

1. Sprawozdanie wstępne (15 pkt)	2
1.1 1 Dziedzina wykorzystania 1 pkt	3
1.2 2 Moduły funkcjonalne, aktorzy, przypadki użycia 3 pkt	5
1.3 3 Struktury relacyjnej bazy danych 2 pkt	20
1.4 4 Użytkownicy bazodanowi 2 pkt	25
1.5 5 Identyfikacja obiektów encji 2 pkt	26
1.6 6 Diagram klas encji 1 pkt	30
1.7 7 Diagramy sekwencji oraz identyfikacja transakcji bazodanowych 3 pkt	31
1.8 8 Konfiguracja uwierzytelniania w aplikacji 1 pkt	42

Sprawozdanie wstępne (15 pkt)

Sprawozdanie wstępne

1 Dziedzina wykorzystania 1 pkt

Przeznaczenie aplikacji

- Aplikacja ma za zadanie zarządzanie apteką internetową, klienci zakupić mogą produkty medyczne, farmaceuci obsługują dostawy oraz zatwierdzają zamówienia wymagające recepty. Klienci apteki mogą zakładać konta oraz zarządzać nimi. Farmaceuta, by uzyskać konto musi mieć je utworzone przez administratora, następnie może on je edytować. Klient może przeglądać produkty, dodawać je do koszyka oraz zakupić je, w razie wymaganej recepty składana jest ona razem z zamówieniem. Farmaceuta może dodawać nowe produkty, aktualizować ich stan, zatwierdzać zamówienia wymagające recepty, dodawać kategorie. Aplikacja ma za zadanie umożliwić klientom skorzystanie z usług apteki bez wychodzenia z domu.
- Konkurencja zachodzi dla następujących zasobów: leków (Medication), zamówień (Order) oraz kont (Account).
- Aplikacja powinna zapewnić możliwość szybkiego i efektywnego składania i obsługi zamówień, w tym celu musi z minimalnym opóźnieniem przetwarzać wiele współbieżnych zapytań od różnych użytkowników i dotyczących tych samych zasobów, przy zachowaniu spójności danych. Są to cechy charakterystyczne modelu OLTP.
- Aplikacja jest bezstanowa, komunikacja pomiędzy warstwami prezentacji i logiki biznesowej odbywa się poprzez REST API.
- Stos technologiczny dla warstwy logiki biznesowej:
 - Java wersja 17
 - EJB wersja 3.2
 - Lombok wersja 1.18.26
 - Hibernate wersja 6.2.0.CR4
 - Payara wersja 6.2023.2
 - PostgreSQL wersja 15.3

Słownik nazw obiektów w modelu danych

nazwa	opis	przechowywane dane	zależności
AccessLevel	obiekt bazowy obiektów przechowujących dane do	rola (Role)	jest w relacji wiele-jeden z obiektem Account
AdminData	reprezentuje poziom dostępu administratora		powiązany z Account ponieważ dziedziczy z AccessLevel
ChemistData	reprezentuje poziom dostępu farmaceuty	numer licencji	powiązany z Account ponieważ dziedziczy z AccessLevel
PatientData	reprezentuje poziom dostępu pacjenta	imię, nazwisko, pesel, nr. telefonu, PESEL i NIP	powiązany z Account ponieważ dziedziczy z AccessLevel
Account	reprezentuje konto użytkownika	<ul style="list-style-type: none">• login• hash hasła• czy konto jest aktywne• czy konto jest zarejestrowane	obiekt jest w relacji jeden-wiele z obiektami AccessLevel, przy czym może zawierać maksymalnie jedną instancję z każdej podklasy
Medication	reprezentuje lek	<ul style="list-style-type: none">• nazwa• kategoria (Category)• cena• ilość na stanie	jest w relacji wiele-jeden z Category, i jeden-wiele z OrderMedication oraz OrderShipment
Order	reprezentuje zamówienie	<ul style="list-style-type: none">• wiele leków wraz z ich ilościami (poprzez ShipmentMedication)• datę zamówienia• odnośnik do pacjenta (klasa PatientData)• odnośnik do farmaceuty (klasa ChemistData)• informację o tym czy zamówienie znajduje się w kolejce.	jest w relacji jeden-wiele z prescription i OrderMedication, oraz wiele-wiele z PatientData i ChemistData
OrderMedication	reprezentuje pozycję w zamówieniu	lek (Medication) oraz jego ilość	jest w relacji wiele-jeden z Order oraz z Medication
Prescription	reprezentuje receptę	nr. recepty	jest w relacji wiele-jeden z Order
Shipment	reprezentuje dostawę do apteki	datę dostawy oraz wiele leków wraz z ilościami (poprzez ShipmentMedication)	jest w relacji jeden-wiele z ShipmentMedication
ShipmentMedication	reprezentuje pozycję w dostawie	lek (Medication) oraz jego ilość	jest w relacji wiele-jeden z Shipment oraz z Medication
Category	reprezentuje kategorię leku	angielska i polska nazwa oraz informacja o tym czy leki z danej kategorii są na receptę	jest w relacji jeden-wiele z Medication

Słownik pozostałych pojęć i obiektów

nazwa	opis	wartości
Role	enum zawierający możliwe poziomy dostępu wraz z polską nazwą	"ADMIN", "PATIENT", "CHEMIST"

