang. REST endpoint) `/myIdentity`, przy pomocy którego użytkownik pobiera informacje o swojej tożsamości, tzn. login oraz listę poziomów dostępu. W przypadku błędu, użytkownik jest identyfikowany jako Gość.

```

export class SidenavComponent implements OnInit {

    ...

```

```

ngOnInit() {
  this.sessionService.getMyIdentity().subscribe((data: AccountData) => {
    this.userIdentity = data;
  }, () => {
    this.userIdentity = {
      login: this.translateService.instant("GUEST"),
      roles: [this.GuestRole]
    };
  });

  ...

  ...

export class SessionService {

  readonly uri: string = environment.apiUrl + '/webresources/session';

  ...
  getMyIdentity(): Observable<AccountData> {
    return this.httpClient.get<AccountData>(this.uri + '/myIdentity');
  }

  ...

```

Na podstawie pobranych informacji wyświetlany jest login oraz poziomy dostępu użytkownika.

```

...

<h2 *ngIf='hasRole(UserRole) || hasRole(AdminRole) || hasRole(ManagerRole)
|| hasRole(VirtualRole)' class="example-h2">{{ "HOME.WELCOME" | translate
}}, {{ userIdentity.login }}</h2>
<p *ngIf='hasRole(UserRole) || hasRole(AdminRole) || hasRole(ManagerRole)
|| hasRole(VirtualRole)' class="example-p">{{ "HOME.USER_ROLES" |
translate }}: {{ rolesStringified | translate }}</p>

...

```

Pobrana lista poziomów dostępu jest wykorzystywana również do określenia, które opcje menu powinny zostać wyświetlone w widoku użytkownika.

*Przykład pojedynczej opcji w menu*

```
...  
  
<button mat-button routerLink='/myAccount' *ngIf='hasRole(UserRole) ||  
hasRole(AdminRole) || hasRole(ManagerRole) '>{{ "SIDE_NAV.MY_ACCOUNT" |  
translate }}</button>  
  
...
```

Dyrektywa `*ngIf` zapewnia wyświetlenie danego elementu `<button>` jeżeli wywołanie metody `hasRole()` zwróci wartość `true`.

Zadaniem metody `hasRole()` jest określenie, czy użytkownik posiada dany poziom dostępu.

```
...  
  
hasRole(role: string): boolean {  
    return this.authUtil.hasRole(role, this.userIdentity);  
}  
  
...  
  
export class AuthUtilService {  
  
    public hasRole(role: string, identity: AccountData): boolean {  
        let hasRole = false;  
        if (identity && identity.roles) {  
            hasRole = identity.roles.indexOf(role) !== -1;  
        }  
        return hasRole;  
    }  
  
    ...  
}
```

## 14 Zabezpieczenia interfejsu użytkownika 3 pkt

### Definicja ról uwzględnianych w kontroli dostępu:

```
<security-role>
  <description/>
  <role-name>USER</role-name>
</security-role>
<security-role>
  <description/>
  <role-name>MANAGER</role-name>
</security-role>
<security-role>
  <description/>
  <role-name>ADMIN</role-name>
</security-role>
<security-role>
  <description/>
  <role-name>VIRTUAL</role-name>
</security-role>
```

### Kontrola dostępu aplikacji:

```
<security-constraint>
  <display-name>ApplicationConstraint</display-name>
  <web-resource-collection>
    <web-resource-name>Appication</web-resource-name>
    <description/>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <user-data-constraint>
    <description/>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>
<security-constraint>
  <display-name>EditMyAccountConstraint</display-name>
  <web-resource-collection>
    <web-resource-name>EditMyAccount</web-resource-name>
    <url-pattern>/webresources/account/myAccountToEdit</url-pattern>
    <url-pattern>/webresources/account/updateMyAccount</url-pattern>
    <url-pattern>/webresources/account/changeMyPassword</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <description/>
    <role-name>USER</role-name>
    <role-name>MANAGER</role-name>
    <role-name>ADMIN</role-name>
  </auth-constraint>
  <user-data-constraint>
    <description/>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>
<security-constraint>
  <display-name>GetAllAccountsConstraint</display-name>
  <web-resource-collection>
    <web-resource-name>GetAllAccounts</web-resource-name>
    <url-pattern>/webresources/account/</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <description/>
    <role-name>MANAGER</role-name>
    <role-name>ADMIN</role-name>
  </auth-constraint>
  <user-data-constraint>
    <description/>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>
<security-constraint>
  <display-name>EditOtherAccountsConstraint</display-name>
  <web-resource-collection>
    <web-resource-name>EditOtherAccount</web-resource-name>
    <url-pattern>/webresources/account/updateAccount</url-pattern>
    <url-pattern>/webresources/account/changeOthersPassword</url-pattern>
    <url-pattern>/webresources/account/alterAccountAccessLevel</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <description/>
```

```

        <role-name>ADMIN</role-name>
    </auth-constraint>
    <user-data-constraint>
        <description/>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>
<security-constraint>
    <display-name>LockUnlockComfirmOtherAccountConstraint</display-name>
    <web-resource-collection>
        <web-resource-name>LockUnlockOtherAccount</web-resource-name>
        <url-pattern>/webresources/account/lockAccount</url-pattern>
        <url-pattern>/webresources/account/unlockAccount</url-pattern>
        <url-pattern>/webresources/account/confirmAccount</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <description/>
        <role-name>MANAGER</role-name>
        <role-name>ADMIN</role-name>
    </auth-constraint>
    <user-data-constraint>
        <description/>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>
<security-constraint>
    <display-name>GetAccesLevelsByAccountConstraint</display-name>
    <web-resource-collection>
        <web-resource-name>GetAccesLevelsByAccount</web-resource-name>
        <url-pattern>/webresources/account/allAccessLevel</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <description/>
        <role-name>MANAGER</role-name>
        <role-name>ADMIN</role-name>
    </auth-constraint>
    <user-data-constraint>
        <description/>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>

```

#### Role w warstwie widoku:

```

<security-role-mapping>
    <role-name>USER</role-name>
    <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>MANAGER</role-name>
    <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>ADMIN</role-name>
    <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>VIRTUAL</role-name>
    <group-name>virtual</group-name>
</security-role-mapping>

```

#### Dostęp do poszczególnych funkcji systemu na podstawie posiadanej roli w module MOK:

```

<security-role-mapping>
    <role-name>getAllAccounts</role-name>
    <group-name>admin</group-name>
    <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>getAllAccessLevels</role-name>
    <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>saveAccountAfterEdit</role-name>

```

```
<group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>confirmAccount</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>lockAccount</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>confirmAccount</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>unlockAccount</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>addAccessLevelToAccount</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>dismissAccessLevelFromAccount</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>updateLoginDateAndIp</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
  <group-name>virtual</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAccountAlevel</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>changeMyPassword</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>changeOthersPassword</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>saveMyAccountAfterEdit</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>alterAccountAccessLevel</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAllArchivalPasswordsByAccount</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getMyAccountByLogin</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
  <group-name>virtual</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAccountById</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>saveMyAccountAfterEdit</role-name>
  <group-name>admin</group-name>
```

```

    <group-name>manager</group-name>
    <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>getAccessLevelById</role-name>
    <group-name>admin</group-name>
</security-role-mapping>