2 Moduły funkcjonalne, aktorzy, przypadki użycia 3 pkt

Lista modułów funkcjonalnych

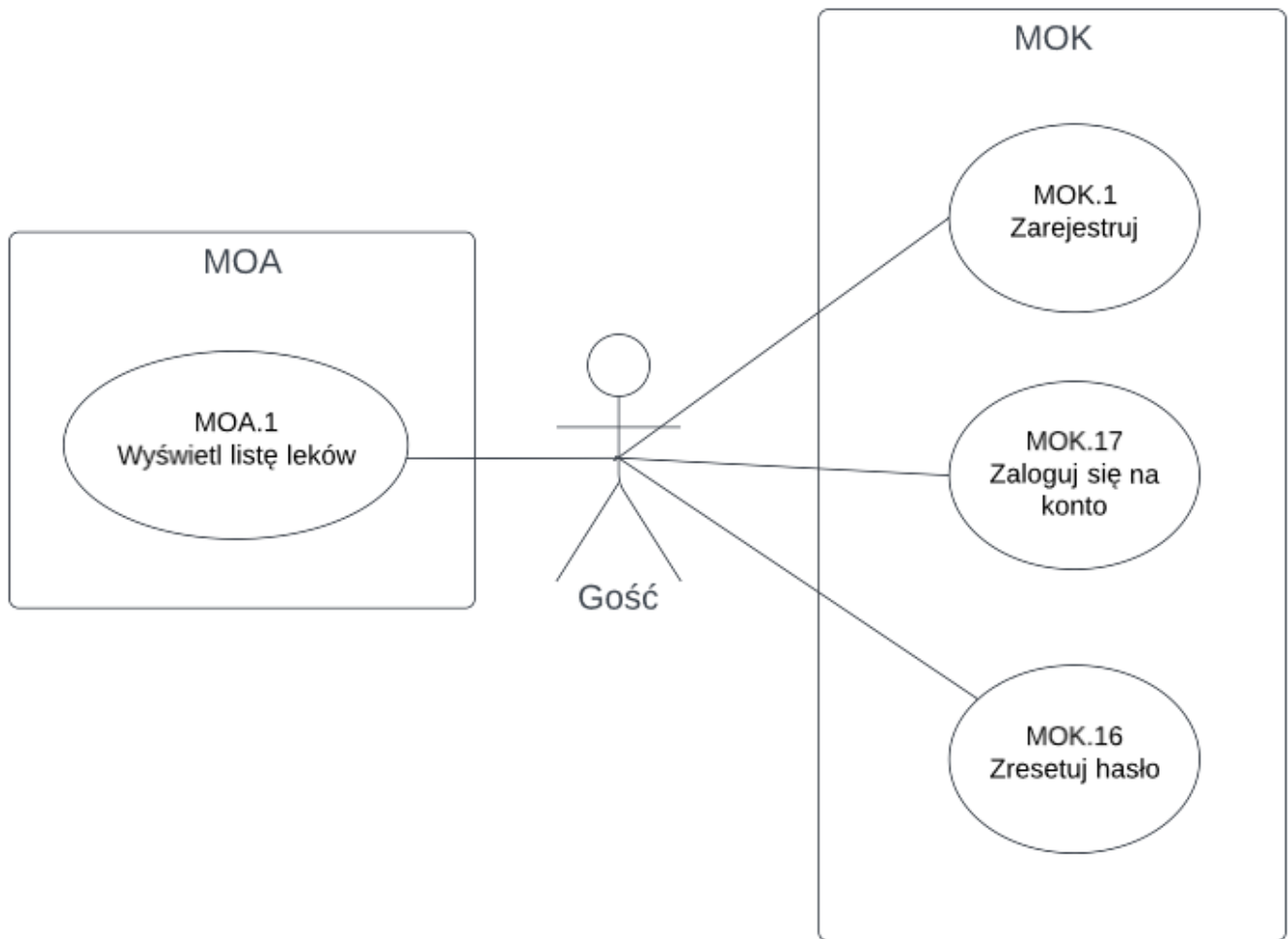
- MOK (Moduł Obsługi Klienta)
- MOA (Moduł Obsługi Apteki)

Lista poziomów dostępu (aktorów)

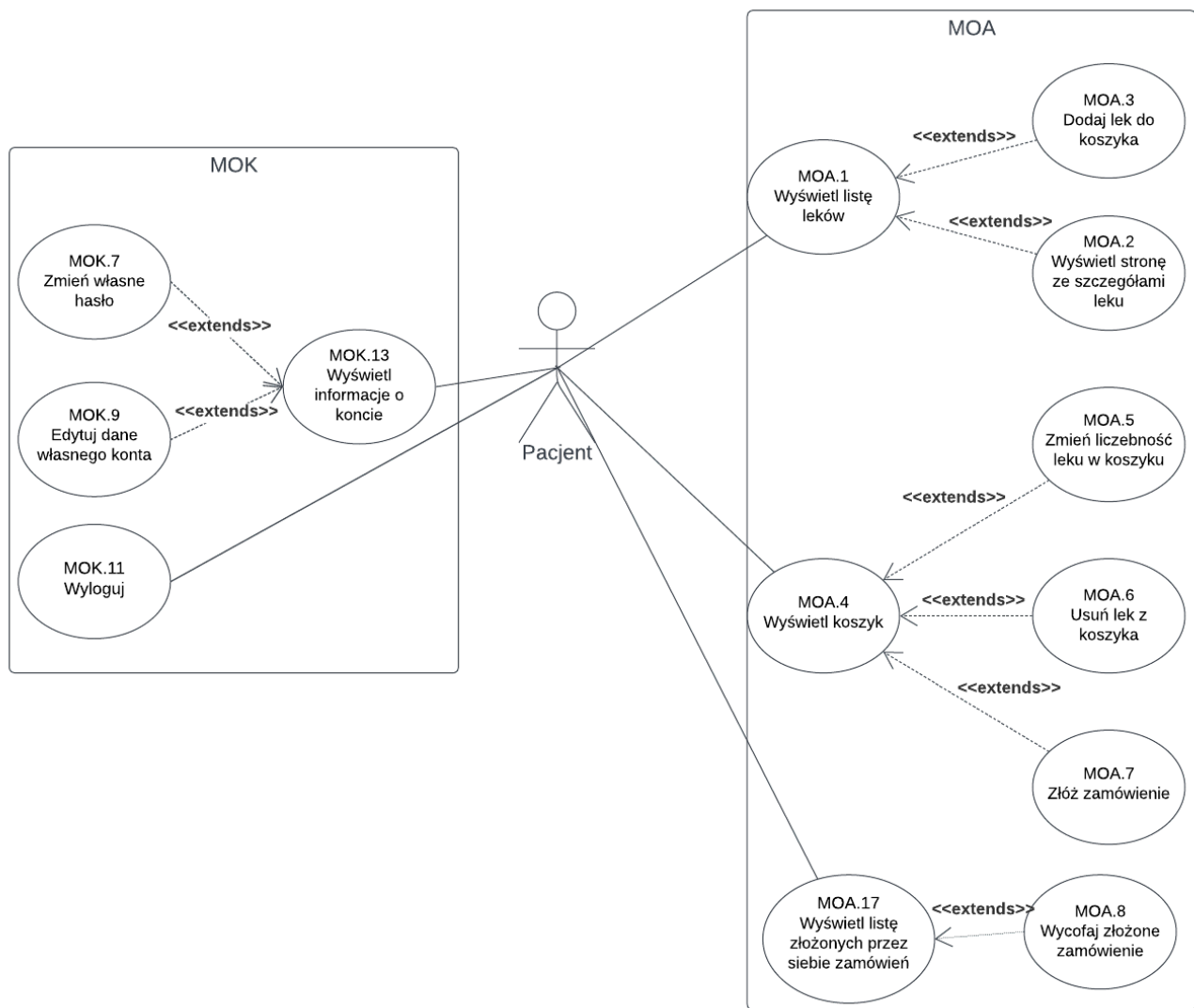
Nazwa poziomu dostępu	Charakteryzacja roli poziomu dostępu w modelu biznesowym aplikacji	Sekwencja czynności prowadzących do utworzenia funkcjonującego konta o danym poziomie dostępu	Wykluczanie
Gość	Wszyscy użytkownicy, którzy mogą wejść na stronę internetową	-	Wyklucza powiązanie z innymi poziomami dostępu
Pacjent	Zwykły użytkownik korzystający z usług apteki internetowej, może zamawiać leki za pośrednictwem strony internetowej	Pacjent samodzielnie rejestruje się podając wymagane dane. Po poprawnej rejestracji system wysyła email z prośbą o potwierdzenie, po którego potwierdzeniu, użytkownik uzyskuje dostęp do konta	Wyklucza powiązanie z poziomem dostępu Farmaceuta
Administrator	Poziom dostępu odpowiedzialny za obsługę kont użytkowników	Tworzony z poziomu bazy danych	Brak
Farmaceuta	Zajmuje się zarządzaniem zamówieniami w aptece internetowej	Konto tworzone przez administratora	Wyklucza powiązanie z poziomem dostępu Pacjent

Diagramy przypadków użycia

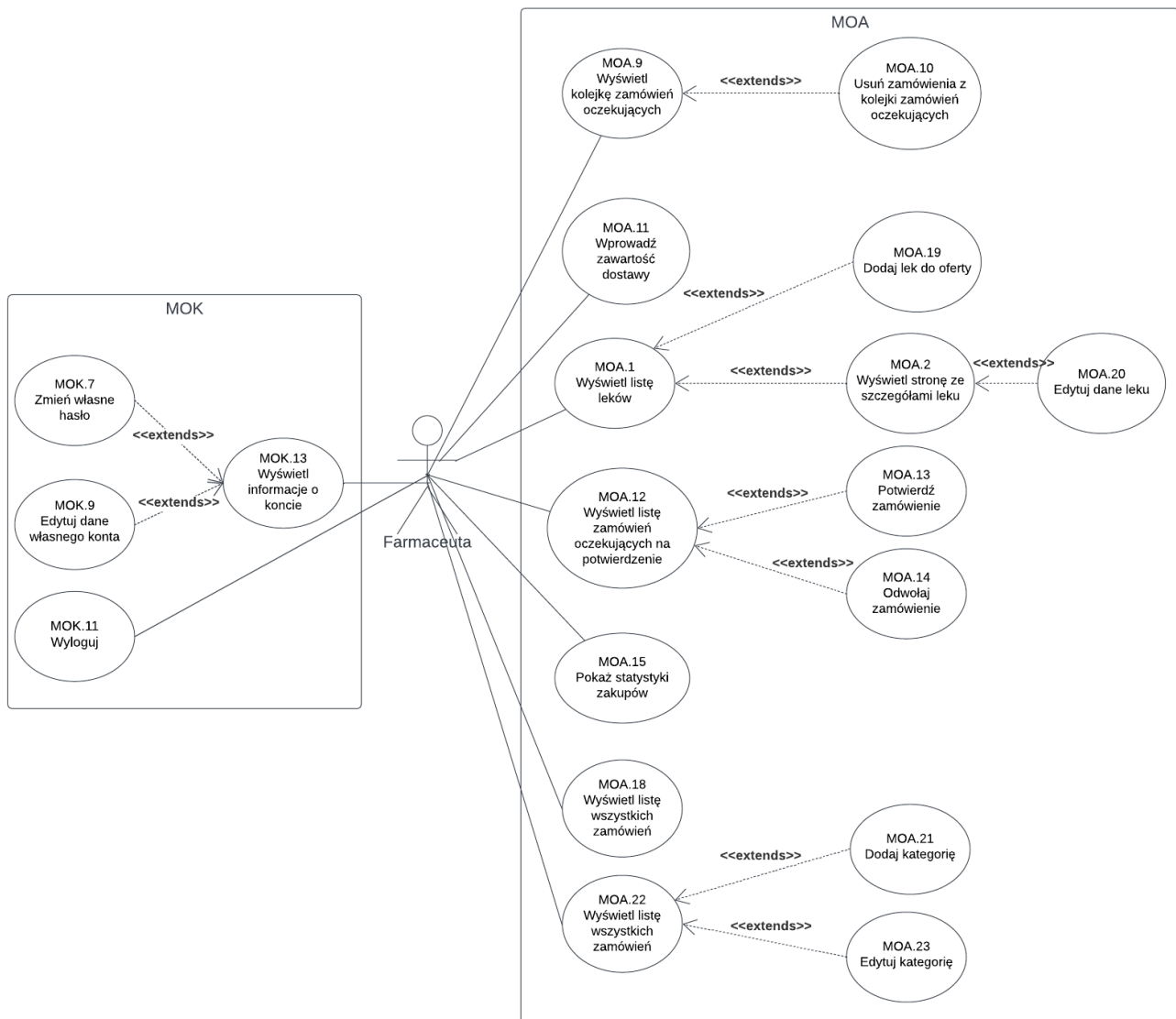
- Forma: odrębne diagramy użycia dla wszystkich poziomów dostępu (aktorów) aplikacji, włącznie z użytkownikiem anonimowym tzw. Gość (nie uwierzytelnionym) oraz System, który nigdy nie może być przypisany do kont użytkowników.
- Na każdym diagramie należy grupować przypadki użycia należące do jednego modułu funkcjonalnego, dla każdego przypadku użycia należy podać akronim modułu i unikalny numer względem tego modułu.
- Diagramy muszą być niezależne. Jeżeli przypadek użycia jest dostępny dla wielu poziomów dostępu, powinien znaleźć się na wszystkich stosownych diagramach.
- Jeżeli dla poszczególnych przypadków użycia występuje relacja rozszerzania, uszczegóławiania to powinna ona być uwzględniona na diagramie.
- Dla każdego przypadku użycia oprócz scenariusza głównego należy wymienić i krótko opisać wszystkie zidentyfikowane scenariusze błędów, a dla każdego z nich wyraźnie należy zaznaczyć przyczynę i lokalizację wystąpienia błędu lub problemu z przetwarzaniem.