```

#### Dostęp do poszczególnych funkcji systemu na podstawie posiadanej roli w module MOP:

```

<security-role-mapping>
    <role-name>addProductByAccount</role-name>
    <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>deleteProductByAccount</role-name>
    <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>updateProduct</role-name>
    <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>setProductState</role-name>
    <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>getAllProducts</role-name>
    <group-name>admin</group-name>
    <group-name>manager</group-name>
    <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>getAllProductsContainName</role-name>
    <group-name>admin</group-name>
    <group-name>manager</group-name>
    <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>getProductFromAccountByName</role-name>
    <group-name>admin</group-name>
    <group-name>manager</group-name>
    <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>setProductCategory</role-name>
    <group-name>manager</group-name>
</security-role-mapping>

```

#### Dostęp do poszczególnych funkcji systemu na podstawie posiadanej roli w module MOZ:

```

<security-role-mapping>
    <role-name>makeOrder</role-name>
    <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>makeOrderPayment</role-name>
    <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>cancelOrder</role-name>
    <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>addShippingMethod</role-name>
    <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>setShippingMethodActive</role-name>
    <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
    <role-name>setOrderStatus</role-name>
    <group-name>user</group-name>

```

```

</security-role-mapping>
<security-role-mapping>
  <role-name>getAllOrders</role-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAllOrdersByAccountAsSeller</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAllOrdersByAccountAsBuyer</role-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>getAllOrdersByDateAndPrice</role-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>

```

#### Dostęp do fasad systemu na podstawie posiadanej roli:

```

<security-role-mapping>
  <role-name>findAll</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>edit</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findAccount</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findByLogin</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
  <group-name>virtual</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findByAccountAndAccessLevel</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>removeAccountAlevel</role-name>
  <group-name>admin</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findArchivalPasswordByAccountId</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findRange</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>count</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
  <group-name>user</group-name>
</security-role-mapping>
<security-role-mapping>
  <role-name>findAccessLevel</role-name>
  <group-name>admin</group-name>
  <group-name>manager</group-name>
</security-role-mapping>

```



Wszystkie hasła przed zapisaniem do bazy są hashowane za pomocą poniższej funkcji.

Funkcja hashująca hasła:

```
public static String sha256(String base) {
    try {
        MessageDigest digest = MessageDigest.getInstance("SHA-256");
        byte[] hash = digest.digest(base.getBytes("UTF-8"));
        StringBuffer hexString = new StringBuffer();

        for (int i = 0; i < hash.length; i++) {
            String hex = Integer.toHexString(0xff & hash[i]);
            if (hex.length() == 1) {
                hexString.append('0');
            }
            hexString.append(hex);
        }

        return hexString.toString();
    } catch (Exception ex) {
        throw new RuntimeException(ex);
    }
}
```

Przykład zastosowania funkcji hashującej został zamieszczony poniżej.

Wykorzystanie skrótu hasel przy rejestracji:

```
@Override
@PermitAll
public void registerAccount(Account account, ServletContext servletContext) throws AppBaseException {
    account.setPassword(HashUtils.sha256(account.getPassword()));
    accountFacade.create(account);

    AccountAlevel level = new AccountAlevel();
    level.setIdAccount(account);
    level.setIdAlevel(accessLevelFacade.findByLevel(DEFAULT_ACCESS_LEVEL));
    accountAlevelFacade.create(level);

    ArchivalPassword archivalPassword = new ArchivalPassword(account.getPassword(),
generateCurrentDate(), account);
    archivalPasswordFacade.create(archivalPassword);

    this.sendMailWithVerificationLink(account.getEmail(), createVerificationLink(account,
servletContext));
}
```

## 15 Zmiany - projekt szczegółowy

### Modyfikacja relacji między kluczami tabel *account* i *user\_data*

Strona poddana modyfikacjom: 3 Struktury relacyjnej bazy danych 2 pkt

Usunięto klucz obcy *account\_user\_data\_fk* w tabeli *account* odnoszący się do kolumny *id* w tabeli *user\_data*. Dodano klucz obcy *user\_data\_account\_fk* w tabeli *user\_data* odnoszący się do kolumny *id* w tabeli *account*.

### Dodanie kolumn w tabeli *account*

Strona poddana modyfikacjom: 3 Struktury relacyjnej bazy danych 2 pkt

Dodano kolumny *last\_login\_date* i *last\_login\_ip* służące do przechowywania odpowiednio: daty ostatniego logowania oraz adresu IP, z którego nastąpiło ostatnie uwierzytelnienie.

### Dodanie tabel *verification\_token*, *archival\_password*, sekwencji *archival\_password\_id\_seq*

Strona poddana modyfikacjom: 3 Struktury relacyjnej bazy danych 2 pkt

Dodano tabelę *verification\_token*, w której przechowywane są tokeny aktywacji kont - generowane ciągi znaków, na podstawie których weryfikowane jest konto po dokonaniu autoryzacji. Tabela odnosi się do tabeli *account* poprzez klucz obcy *verification\_token\_account\_fk*.

Dodano tabelę *archival\_password*, w której przechowywane są ustawiane przez użytkownika hasła wraz z datą ich ustawienia. Tabela odnosi się do tabeli *account* poprzez klucz obcy *archival\_password\_id\_account\_fk*.

Dodano sekwencję *archival\_password\_id\_seq* do generacji identyfikatorów rekordów tabeli *archival\_password*.

## Dodanie uprawnień użytkownikom bazodanowym do nowoutworzonych tabel i sekwencji

Strona poddana modyfikacjom: 4 Użytkownicy bazodanowi 2 pkt

Dodano użytkownikowi *ssbd01mok* uprawnienia do dokonywania SELECT, INSERT, UPDATE na tabeli *verification\_token*, natomiast SELECT, UPDATE do tabeli *archival\_password* oraz SELECT, USAGE na sekwencji *archival\_password\_id\_seq*. Użytkownicy *ssbd01* i *ssbd01admin* posiadają uprawnienia do dokonywania SELECT, UPDATE, INSERT, DELETE, TRUNCATE, REFERENCES, TRIGGER na tabelach *verification\_token* i *archival\_password*, oraz SELECT, UPDATE, USAGE na sekwencji *archival\_password\_id\_seq*.

## Sprawozdanie końcowe (15 pkt)

## Sprawozdanie końcowe

### 16 Zabezpieczenie transmisji HTTP 1 pkt

Bezpieczne, szyfrowane połączenie HTTPS zapewniane jest poprzez wpis w deskrytorze `web.xml`. Każdy adres który odpowiada wzorcowy podanemu w `<url-pattern>URL</url-pattern>` i posiada w swojej strukturze znacznik `<transport-guarantee>` z argumentem `CONFIDENTIAL` będzie komunikował się za pomocą szyfrowanego protokołu HTTPS. Poniżej zaprezentowany jest przykład:

```
<security-constraint>
  <display-name>ApplicationConstraint</display-name>
  <web-resource-collection>
    <web-resource-name>Application</web-resource-name>
    <description/>
    <url-pattern>*/</url-pattern>
  </web-resource-collection>
  <user-data-constraint>
    <description/>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>
```