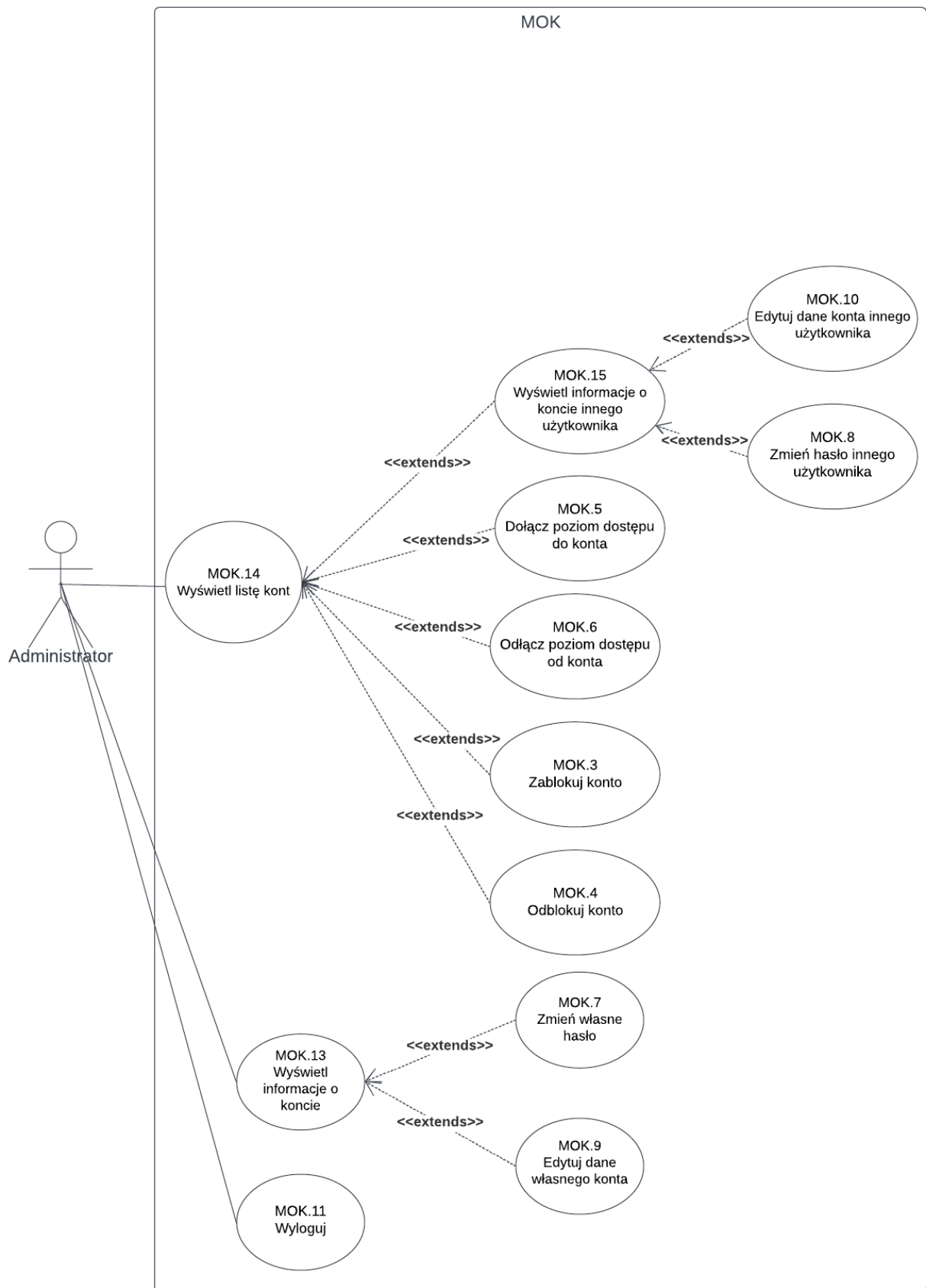
1,2.1. Diagram przypadków użycia dla aktora Gość



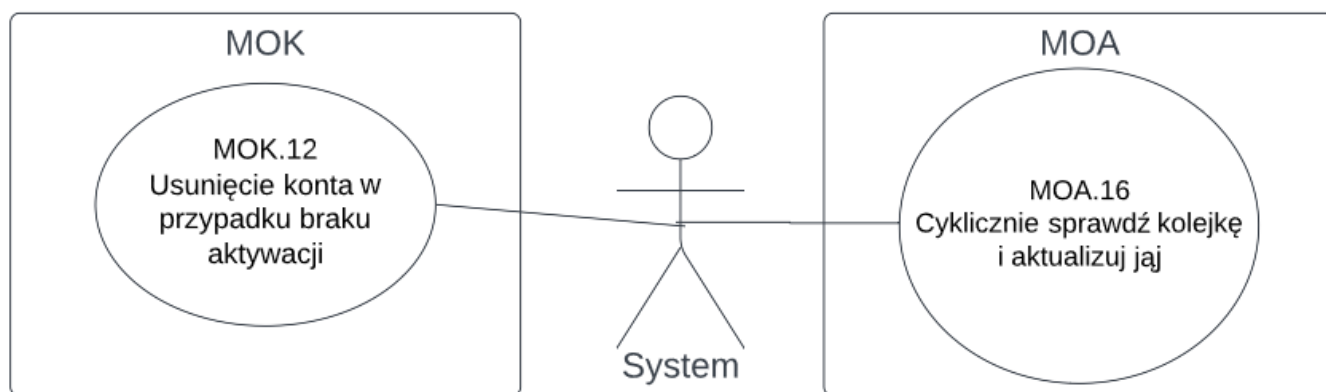
1,2.2. Diagram przypadków użycia dla aktora Pacjent



1,2.3. Diagram przypadków użycia dla aktora Farmaceuta



1,2.4. Diagram przypadków użycia dla aktora Administrator



1,2.4. Diagram przypadków użycia dla aktora System

Nazwa przypadku użycia	MOK.1 Zarejestruj
Aktorzy	Gość
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik wybiera opcję zarejestrowania się w systemie 2. Aplikacja wyświetla formularz rejestracji 3. Użytkownik podaje niezbędne do rejestracji dane i zatwierdza formularz 4. Aplikacja potwierdza zgodność podanych danych 5. Aplikacja dodaje konto nowego użytkownika 6. Aplikacja wysyła mail aktywacji na skrzynkę pocztową użytkownika 7. Aplikacja wyświetla komunikat o poprawnym otwarciu konta 8. Użytkownik aktywuje konto poprzez link z maila
Rozszerzenia	4a. Podane dane są niepoprawne i użytkownik musi ponownie podać dane

Nazwa przypadku użycia	MOK.2 Utwórz konto
Aktorzy	Administrator
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik wybiera opcję utwórz konto 2. Aplikacja wyświetla formularz rejestracji 3. Użytkownik wybiera poziom dostępu i podaje niezbędne do rejestracji dane, oraz zatwierdza formularz 4. Aplikacja potwierdza zgodność podanych danych 5. Aplikacja dodaje nowe konto
Rozszerzenia	4a. Podane dane są niepoprawne i użytkownik musi ponownie podać dane

Nazwa przypadku użycia	MOK.3 Zablokuj konto
Aktorzy	Administrator
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik wyszukuje użytkownika na liście i na danej pozycji wybiera opcję zablokowania konta 2. Aplikacja sprawdza czy dany użytkownik nie jest zablokowany 3. Aplikacja blokuje konto 4. Użytkownik otrzymuje powiadomienie o poprawnie wykonanej czynności
Rozszerzenia	2a. Konto zostało już zablokowane - Użytkownik otrzymuje powiadomienie o braku możliwości zablokowania konta

Nazwa przypadku użycia	MOK.4 Odblokuj konto
Aktorzy	Administrator
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik wyszukuje użytkownika na liście i na danej pozycji wybiera opcję odblokowanie konta 2. Aplikacja sprawdza czy dany użytkownik nie jest odblokowany 3. Aplikacja odblokowuje konto 4. Użytkownik otrzymuje powiadomienie o poprawnie wykonanej czynności
Rozszerzenia	brak

Nazwa przypadku użycia	MOK.5 Dołącz poziom dostępu do konta
Aktorzy	Administrator
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik wyszukuje użytkownika na liście i z menu wybiera opcję przydzielania poziomu dostępu 2. Aplikacja pobiera listę poziomów dostępu dla użytkownika 3. Aplikacja wyświetla listę możliwych poziomów dostępu do wyboru 4. Użytkownik wybiera rolę, którą chce dodać z listy, uzupełnia dane roli i zatwierdza formularz 5. Aplikacja wyświetla komunikat aby potwierdzić wykonanie czynności 6. Użytkownik potwierdza dołączenie poziomu dostępu 7. Aplikacja przydziela poziom dostępu do konta 8. Aplikacja wyświetla komunikat o poprawnym przypisaniu dostępu
Rozszerzenia	brak

Nazwa przypadku użycia	MOK.6 Odłącz poziom dostępu od konta
Aktorzy	Administrator
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik wyszukuje użytkownika na liście i z menu wybiera opcję przydzielania poziomu dostępu 2. Aplikacja pobiera listę poziomów dostępu dla użytkownika 3. Aplikacja wyświetla listę możliwych poziomów dostępu do wyboru 4. Użytkownik wybiera rolę, którą chce odłączyć z listy i zatwierdza swój wybór 5. Aplikacja wyświetla komunikat aby potwierdzić wykonanie czynności 6. Użytkownik potwierdza dołączenie poziomu dostępu 7. Aplikacja przydziela poziom dostępu do konta 8. Aplikacja wyświetla komunikat o poprawnym przypisaniu dostępu
Rozszerzenia	brak

Nazwa przypadku użycia	MOK.7 Zmień własne hasło
Aktorzy	Administrator, Farmaceuta, Pacjent

Scenariusz główny	1. Użytkownik wybiera opcję zmiany własnego hasła 2. Aplikacja wyświetla formularz zmiany hasła 3. Użytkownik wypełnia formularz zmiany hasła i zatwierdza je 4. Aplikacja wyświetla prośbę o potwierdzenie zmiany hasła 5. Użytkownik potwierdza operację 6. Aplikacja potwierdza poprawność nowego hasła 7. Aplikacja weryfikuje wartość starego hasła 8. Aplikacja aktualizuje hasło użytkownika 9. Aplikacja wysyła komunikat o pomyślnej zmianie hasła i wylogowuje użytkownika
Rozszerzenia	5a. Użytkownik anuluje zmianę hasła klikając przycisk "anuluj" 6a. Hasło nie spełnia kryteriów - użytkownik musi wprowadzić hasło jeszcze raz 7a. Stare hasło jest niepoprawne

Nazwa przypadku użycia	MOK.8 Zmień hasło innego użytkownika
Aktorzy	Administrator
Scenariusz główny	1. Użytkownik wybiera opcję zmiany hasła na stronie z informacjami o koncie wybranego użytkownika 2. Aplikacja wyświetla formularz zmiany hasła 3. Użytkownik wypełnia formularz zmiany hasła i zatwierdza je 4. Aplikacja wyświetla prośbę o potwierdzenie zmiany hasła 5. Użytkownik potwierdza operację 6. Aplikacja potwierdza zgodność podanych danych 7. Aplikacja aktualizuje hasło użytkownika 8. Aplikacja wysyła komunikat o pomyślnej zmianie hasła
Rozszerzenia	5a. Użytkownik anuluje zmianę hasła klikając przycisk "anuluj" 6a. Hasło nie spełnia kryteriów - użytkownik musi wprowadzić hasło jeszcze raz - aplikacja wyświetla komunikat o błędzie

Nazwa przypadku użycia	MOK.9 Edytuj dane własnego konta
Aktorzy	Administrator, Farmaceuta, Pacjent
Scenariusz główny	1. Użytkownik wybiera opcję edycji danych 2. Aplikacja wyświetla formularz zmiany danych 3. Użytkownik zmienia dane i zatwierdza formularz 4. Aplikacja wyświetla prośbę o potwierdzenie zmiany danych 5. Użytkownik potwierdza operację 6. Aplikacja potwierdza poprawność danych 7. Aplikacja aktualizuje dane użytkownika 8. Aplikacja wyświetla komunikat o pomyślnej zmianie danych
Rozszerzenia	5a. Podane dane są niepoprawne i użytkownik musi ponownie podać dane - aplikacja wyświetla komunikat o błędzie

Nazwa przypadku użycia	MOK.10 Edytuj dane konta innego użytkownika
------------------------	---