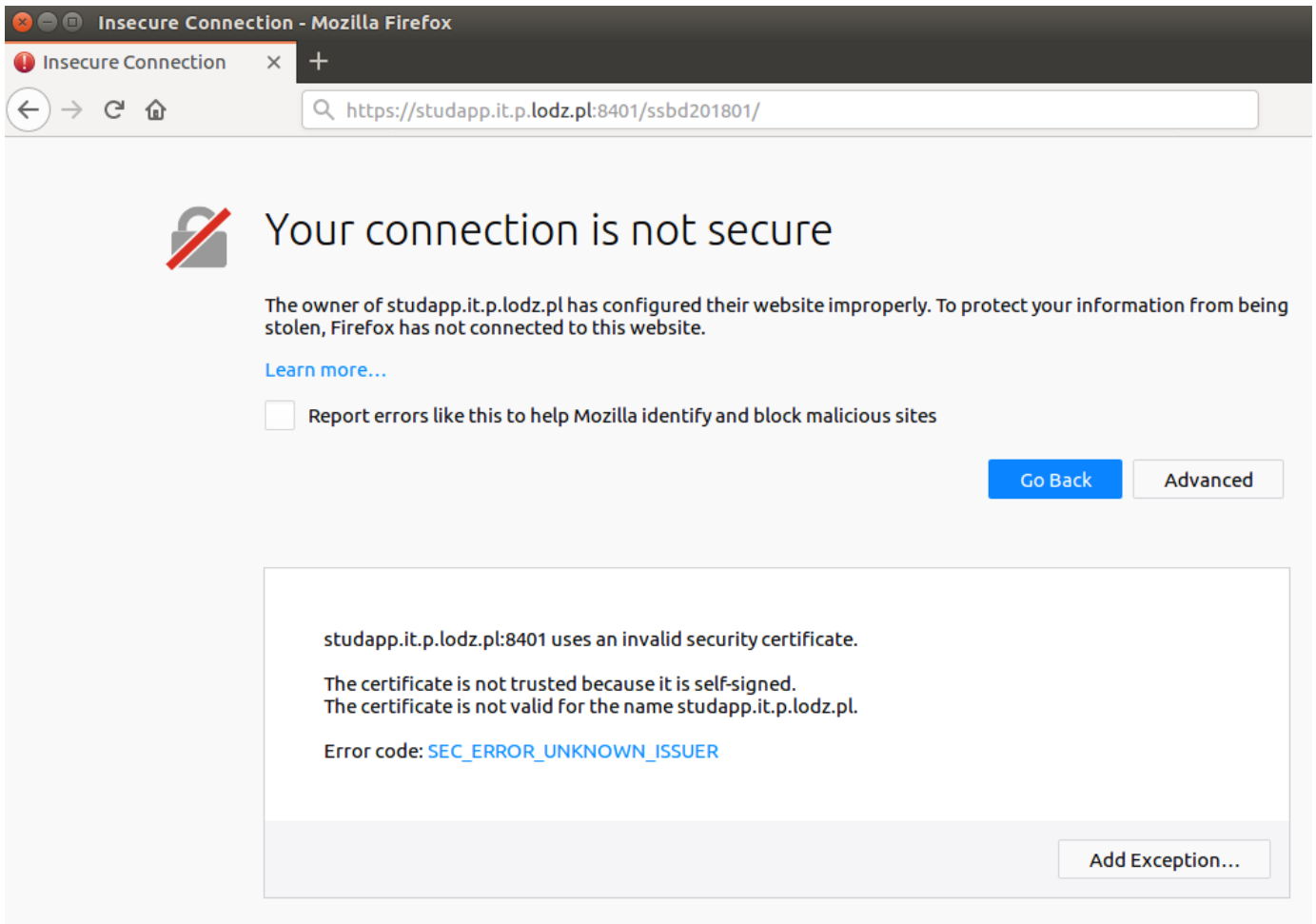
Poniżej przedstawiony jest początkowy przebieg/inicjalizacja komunikacji pomiędzy klientem (przeglądarką), a serwerem aplikacji.

1. Klient wysyła zapytanie `https://` do serwera znajdującego się pod danym adresem z prośbą o przedstawienie tożsamości i przesłanie posiadanego certyfikatu.
2. Serwer odpowiada przesyłając kopię posiadanego certyfikatu i klucz publiczny serwera.
3. Klient sprawdza korzeń (*root*) certyfikatu w liście zaufanych certyfikatów które posiada lokalnie. Sprawdza również czy nie został on już odwołany lub czy nie wygaś. Następnie jeżeli przeglądarka stwierdza zgodność przekazanego certyfikatu to tworzy symetryczny klucz sesji który po zaszyfrowaniu kluczem przesłanym przez serwer przesyła jako odpowiedź do serwera.
4. Serwer deszyfruje klucz symetryczny z pomocą posiadanego klucza prywatnego i przesyła do klienta potwierdzenie zaszyfrowane kluczem symetrycznym rozpoczynając w ten sposób komunikację szyfrowaną (bezpieczną).

Powodem dla którego przeglądarka traktuje nawiązane połączenie jako niezaufane jest brak podpisania certyfikatu poprzez zaufanego wydawcę.

Cała nasza aplikacja wymaga połączenia z certyfikacją SSL, a wykonanie zapytania poprzez protokół HTTP powoduje natychmiastowe przekierowanie na komunikację poprzez protokół HTTPS.

Poniższy zrzut z ekranu przeglądarki obrazuje komunikat o niezaufanym certyfikacie, ponieważ nie jest on poparty przez zaufanego wydawcę certyfikatu.



## 17 Weryfikacja poprawności danych 3 pkt

Przykład konwersji z typu String do BigDecimal

```
private void validatePriceAndQtyOfProduct(BasicProductDto product) throws
AppBaseException{
    try {
        BigDecimal price = new BigDecimal(product.getPrice());
    } catch(NumberFormatException e) {
        throw new ProductPriceException("product_price_exception");
    }
    try {
        BigDecimal qty = new BigDecimal(product.getQty());
    } catch(NumberFormatException e) {
        throw new ProductQtyException("product_qty_exception");
    }
}
```

Weryfikacja wypełnienia pola w warstwie prezentacji

```
<mat-form-field>
  <input matInput [placeholder]=" 'PRODUCT.NAME' | translate"
```

```

formControlName='name' required>
</mat-form-field>
<div class="errors-space" *ngIf="isRequiredSatisfied('name')">
    {{ "FORM.FIELD_REQUIRED" | translate }}
</div>

```

#### Weryfikacja przez metodę warstwy logiki

```

private void validateToken(String token) throws WrongTokenException {
    if (token == null || token.length() < 32) {
        throw new WrongTokenException("wrong_token");
    }
}

```

#### Metoda wykorzystująca adnotację BeanValidation w celu weryfikacji

```

private void validateConstraints(Account account) throws AppBaseException {
    ValidatorFactory factory = Validation.
buildDefaultValidatorFactory();
    Validator validator = factory.getValidator();
    Set<ConstraintViolation<Account>> constraintViolations = validator.
validate(account);
    List<String> errors = new ErrorCodes().getAllErrors();

    if (constraintViolations.size() > 0) {
        for (int i = 0; i < errors.size(); i++) {
            for (ConstraintViolation<Account> temp :
constraintViolations) {
                if (errors.get(i).equals(temp.getMessage())) {
                    throw new ConstraintException("constraint_error",
errors.get(i));
                }
            }
        }
    }
}

```

## 18 Obsługa błędów 3 pkt

Nasza aplikacja została wzbogacona o system zgłaszania wyjątków aplikacyjnych, który ma posłużyć jako narzędzie do wykrywania i obsługiwanie poprzez zgłoszenie niestandardowych przebiegów podczas działania systemu.

Przykładem zgłaszanego wyjątku aplikacyjnego jest przekroczenie w zamówieniu liczby dostępnych produktów.

Poniżej znajduje się fragment kodu aplikacji sprawdzający ten przypadek.:

```

private void checkIfNotBelowZero(Double number) throws AppBaseException{
    if (number < 0.0) {
        throw new ProductNotEnoughException("product_qty_not_enough");
    }
}

```

Klasa obsługująca ten przypadek:

```

public class ProductNotEnoughException extends ProductException {

    public ProductNotEnoughException(String message, Throwable cause) {
        super(message, cause);
    }

    public ProductNotEnoughException(String message) {
        super(message);
    }

    @Override
    public String getCode() {
        return super.getCode() + ".product_qty_not_enough";
    }
}

```

### Przebieg informacji

Podczas wystąpienia tego typu wyjątku zwracany jest do warstwy prezentacji kod błędu w formacie dotyczącym poszczególnych składowych (typ wyjątku, moduł oraz pakiet z jakiego pochodzi, kontekst błędu). Kod ten jest następnie mapowany na odpowiedni komunikat błędu który wyświetli się w przeglądarce użytkownika danej sesji.

Poniżej znajduje się przykład mapowania wyżej wymienionego wyjątku:

```
"app.exception.mop.product.product_qty_not_enough": "Sprzedajcy nie dysponuje podan iloci produktu"
```

### Wyjątek kontenera EJB:

Poniższy fragment kodu aplikacji odpowiada za obsłużenie sytuacji w której to użytkownik próbuje się dostać do zasobu do którego nie ma prawa dostępu. Odpowiedni kod błędu zostanie zwrócony do warstwy prezentacji która to przekieruje danego użytkownika na stronę (zasób URL) z kodem błędu 403.

```

@Provider
@Priority(1)
public class EJBAccessExceptionMapper implements ExceptionMapper<EJBAccessException> {

    @Override
    public Response toResponse(EJBAccessException exception) {
        return Response.status(Response.Status.FORBIDDEN)
            .build();
    }
}

```

### Opis konfiguracji:

Użytkownik wysyłający zapytanie HTTP na nieistniejący zasób uruchamia procedurę gdzie warstwa prezentacji sprawdza zbiór dostępnych zasobów URL i w przypadku gdy takiego nie znajdzie przekierowuje danego użytkownika na stronę (zasób URL) z błędem o kodzie 404.

Podczas wystąpienia nieznanego błędu podczas działania aplikacji, system informuje o tym użytkownika wyświetlając informację o napotkaniu nieznanego błędu. Oczywiście nasz system został zaprojektowany i zaimplementowany w sposób iż tego typu przypadki nie występują. Ponadto każda operacja jest odnotowywana w dzienniku zdarzeń co pozwala namierzyć ew. problem.

## 19 Wersje językowe interfejsu użytkownika 2 pkt

Do internacjonalizacji interfejsu aplikacji wykorzystano bibliotekę **ngx-translate**.

```

const Languages = ['pl', 'en'];

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})

```

```

export class AppComponent {

  constructor(private translateService: TranslateService) {
    this.setupTranslationService();
  }

  private setupTranslationService() {
    this.translateService.addLangs(Languages);
    this.translateService.setDefaultLang(this.translateService.
getBrowserLang());
  }
}

```

Przedstawiony powyżej fragment kodu przedstawia główny komponent aplikacji. Klasa **TranslateService**, udostępniana przez bibliotekę ngx-translate, umożliwia dostęp do metod pozwalających na konfigurację mechanizmu zapewniającego internacjonalizację. Metoda **addLangs()** umożliwia zdefiniowanie listy języków obsługiwanych przez aplikację. Wywołanie **this.translateService.setDefaultLang(this.translateService.getBrowserLang())** zapewnia wykorzystanie przez aplikację języka przeglądarki jako języka domyślnego.

Przez kod zawarty w bibliotece wyszukiwane są pliki pl.json i en.json (ich nazwy są determinowane przez zmienną **Languages** w listingu powyżej). Są to pliki słowników, z których aplikacja będzie korzystać.

Przykładowy fragment pliku słownika:

```

{
  "HOME": {
    "TITLE": "Strona główna",
    "WELCOME": "Witaj",
    "USER_ROLES": "Poziomy dostępu użytkownika",
    "ACTIVATE_ACCOUNT": "Musisz aktywować konto!"
  },
  "SIDE_NAV": {
    "MAIN_PAGE": "Strona główna",
    "REGISTER": "Rejestracja",
    "LOGIN": "Logowanie",
    "ACCOUNTS": "Lista kont",
    "STATISTICS": "Statystyki kont",
    "MY_ACCOUNT": "Moje konto",
    "CHANGE_MY_PASSWORD": "Zmień hasło",
    "LOGOUT": "Wyloguj",
    "GUEST": "Go",
    "CREATE_ACCOUNT": "Utwórz konto",
    "ACTIVATE_ACCOUNT": "Aktywuj konto",
    "PRODUCTS": "Produkty",
    "MY_PRODUCTS": "Moje produkty",
    "MY_ORDERS": "Moje zamówienia",
    "ALL_ORDERS": "Wszystkie zamówienia"
  }
}
...

```

```

    "app.exception.mop.category.category_unique_name": "Kategoria o podanej nazwie ju istnieje!",
    "app.exception.mop.category.category_name_length" : "Nazwa tej kategorii jest zbyt duga.",
    "app.exception.moz.order.order_qty_null_exception": "Podaj ilo produktu jak zamierzasz kupi.",
    "app.exception.moz.shipping_method.not_found": "Nie wybrano sposobu dostawy. Bdzie dziaa lepiej jak wybierzesz ;)",
    "app.exception.moz.order.qty_format_error": "Zy format podanej liczby. Akceptowany format to np. 1.2 lub 3 ",
    "app.exception.mop.product.product_qty_not_enough": "Sprzedajcy nie dysponuje podan iloci produktu",
    "app.exception.moz.order.qty_error": "Jak nie chcesz kupi to prosz nie wpisuj 0 lub mniej",
    "app.exception.moz.order.qty_too_low": "Podana ilo jest zbyt niska",
    "order_status_not_found": "Nie znaleziono statusu zamówienia.",
    "order_not_found": "Nie znaleziono zamówienia."
}

```

W plikach słowników zawarte są zarówno klucze wartości w warstwie widoku, jak i klucze błędów otrzymywanych z serwera.

W dolnej części powyższego listingu można zauważyć również kody błędów dostarczanych z warstwy logiki wraz z tłumaczeniami.

Wartości dla kluczy mogą być dobierane przy pomocy następującej konstrukcji (w tym przypadku jest to przykład tłumaczenia komunikatu błędu walidatora):

```

...
<div class="errors-space" *ngIf="isRequiredSatisfied('firstPassword')">
  {{ "FORM.FIELD_REQUIRED" | translate }}
</div>
...

```

Rurka (ang. "pipe", jest to klasa, której zadaniem zazwyczaj jest prosta transformacja tekstu w szablonie HTML) **translate** umożliwia zamianę klucza, w tym przypadku "FORM.FIELD\_REQUIRED", na przypisaną do niego wartość znaną w słowniku.