Aktorzy	Administrator
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik wybiera opcję edytowania konta innego użytkownika 2. Aplikacja wyświetla formularz do zmiany danych 3. Użytkownik zmienia dane i zatwierdza formularz 4. Aplikacja wyświetla prośbę o potwierdzenie zmian 5. Użytkownik potwierdza operację 6. Aplikacja potwierdza zgodność podanych danych 7. Aplikacja aktualizuje dane konta, wyświetla komunikat o pomyślnej aktualizacji danych oraz wysyła e-mail informujący o zmianie danych
Rozszerzenia	6a. Wprowadzone dane są niepoprawne - aplikacja wyświetla komunikat o niepoprawności danych

Nazwa przypadku użycia	MOK.11 Wyloguj
Aktorzy	Administrator, Farmaceuta, Pacjent
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik wybiera opcję wylogowania 2. Aplikacja wyświetla komunikat o potwierdzeniu wylogowania 3. Użytkownik potwierdza wylogowanie 4. Aplikacja porzuca token JWT 5. Użytkownik zostaje przekierowany na stronę główną
Rozszerzenia	3a. Użytkownik anuluje wylogowanie

Nazwa przypadku użycia	MOK.12 Usun konto w przypadku braku aktywacji
Aktorzy	System
Scenariusz główny	<ol style="list-style-type: none"> 1. Aplikacja oczekuje określony czas na potwierdzenie rejestracji przez użytkownika 2. Aplikacja usuwa konto jeśli minie określony czas
Rozszerzenia	brak

Nazwa przypadku użycia	MOK.13 Wyświetl informacje o koncie
Aktorzy	Administrator, Farmaceuta, Pacjent
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik wybiera opcję wyświetlenia informacji o swoim koncie 2. Aplikacja wyświetla informacje o koncie użytkownika
Rozszerzenia	brak

Nazwa przypadku użycia	MOK.14 Wyświetl listę kont
Aktorzy	Administrator
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik wybiera opcję wyświetlenia listy kont 2. Aplikacja wyświetla listę wszystkich kont
Rozszerzenia	brak

Nazwa przypadku użycia	MOK.15 Wyświetl informacje o koncie innego użytkownika
Aktorzy	Administrator
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik wyszukuje użytkownika na liście i na danej pozycji wybiera opcję informacji o koncie 2. Aplikacja wyświetla informacje na temat wybranego konta

Rozszerzenia	brak
Nazwa przypadku użycia	MOK.16 Zresetuj hasło
Aktorzy	Gość
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik zgłasza chęć zresetowania hasła 2. Aplikacja wyświetla formularz do wprowadzenia adresu e-mail powiązanego z kontem użytkownika 3. Użytkownik wypełnia i zatwierdza formularz 4. Aplikacja wyszukuje konto o podanym adresie email 5. Aplikacja wysyła na podany adres e-mail link do resetowania hasła 6. Użytkownik wchodzi na stronę podaną w wiadomości email 7. Aplikacja wyświetla formularz związany z ustawieniem nowego hasła 8. Użytkownik wypełnia i zatwierdza formularz 9. Aplikacja sprawdza poprawność podanych w formularzu danych 10. Aplikacja ustawia nową wartość hasła użytkownika 11. Aplikacja wyświetla komunikat o poprawnej zmianie hasła i przekierowuje użytkownika na stronę
Rozszerzenia	<ol style="list-style-type: none"> 4a. Aplikacja wyświetla komunikat o niepoprawnym adresie email 9a. Podane hasła nie spełniają kryteriów - aplikacja wyświetla komunikat o niespełnieniu kryteriów

Nazwa przypadku użycia	MOK.17 Zaloguj się na konto
Aktorzy	Gość
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownika zgłasza chęć zalogowania się 2. Aplikacja wyświetla formularz logowania 3. Użytkownik podaje dane potrzebne do logowania i zatwierdza formularz 4. Aplikacja potwierdza zgodność podanych danych 5. Aplikacja wyświetla stronę główną poziomu dostępu użytkownika
Rozszerzenia	<ol style="list-style-type: none"> 4a. Podane dane są niepoprawne i należy ponownie podać dane 5a. Użytkownik ma przypisany więcej niż jeden poziom dostępu <ol style="list-style-type: none"> 1. Aplikacja wyświetla listę z poziomami dostępu użytkownika 2. Użytkownik wybiera jeden z poziomów dostępu poprzez kliknięcie 3. Aplikacja wyświetla stronę główną wybranego poziomu dostępu

Nazwa przypadku użycia	MOA.1 Wyświetl listę leków
Aktorzy	Gość, Farmaceuta, Pacjent
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik wybiera opcję wyświetlenia listy leków 2. Aplikacja wyświetla listę leków
Rozszerzenia	brak

Nazwa przypadku użycia	MOA.2 Wyświetl stronę ze szczegółami leku
Aktorzy	Farmaceuta, Pacjent
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik wybiera lek z listy i klika na niego 2. Aplikacja przekierowuje użytkownika na stronę wybranego leku

Rozszerzenia	brak
--------------	------

Nazwa przypadku użycia	MOA.3 Dodaj lek do koszyka
Aktorzy	Pacjent
Scenariusz główny	1. Użytkownik wybiera opcję dodania danego leku z listy do zamówienia 2. Aplikacja dodaje lek do zamówienia użytkownika
Rozszerzenia	brak

Nazwa przypadku użycia	MOA.4 Wyświetl koszyk
Aktorzy	Pacjent
Scenariusz główny	1. Użytkownik wybiera opcję wyświetlenia kompletnego zamówienia 2. Aplikacja wyświetla listę produktów dodanych do zamówienia
Rozszerzenia	brak

Nazwa przypadku użycia	MOA.5 Zmień liczebność leku w koszyku
Aktorzy	Pacjent
Scenariusz główny	1. Użytkownik wybiera opcję zmiany ilości leku w koszyku 2. Aplikacja wyświetla formularz zmiany liczebności leku 3. Użytkownik wypełnia i zatwierdza formularz 4. Aplikacja wyświetla prośbę o potwierdzenie zmiany danych 5. Użytkownik potwierdza zmianę danych 6. Aplikacja weryfikuje formularz 7. Aplikacja aktualizuje ilość leku w zamówieniu 8. Aplikacja wyświetla komunikat o poprawnej zmianie ilości leku
Rozszerzenia	5a. Użytkownik anuluje zmianę hasła klikając przycisk "anuluj" 6a. Podana wartość jest niepoprawna - aplikacja wyświetla komunikat o złej wartości