Podobnie dobierane są wartości zadeklarowane w kodzie języka TypeScript (tutaj jest to przykład tłumaczenia komunikatu błędu otrzymanego z warstwy logiki):

```

this.formValidationMessage = this.translateService.instant(errorResponse.error.message);

```

Klucz błędu (wartość zmiennej `errorResponse.error.message`) zostaje pobrany z serwera i jest tłumaczony z użyciem słownika, podobnie jak we wcześniejszym przykładzie.

## 20 Opis przypadkow uzycia MVCv2 6 pkt

numer	oznaczenie	autor	opis	nazwa komponentu ejb, metody realizujące przypadek	role	nawigacja
1	MOK.1	Agata Kania, Piotr Biernacki, Michał Malec, Filip Różański, Dominik Lange	Metoda udostępniająca endpoint REST dla klienta w celu rejestracji nowego użytkownika	<ul style="list-style-type: none"> <li>AccountManager.registerAccount</li> <li>AccountFacade.create</li> <li>AccessLevelFacade.findByLevel</li> <li>AccountAlevelFacade.create</li> <li>ArchivalPasswordFacadeLocal.create</li> </ul>	<ul style="list-style-type: none"> <li>Gość</li> </ul>	<ul style="list-style-type: none"> <li>main</li> <li>register</li> <li>login</li> </ul>
2	MOK.2	Agata Kania, Piotr Biernacki, Michał Malec, Filip Różański, Dominik Lange	Metoda udostępniająca endpoint REST dla klienta w celu rejestracji nowego użytkownika	<ul style="list-style-type: none"> <li>AccountManager.registerAccount</li> <li>AccountFacade.create</li> <li>AccessLevelFacade.findByLevel</li> <li>AccountAlevelFacade.create</li> <li>ArchivalPasswordFacadeLocal.create</li> </ul>	<ul style="list-style-type: none"> <li>Administrator</li> </ul>	<ul style="list-style-type: none"> <li>main</li> <li>accounts</li> <li>statistics</li> <li>myAccount</li> <li>changeMyPassword</li> <li>register</li> <li>products</li> <li>makeOrder</li> <li>myProducts</li> <li>addProduct</li> <li>orders</li> <li>categories</li> <li>ownOrders</li> <li>shippingMethods</li> <li>addShippingMethod</li> </ul>
3	MOK.3	Agata Kania, Piotr Biernacki, Michał Malec, Filip Różański, Dominik Lange	Metoda udostępniająca endpoint REST dla klienta w celu zablokowania konta o podanym numerze ID	<ul style="list-style-type: none"> <li>AccountManager.getAllAccounts</li> <li>AccountFacade.findAll</li> <li>AccountManager.lockAccount</li> <li>AccountFacade.find</li> <li>AccountFacade.edit</li> </ul>	<ul style="list-style-type: none"> <li>Moderator</li> <li>Administrator</li> </ul>	<ul style="list-style-type: none"> <li>main</li> <li>accounts</li> <li>statistics</li> <li>myAccount</li> <li>changeMyPassword</li> <li>register</li> <li>products</li> <li>makeOrder</li> <li>myProducts</li> <li>addProduct</li> <li>orders</li> <li>categories</li> <li>ownOrders</li> <li>shippingMethods</li> <li>addShippingMethod</li> </ul>
4	MOK.4	Agata Kania, Piotr Biernacki, Michał Malec, Filip Różański,	Metoda udostępniająca endpoint REST dla klienta w celu	<ul style="list-style-type: none"> <li>AccountManager.getAllAccounts</li> <li>AccountFacade.findAll</li> <li>AccountManager.unlockAccount</li> <li>AccountFacade.find</li> <li>AccountFacade.edit</li> </ul>	<ul style="list-style-type: none"> <li>Moderator</li> <li>Administrator</li> </ul>	<ul style="list-style-type: none"> <li>main</li> <li>accounts</li> <li>statistics</li> <li>myAccount</li> <li>changeMyPassword</li> </ul>



		Dominik Lange	odblokowania konta o podanym numerze ID			<ul style="list-style-type: none"> <li>• register</li> <li>• products</li> <li>• makeOrder</li> <li>• myProducts</li> <li>• addProduct</li> <li>• orders</li> <li>• categories</li> <li>• ownOrders</li> <li>• shippingMethods</li> <li>• addShippingMethod</li> </ul>
5	MOK.5	Agata Kania, Piotr Biernacki, Michał Malec, Filip Różański, Dominik Lange	Metoda udostępniająca endpoint REST dla klienta w celu zmiany poziomów dostępu dla danego konta	<ul style="list-style-type: none"> <li>• AccountManager.getAllAccounts</li> <li>• AccountFacade.findAll</li> <li>• AccountManager.getAccountById</li> <li>• AccountFacade.find</li> <li>• AccountManager.alterAccountAccessLevel</li> <li>• AccessLevelFacade.findByLevel</li> </ul>	<ul style="list-style-type: none"> <li>• Administrator</li> </ul>	<ul style="list-style-type: none"> <li>• main</li> <li>• accounts</li> <li>• statistics</li> <li>• myAccount</li> <li>• changeMyPassword</li> <li>• register</li> <li>• products</li> <li>• makeOrder</li> <li>• myProducts</li> <li>• addProduct</li> <li>• orders</li> <li>• categories</li> <li>• ownOrders</li> <li>• shippingMethods</li> <li>• addShippingMethod</li> </ul>
6	MOK.6	Agata Kania, Piotr Biernacki, Michał Malec, Filip Różański, Dominik Lange	Metoda udostępniająca endpoint REST dla klienta w celu zmiany poziomów dostępu dla danego konta	<ul style="list-style-type: none"> <li>• AccountManager.getAllAccounts</li> <li>• AccountFacade.findAll</li> <li>• AccountManager.getAccountById</li> <li>• AccountFacade.find</li> <li>• AccountManager.alterAccountAccessLevel</li> <li>• AccessLevelFacade.findByLevel</li> </ul>	<ul style="list-style-type: none"> <li>• Administrator</li> </ul>	<ul style="list-style-type: none"> <li>• main</li> <li>• accounts</li> <li>• statistics</li> <li>• myAccount</li> <li>• changeMyPassword</li> <li>• register</li> <li>• products</li> <li>• makeOrder</li> <li>• myProducts</li> <li>• addProduct</li> <li>• orders</li> <li>• categories</li> <li>• ownOrders</li> <li>• shippingMethods</li> <li>• addShippingMethod</li> </ul>
7	MOK.7	Agata Kania, Piotr Biernacki, Michał Malec, Filip Różański, Dominik Lange	Metoda udostępniająca endpoint REST dla klienta w celu zmiany własnego hasła	<ul style="list-style-type: none"> <li>• AccountManager.getMyAccountByLogin</li> <li>• AccountFacade.findByLogin</li> <li>• AccountFacade.changeMyPassword</li> <li>• AccountFacade.edit</li> <li>• ArchivalPasswordFacade.create</li> </ul>	<ul style="list-style-type: none"> <li>• Użytkownik</li> <li>• Moderator</li> <li>• Administrator</li> </ul>	<ul style="list-style-type: none"> <li>• main</li> <li>• myAccount</li> <li>• changeMyPassword</li> <li>• products</li> <li>• makeOrder</li> <li>• myProducts</li> <li>• addProduct</li> <li>• ownOrders</li> </ul>
8	MOK.8	Agata Kania, Piotr Biernacki, Michał Malec, Filip Różański, Dominik Lange	Metoda udostępniająca endpoint REST dla klienta w celu zmiany hasła danego konta przez administratora	<ul style="list-style-type: none"> <li>• AccountManager.getAccountById</li> <li>• AccountFacade.findAll</li> <li>• AccountManager.changeOthersPassword</li> <li>• AccountFacade.edit</li> <li>• ArchivalPasswordFacade.create</li> </ul>	<ul style="list-style-type: none"> <li>• Administrator</li> </ul>	<ul style="list-style-type: none"> <li>• main</li> <li>• accounts</li> <li>• statistics</li> <li>• myAccount</li> <li>• changeMyPassword</li> <li>• register</li> <li>• products</li> <li>• makeOrder</li> <li>• myProducts</li> <li>• addProduct</li> </ul>

						<ul style="list-style-type: none"> <li>orders</li> <li>categories</li> <li>ownOrders</li> <li>shippingMethods</li> <li>addShippingMethod</li> </ul>
9	MOK.9	Agata Kania, Piotr Biernacki, Michał Malec, Filip Różański, Dominik Lange	Metoda udostępniająca endpoint REST dla klienta w celu zaktualizowania danych swojego konta	<ul style="list-style-type: none"> <li>AccountManager. getMyAccountByLogin <ul style="list-style-type: none"> <li>AccountFacade.findByLogin</li> </ul> </li> <li>AccountManager. saveMyAccountAfterEdit <ul style="list-style-type: none"> <li>AccountFacade.edit</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Użytkownik</li> <li>Moderator</li> <li>Administrator</li> </ul>	<ul style="list-style-type: none"> <li>main</li> <li>myAccount</li> <li>changeMyPassword</li> <li>products</li> <li>makeOrder</li> <li>myProducts</li> <li>addProduct</li> <li>ownOrders</li> </ul>
10	MOK.10	Agata Kania, Piotr Biernacki, Michał Malec, Filip Różański, Dominik Lange	Metoda udostępniająca endpoint REST dla klienta w celu zaktualizowania danych konta przez administratora	<ul style="list-style-type: none"> <li>AccountManager.getAccountById <ul style="list-style-type: none"> <li>AccountFacade.find</li> </ul> </li> <li>AccountManager. saveAccountAfterEdit <ul style="list-style-type: none"> <li>AccountFacade.edit</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Administrator</li> </ul>	<ul style="list-style-type: none"> <li>main</li> <li>accounts</li> <li>statistics</li> <li>myAccount</li> <li>changeMyPassword</li> <li>register</li> <li>products</li> <li>makeOrder</li> <li>myProducts</li> <li>addProduct</li> <li>orders</li> <li>categories</li> <li>ownOrders</li> <li>shippingMethods</li> <li>addShippingMethod</li> </ul>
11	MOK.11	Michał Malec, Dominik Lange	Metoda umostępniająca endpoint REST w celu zamknięcia sesji użytkownika		<ul style="list-style-type: none"> <li>Użytkownik</li> <li>Moderator</li> <li>Administrator</li> </ul>	<ul style="list-style-type: none"> <li>main</li> <li>myAccount</li> <li>changeMyPassword</li> <li>products</li> <li>makeOrder</li> <li>myProducts</li> <li>addProduct</li> <li>ownOrders</li> </ul>
12	MOK.12	Agata Kania, Piotr Biernacki, Michał Malec, Filip Różański, Dominik Lange	Metoda udostępniająca endpoint REST dla klienta w celu pobrania listy kont	<ul style="list-style-type: none"> <li>AccountManager.getAllAccounts <ul style="list-style-type: none"> <li>AccountFacade.findAll</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Moderator</li> <li>Administrator</li> </ul>	<ul style="list-style-type: none"> <li>main</li> <li>accounts</li> <li>statistics</li> <li>myAccount</li> <li>changeMyPassword</li> <li>register</li> <li>products</li> <li>makeOrder</li> <li>myProducts</li> <li>addProduct</li> <li>orders</li> <li>categories</li> <li>ownOrders</li> <li>shippingMethods</li> <li>addShippingMethod</li> </ul>
13	MOK.13	Agata Kania, Piotr Biernacki, Michał Malec,	Metoda udostępniająca endpoint REST dla klienta w celu pobrania listy kont	<ul style="list-style-type: none"> <li>AccountManager.getAllAccounts <ul style="list-style-type: none"> <li>AccountFacade.findAll</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Moderator</li> <li>Administrator</li> </ul>	<ul style="list-style-type: none"> <li>main</li> <li>accounts</li> <li>statistics</li> <li>myAccount</li> </ul>

		Filip Różański, Dominik Lange				<ul style="list-style-type: none"> <li>• changeMyPassw ord</li> <li>• register</li> <li>• products</li> <li>• makeOrder</li> <li>• myProducts</li> <li>• addProduct</li> <li>• orders</li> <li>• categories</li> <li>• ownOrders</li> <li>• shippingMethods</li> <li>• addShippingMet hod</li> </ul>
14	MOK.14	Agata Kania, Piotr Biernacki, Michał Malec, Filip Różański, Dominik Lange	Metoda udostępniająca endpoint REST dla klienta w celu pobrania listy kont	<ul style="list-style-type: none"> <li>• AccountManager.getAllAccounts</li> <li>• AccountFacade.findAll</li> </ul>	<ul style="list-style-type: none"> <li>• Moderator</li> <li>• Administrator</li> </ul>	<ul style="list-style-type: none"> <li>• main</li> <li>• accounts</li> <li>• statistics</li> <li>• myAccount</li> <li>• changeMyPassw ord</li> <li>• register</li> <li>• products</li> <li>• makeOrder</li> <li>• myProducts</li> <li>• addProduct</li> <li>• orders</li> <li>• categories</li> <li>• ownOrders</li> <li>• shippingMethods</li> <li>• addShippingMet hod</li> </ul>
15	MOP.1	Agata Kania	Metoda REST API służąca do dodawania nowego produktu przez użytkownika	<ul style="list-style-type: none"> <li>• ProductManager. addProductByAccountLogin</li> <li>• CategoryManager.getCategoryById</li> <li>• CategoryFacade.find</li> <li>• ProductFacade.findByLogin</li> <li>• ProductFacade.create</li> </ul>	<ul style="list-style-type: none"> <li>• Użytkownik</li> </ul>	<ul style="list-style-type: none"> <li>• main</li> <li>• myAccount</li> <li>• changeMyPassw ord</li> <li>• products</li> <li>• makeOrder</li> <li>• myProducts</li> <li>• addProduct</li> <li>• ownOrders</li> </ul>
16	MOP.2	Piotr Biernacki	Metoda udostępniająca endpoint REST dla klienta w celu usunięcia produktu o podanym numerze ID	<ul style="list-style-type: none"> <li>• ProductManager. deleteProductByAccount</li> <li>• ProductFacade.find</li> <li>• AccountFacade.findByLogin</li> <li>• ProductFacade.remove</li> <li>• AccountFacade.edit</li> </ul>	<ul style="list-style-type: none"> <li>• Użytkownik</li> </ul>	<ul style="list-style-type: none"> <li>• main</li> <li>• myAccount</li> <li>• changeMyPassw ord</li> <li>• products</li> <li>• makeOrder</li> <li>• myProducts</li> <li>• addProduct</li> <li>• ownOrders</li> </ul>
17	MOP.3	Michał Malec	Metoda REST API służąca do edycji produktu przez użytkownika	<ul style="list-style-type: none"> <li>• ProductManager.editProduct</li> <li>• ProductFacade.find</li> <li>• UnitFacade.find</li> <li>• ProductFacade.edit</li> </ul>	<ul style="list-style-type: none"> <li>• Użytkownik</li> </ul>	<ul style="list-style-type: none"> <li>• main</li> <li>• myAccount</li> <li>• changeMyPassw ord</li> <li>• products</li> <li>• makeOrder</li> <li>• myProducts</li> <li>• addProduct</li> <li>• ownOrders</li> </ul>
18	MOP.4	Michał Malec	Metoda udostępniająca endpoint REST dla klienta w celu aktywowania /dezaktywowania	<ul style="list-style-type: none"> <li>• ProductManager.activeProduct</li> <li>• ProductFacade.find</li> <li>• ProductFacade.edit</li> <li>• ProductManager.deactiveProduct</li> <li>• ProductFacade.find</li> </ul>	<ul style="list-style-type: none"> <li>• Użytkownik</li> </ul>	<ul style="list-style-type: none"> <li>• main</li> <li>• myAccount</li> <li>• changeMyPassw ord</li> <li>• products</li> </ul>