Nazwa przypadku użycia	MOA.6 Usuń lek z koszyka
Aktorzy	Pacjent
Scenariusz główny	1. Użytkownik klika przycisk usuń przy wybranej pozycji z zamówienia 2. Aplikacja wyświetla okno z prośbą potwierdzenia akcji 3. Użytkownik potwierdza usunięcie pozycji z zamówienia 4. Aplikacja usuwa dany lek z koszyka 5. Aplikacja wyświetla komunikat o usunięciu leku z koszyka
Rozszerzenia	brak

Nazwa przypadku użycia	MOA.7 Złóż zamówienie
Aktorzy	Pacjent

Scenariusz główny	1. Użytkownik wybiera opcję złożenia zamówienia 2. Aplikacja wyświetla okno z prośbą potwierdzenia akcji 3. Użytkownik potwierdza złożenie zamówienia 4. Aplikacja potwierdza możliwość wykonania zamówienia 5. Aplikacja wyświetla komunikat o poprawnym złożeniu zamówienia
Rozszerzenia	3a. Użytkownik anuluje zmianę hasła klikając przycisk "anuluj" 4a. Leki w zamówieniu nie są dostępne na stanie, system dodaje zamówienie do kolejki zamówień oczekujących 4b. Zamówienie zawiera leki na receptę, system dodaje zamówienie do listy do zatwierdzenia

Nazwa przypadku użycia	MOA.8 Wycofaj złożone zamówienie
Aktorzy	Farmaceuta
Scenariusz główny	1. Użytkownik wybiera zamówienie z listy swoich zamówień i klika przycisk wycofania zamówienia - jeśli zamówienie jest w kolejce do realizacji 2. Aplikacja wyświetla okno z prośbą potwierdzenia akcji 3. Użytkownik potwierdza wycofanie złożonego zamówienia 4. Aplikacja usuwa zamówienie z kolejki zamówień 5. Aplikacja wyświetla komunikat o poprawnym wycofaniu zamówień
Rozszerzenia	3a. Użytkownik anuluje zmianę hasła klikając przycisk "anuluj"

Nazwa przypadku użycia	MOA.9 Wyświetl kolejkę zamówień oczekujących
Aktorzy	Farmaceuta
Scenariusz główny	1. Użytkownik wybiera opcję wyświetlenia kolejki zamówień oczekujących 2. Aplikacja wyświetla listę zamówień oczekujących
Rozszerzenia	brak

Nazwa przypadku użycia	MOA.10 Usuń zamówienia z kolejki zamówień oczekujących
Aktorzy	Farmaceuta
Scenariusz główny	1. Użytkownik klika przycisk usunięcia wybranego zamówienia z kolejki 2. Aplikacja wyświetla okno z prośbą potwierdzenia akcji 3. Użytkownik potwierdza wycofanie złożonego zamówienia 4. Aplikacja usuwa wybrane zamówienie z kolejki 5. Aplikacja wyświetla komunikat o usunięciu zamówienia z kolejki
Rozszerzenia	3a. Użytkownik anuluje zmianę hasła klikając przycisk "anuluj"

Nazwa przypadku użycia	MOA.11 Wprowadź zawartość dostawy
Aktorzy	Farmaceuta

Scenariusz główny	1. Użytkownik klika przycisk wprowadź zawartość dostawy 2. Aplikacja wyświetla formularz wprowadzania zawartości dostawy 3. Użytkownik wprowadza leki oraz ich liczebność, które przysły w dostawie 4. Użytkownik przesyła formularz wprowadzania zawartości dostawy 5. Aplikacja wyświetla okno z prośbą potwierdzenia akcji 6. Aplikacja zapisuje zawartość dostawy 7. Aplikacja wyświetla komunikat o pomyślnie wykonanej operacji
Rozszerzenia	3a. Użytkownik dodaje za pomocą osobnego formularza dane leków, które doszły w dostawie ale nie ma ich w systemie

Nazwa przypadku użycia	MOA.12 Wyświetl listę zamówień oczekujących na potwierdzenie
Aktorzy	Farmaceuta
Scenariusz główny	1. Użytkownik wybiera opcję wyświetlenia kolejki zamówień oczekujących na potwierdzenie 2. Aplikacja wyświetla listę zamówień oczekujących na potwierdzenie
Rozszerzenia	brak

Nazwa przypadku użycia	MOA.13 Potwierdź zamówienie
Aktorzy	Farmaceuta
Scenariusz główny	1. Użytkownik wybiera z listy zamówienie i klika przy nim przycisk potwierdzenia 2. Aplikacja wyświetla okno z prośbą potwierdzenia akcji 3. Użytkownik potwierdza wycofanie złożonego zamówienia 4. Aplikacja potwierdza zamówienie 5. Aplikacja wyświetla komunikat o potwierdzeniu zamówienia
Rozszerzenia	3a. Użytkownik anuluje zmianę hasła klikając przycisk "anuluj"

Nazwa przypadku użycia	MOA.14 Odwołaj zamówienie
Aktorzy	Farmaceuta
Scenariusz główny	1. Użytkownik wybiera z listy zamówienie i klika przy nim przycisk odwołania 2. Aplikacja wyświetla okno z prośbą potwierdzenia akcji 3. Użytkownik potwierdza wycofanie złożonego zamówienia 4. Aplikacja potwierdza zamówienie 5. Aplikacja wyświetla komunikat o odwołaniu zamówienia
Rozszerzenia	3a. Użytkownik anuluje zmianę hasła klikając przycisk "anuluj"

Nazwa przypadku użycia	MOA.15 Pokaż statystyki zakupów
Aktorzy	Farmaceuta
Scenariusz główny	1. Użytkownik wybiera opcję pokazania statystyki zakupów 2. Aplikacja wyświetla statystyki sprzedaży
Rozszerzenia	brak

Nazwa przypadku użycia	MOA.16 Cyklicznie sprawdź kolejkę i aktualizuj ją
Aktorzy	System

Scenariusz główny	1. Aplikacja porównuje stan magazynowy z kolejką zadań oczekujących na realizację 2. Aplikacja zatwierdza zamówienie, które mogą być skompletowane
Rozszerzenia	brak

Nazwa przypadku użycia	MOA.17 Wyświetl listę złożonych przez siebie zamówień zamówień
Aktorzy	Pacjent
Scenariusz główny	1. Użytkownik zgłasza chęć wyświetlenia listy złożonych zamówień 2. Aplikacja wyświetla listę złożonych zamówień
Rozszerzenia	brak