			produktu o podanym numerze ID	<ul style="list-style-type: none"> <li>ProductFacade.edit</li> </ul>		<ul style="list-style-type: none"> <li>makeOrder</li> <li>myProducts</li> <li>addProduct</li> <li>ownOrders</li> </ul>
19	MOP.5	Agata Kania, Michał Malec	Metoda udostępniająca endpoint REST dla klienta w celu pobrania listy wszystkich produktów	<ul style="list-style-type: none"> <li>ProductManager.getAllProducts</li> <li>ProductFacade.findAll</li> </ul>	<ul style="list-style-type: none"> <li>Użytkownik</li> <li>Moderator</li> <li>Administrator</li> </ul>	<ul style="list-style-type: none"> <li>main</li> <li>myAccount</li> <li>changeMyPassword</li> <li>products</li> <li>makeOrder</li> <li>myProducts</li> <li>addProduct</li> <li>ownOrders</li> </ul>
20	MOP.6	Agata Kania	Metoda udostępniająca endpoint REST dla klienta w celu pobrania listy wszystkich produktów	<ul style="list-style-type: none"> <li>ProductManager.getAllProducts</li> <li>ProductFacade.findAll</li> </ul>	<ul style="list-style-type: none"> <li>Użytkownik</li> <li>Moderator</li> <li>Administrator</li> </ul>	<ul style="list-style-type: none"> <li>main</li> <li>myAccount</li> <li>changeMyPassword</li> <li>products</li> <li>makeOrder</li> <li>myProducts</li> <li>addProduct</li> <li>ownOrders</li> </ul>
21	MOP.7	Dominik Lange	Metoda REST API służąca do dodawania nowych kategorii produktów	<ul style="list-style-type: none"> <li>CategoryManager.addCategory</li> <li>CategoryFacade.create</li> </ul>	<ul style="list-style-type: none"> <li>Moderator</li> </ul>	<ul style="list-style-type: none"> <li>main</li> <li>accounts</li> <li>statistics</li> <li>myAccount</li> <li>changeMyPassword</li> <li>register</li> <li>products</li> <li>makeOrder</li> <li>myProducts</li> <li>addProduct</li> <li>orders</li> <li>categories</li> <li>ownOrders</li> <li>shippingMethods</li> <li>addShippingMethod</li> </ul>
22	MOP.8	Agata Kania, Michał Malec	Metoda udostępniająca endpoint REST dla klienta w celu pobrania listy produktów danego użytkownika	<ul style="list-style-type: none"> <li>ProductManager.getMyProducts</li> <li>ProductFacade.findByOwnerLogin</li> </ul>	<ul style="list-style-type: none"> <li>Użytkownik</li> </ul>	<ul style="list-style-type: none"> <li>main</li> <li>myAccount</li> <li>changeMyPassword</li> <li>products</li> <li>makeOrder</li> <li>myProducts</li> <li>addProduct</li> <li>ownOrders</li> </ul>
23	MOP.9	Dominik Lange, Filip Różański	Metoda REST API służąca do pobierania wszystkich kategorii produktów	<ul style="list-style-type: none"> <li>CategoryManager.getAllCategories</li> <li>CategoryFacade.findAll</li> </ul>	<ul style="list-style-type: none"> <li>Moderator</li> </ul>	<ul style="list-style-type: none"> <li>main</li> <li>accounts</li> <li>statistics</li> <li>myAccount</li> <li>changeMyPassword</li> <li>register</li> <li>products</li> <li>makeOrder</li> <li>myProducts</li> <li>addProduct</li> <li>orders</li> <li>categories</li> <li>ownOrders</li> </ul>

						<ul style="list-style-type: none"> <li>• shippingMethods</li> <li>• addShippingMethod</li> </ul>
24	MOP.10	Filip Różański	Metoda REST API służąca do aktywowania /deaktywowania kategorii produktów podanym numerze	<ul style="list-style-type: none"> <li>• CategoryManager.activateCategory <ul style="list-style-type: none"> <li>• CategoryFacade.find</li> <li>• CategoryFacade.edit</li> </ul> </li> <li>• CategoryManager.deactivateCategory <ul style="list-style-type: none"> <li>• CategoryFacade.find</li> <li>• CategoryFacade.edit</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Moderator</li> </ul>	<ul style="list-style-type: none"> <li>• main</li> <li>• accounts</li> <li>• statistics</li> <li>• myAccount</li> <li>• changeMyPassword</li> <li>• register</li> <li>• products</li> <li>• makeOrder</li> <li>• myProducts</li> <li>• addProduct</li> <li>• orders</li> <li>• categories</li> <li>• ownOrders</li> <li>• shippingMethods</li> <li>• addShippingMethod</li> </ul>
25	MOZ.1	Piotr Biernacki	Metoda udostępniająca endpoint REST dla klienta w celu wykonania zamówienia produktu	<ul style="list-style-type: none"> <li>• OrderManager.makeOrder <ul style="list-style-type: none"> <li>• ProductFacade.find</li> <li>• UnitFacade.find</li> <li>• AccountFacade.findByLogin</li> <li>• ShippingMethodFacade.find</li> <li>• OrderStatusFacade.find</li> <li>• OrderShippingFacade.create</li> <li>• OrderFacade.create</li> <li>• OrderProductsFacade.create</li> <li>• ProductFacade.edit</li> <li>• AccountFacade.edit</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Użytkownik</li> </ul>	<ul style="list-style-type: none"> <li>• main</li> <li>• myAccount</li> <li>• changeMyPassword</li> <li>• products</li> <li>• makeOrder</li> <li>• myProducts</li> <li>• addProduct</li> <li>• ownOrders</li> </ul>
26	MOZ.2	Agata Kania, Michał Malec	Metoda udostępniająca endpoint REST dla klienta w celu pobrania wszystkich metod wysyłki	<ul style="list-style-type: none"> <li>• ShippingManager.getAllMethods <ul style="list-style-type: none"> <li>• ShippingFacade.findAll</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Moderator</li> </ul>	<ul style="list-style-type: none"> <li>• main</li> <li>• accounts</li> <li>• statistics</li> <li>• myAccount</li> <li>• changeMyPassword</li> <li>• register</li> <li>• products</li> <li>• makeOrder</li> <li>• myProducts</li> <li>• addProduct</li> <li>• orders</li> <li>• categories</li> <li>• ownOrders</li> <li>• shippingMethods</li> <li>• addShippingMethod</li> </ul>
27	MOZ.3	Filip Różański	Dodawanie nowych metod wysyłki	<ul style="list-style-type: none"> <li>• ShippingManager.addShippingMethod <ul style="list-style-type: none"> <li>• AccountFacade.findByLogin</li> <li>• ShippingFacade.create</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Moderator</li> </ul>	<ul style="list-style-type: none"> <li>• main</li> <li>• accounts</li> <li>• statistics</li> <li>• myAccount</li> <li>• changeMyPassword</li> <li>• register</li> <li>• products</li> <li>• makeOrder</li> <li>• myProducts</li> <li>• addProduct</li> <li>• orders</li> <li>• categories</li> <li>• ownOrders</li> <li>• shippingMethods</li> <li>• addShippingMethod</li> </ul>

28	MOZ.4	Agata Kania	Metoda REST API służąca do aktywacji /deaktywacji metody wysyłki	<ul style="list-style-type: none"> <li>ShippingManager. activateShippingMethod <ul style="list-style-type: none"> <li>ShippingFacade.find</li> <li>ShippingFacade.edit</li> </ul> </li> <li>ShippingManager. deactivateShippingMethod <ul style="list-style-type: none"> <li>ShippingFacade.find</li> <li>ShippingFacade.edit</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Moderator</li> </ul>	<ul style="list-style-type: none"> <li>main</li> <li>accounts</li> <li>statistics</li> <li>myAccount</li> <li>changeMyPassw ord</li> <li>register</li> <li>products</li> <li>makeOrder</li> <li>myProducts</li> <li>addProduct</li> <li>orders</li> <li>categories</li> <li>ownOrders</li> <li>shippingMethods</li> <li>addShippingMet hod</li> </ul>
29	MOZ.5	Dominik Lange	Metoda umożliwiająca edycję stanu zamówienia	<ul style="list-style-type: none"> <li>OrderManager.getOrder1ById</li> <li>OrderManager.getOrderStatusById</li> <li>OrderManager.setOrderStatus <ul style="list-style-type: none"> <li>OrderFacade.edit</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Użytkownik</li> </ul>	<ul style="list-style-type: none"> <li>main</li> <li>myAccount</li> <li>changeMyPassw ord</li> <li>products</li> <li>makeOrder</li> <li>myProducts</li> <li>addProduct</li> <li>ownOrders</li> </ul>
30	MOZ.6	Dominik Lange, Michał Malec	Metoda udostępniająca endpoint REST dla klienta w celu pobrania wszystkich zamówień	<ul style="list-style-type: none"> <li>OrderManager.getAllOrders <ul style="list-style-type: none"> <li>OrderFacade.findAll</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Moderator</li> </ul>	<ul style="list-style-type: none"> <li>main</li> <li>accounts</li> <li>statistics</li> <li>myAccount</li> <li>changeMyPassw ord</li> <li>register</li> <li>products</li> <li>makeOrder</li> <li>myProducts</li> <li>addProduct</li> <li>orders</li> <li>categories</li> <li>ownOrders</li> <li>shippingMethods</li> <li>addShippingMet hod</li> </ul>
31	MOZ.7	Dominik Lange, Michał Malec	Metoda udostępniająca endpoint REST dla klienta w celu pobrania zamówień jako kupujący /sprzedający	<ul style="list-style-type: none"> <li>OrderManager.getAllByBuyer <ul style="list-style-type: none"> <li>OrderFacade.findByBuyerLogin</li> </ul> </li> <li>OrderManager.getAllBySeller <ul style="list-style-type: none"> <li>OrderFacade.findBySellerLogin</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Użytkownik</li> </ul>	<ul style="list-style-type: none"> <li>main</li> <li>myAccount</li> <li>changeMyPassw ord</li> <li>products</li> <li>makeOrder</li> <li>myProducts</li> <li>addProduct</li> <li>ownOrders</li> </ul>
32	MOZ.8	Dominik Lange	Metoda udostępniająca endpoint REST dla klienta w celu pobrania zamówień jako kupujący /sprzedający	<ul style="list-style-type: none"> <li>OrderManager.getAllByBuyer <ul style="list-style-type: none"> <li>OrderFacade.findByBuyerLogin</li> </ul> </li> <li>OrderManager.getAllBySeller <ul style="list-style-type: none"> <li>OrderFacade.findBySellerLogin</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Moderator</li> <li>Użytkownik</li> </ul>	<ul style="list-style-type: none"> <li>main</li> <li>myAccount</li> <li>changeMyPassw ord</li> <li>products</li> <li>makeOrder</li> <li>myProducts</li> <li>addProduct</li> <li>ownOrders</li> </ul>

## 21 Zmiany - projekt koncowy

### Zmiana uprawnień użytkowników bazodanowych ssbd01mop i ssbd01moz

Strona poddana modyfikacjom: [4 Użytkownicy bazodanowi 2 pkt](#)

Dodanie użytkownikom bazodanowym ssbd01mop i ssbd01moz uprawnień UPDATE na tabeli *account*. Dodanie użytkownikowi bazodanowemu ssbd01mop uprawnień DELETE na tabeli *product*. Modyfikacja innych uprawnień.

### Modyfikacja struktury tabel *order*, *shipping\_method*

Strona poddana modyfikacjom: [3 Struktury relacyjnej bazy danych 2 pkt](#)

Usunięto kolumny *order\_number*, *is\_paid* w tabeli *order*. Dodano kolumnę *created\_by* w tabeli *shipping\_method* będącą kluczem obcym, odnoszącym się do tabeli *account*.

### Modyfikacja zestawienia klas encji, tabeli krzyżowej

Strona poddana modyfikacjom: [5 Identyfikacja obiektów encji](#)

*Tabela1.* Zmodyfikowano wiersz z klasą Encyjną *Account*.

*Tabela1.* Dodano klasę encyjną *ArchivalPassword*.

*Tabela4.* Zmodyfikowano tabelę krzyżową.

### Modyfikacja opisu przypadków użycia

Strona poddana modyfikacjom: [Strona tytułowa](#)

### Zaktualizowano diagramy przypadków użycia dla aktorów Moderator i Użytkownik.

Dodano MOP.8 do diagramu przypadków użycia aktora Użytkownik.

Usunięto MOZ.2 i MOZ.3 z diagramu przypadków użycia aktora Użytkownik.

Dodano MOP.9, MOP.10, MOZ.2 i MOZ.3 do diagramu przypadków użycia aktora Moderator.

Zmieniono nazwę oraz przypadku użycia MOP.7

Zmieniono nazwę i definicję przypadku użycia MOZ.4

Zaktualizowano scenariusze główne i scenariusze błędów dla PU: MOK.3, MOK.4, MOK.5, MOK.12, MOP.5, MOP.7, MOP.8, MOP.9, MOP.10, MOZ.1, MOZ.2, MOZ.3, MOZ.4, MOZ.6, MOZ.7, MOZ.8