Nazwa przypadku użycia	MOA.18 Wyświetl listę wszystkich zamówień
Aktorzy	Farmaceuta
Scenariusz główny	1. Użytkownik wybiera opcję wyświetlenia wszystkich zamówień 2. Aplikacja wyświetla listę wszystkich zamówień
Rozszerzenia	brak

Nazwa przypadku użycia	MOA.19 Dodaj lek do oferty
Aktorzy	Farmaceuta
Scenariusz główny	1. Użytkownik wybiera opcję dodania leku do oferty 2. Aplikacja wyświetla formularz dodania leku 3. Użytkownik wypełnia i zatwierdza formularz 4. Aplikacja wyświetla okno z prośbą potwierdzenia akcji 5. Użytkownik zatwierdza akcję 6. Aplikacja sprawdza poprawność danych w formularzu 7. Aplikacja zapisuje dodany lek 8. Aplikacja wyświetla komunikat o poprawnym dodaniu leku
Rozszerzenia	5a. Użytkownik anuluje zmianę hasła klikając przycisk "anuluj" 6a. Dane w formularzu są niepoprawne i trzeba je wprowadzić ponownie - aplikacja wyświetla komunikat o niepoprawnych danych

Nazwa przypadku użycia	MOA.20 Edytuj dane leku
Aktorzy	Farmaceuta
Scenariusz główny	1. Użytkownik wybiera opcję edytowania leku do oferty 2. Aplikacja wyświetla formularz edytowania leku 3. Użytkownik wypełnia i zatwierdza formularz 4. Aplikacja wyświetla okno z prośbą potwierdzenia akcji 5. Użytkownik zatwierdza akcję 6. Aplikacja sprawdza poprawność danych w formularzu 7. Aplikacja aktualizuje edytowany lek 8. Aplikacja wyświetla komunikat o poprawnym edytowaniu leku

Rozszerzenia	5a. Użytkownik anuluje zmianę hasła klikając przycisk "anuluj"
	6a. Dane w formularzu są niepoprawne i trzeba je wprowadzić ponownie - aplikacja wyświetla komunikat o niepoprawnych danych

Nazwa przypadku użycia	MOA.21 Dodaj kategorię
Aktorzy	Farmaceuta
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik wybiera opcję dodania kategorii 2. Aplikacja wyświetla formularz dodania leku 3. Użytkownik wypełnia i zatwierdza formularz 4. Aplikacja wyświetla okno z prośbą potwierdzenia akcji 5. Użytkownik zatwierdza akcję 6. Aplikacja sprawdza poprawność danych w formularzu 7. Aplikacja zapisuje dodany lek 8. Aplikacja wyświetla komunikat o poprawnym dodaniu leku
Rozszerzenia	5a. Użytkownik anuluje zmianę hasła klikając przycisk "anuluj"
	6a. Dane w formularzu są niepoprawne i trzeba je wprowadzić ponownie - aplikacja wyświetla komunikat o niepoprawnych danych

Nazwa przypadku użycia	MOA.22 Wyświetl kategorie
Aktorzy	Farmaceuta
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik wybiera opcję wyświetlenia wszystkie kategorie 2. Aplikacja wyświetla listę wszystkich kategorii
Rozszerzenia	brak

Nazwa przypadku użycia	MOA.23 Edytuj kategorie
Aktorzy	Farmaceuta
Scenariusz główny	<ol style="list-style-type: none"> 1. Użytkownik wybiera opcję edytowania kategorii 2. Aplikacja wyświetla formularz edytowania kategorii 3. Użytkownik wypełnia i zatwierdza formularz 4. Aplikacja wyświetla okno z prośbą potwierdzenia akcji 5. Użytkownik zatwierdza akcję 6. Aplikacja sprawdza poprawność danych w formularzu 7. Aplikacja aktualizuje edytowaną kategorię 8. Aplikacja wyświetla komunikat o poprawnym edytowaniu kategorii
Rozszerzenia	5a. Użytkownik anuluje zmianę hasła klikając przycisk "anuluj"
	6a. Dane w formularzu są niepoprawne i trzeba je wprowadzić ponownie - aplikacja wyświetla komunikat o niepoprawnych danych

3 Struktury relacyjnej bazy danych 2 pkt

Struktury tabel

```
CREATE TABLE access_level
(
    id          BIGINT GENERATED BY DEFAULT AS IDENTITY NOT NULL, → klucz główny → ograniczenie not null, generowany automatycznie z użyciem
strategii GenerationType.IDENTITY
    version     BIGINT, → pole wersji
    creation_date  TIMESTAMP WITHOUT TIME ZONE NOT NULL, → czas utworzenia encji → ograniczenie not null
    modification_date  TIMESTAMP WITHOUT TIME ZONE, → czas modyfikacji encji
    created_by    BIGINT NOT NULL, → klucz obcy powiązany z kontem odpowiedzialnego za stworzenie encji → ograniczenie not null
    modified_by   BIGINT, → klucz obcy powiązany z kontem odpowiedzialnego za ostatnią modyfikację encji
    access_level_role VARCHAR(255) NOT NULL, → nazwa odpowiedniego poziomu dostępu → ograniczenie not null
    active        BOOLEAN DEFAULT TRUE NOT NULL, → wartość logiczna określająca, czy dany poziom dostępu jest aktywny → ograniczenie not null, domyślna
wartość true
    account_id    BIGINT NOT NULL, → klucz obcy powiązany z kontem o określonym poziomie dostępu → ograniczenie not null
    CONSTRAINT pk_access_level PRIMARY KEY (id)
);

ALTER TABLE access_level
    ADD CONSTRAINT FK_ACCESSLEVEL_ON_account FOREIGN KEY (account_id) REFERENCES account (id);

ALTER TABLE access_level
    ADD CONSTRAINT FK_ACCESSLEVEL_ON_CREATED_BY FOREIGN KEY (created_by) REFERENCES account (id);

ALTER TABLE access_level
    ADD CONSTRAINT FK_ACCESSLEVEL_ON_MODIFIED_BY FOREIGN KEY (modified_by) REFERENCES account (id);

CREATE TABLE account
(
    id          BIGINT GENERATED BY DEFAULT AS IDENTITY NOT NULL, -> klucz główny → ograniczenie not null, generowany automatycznie z użyciem
strategii GenerationType.IDENTITY
    version     BIGINT, → pole wersji
    creation_date  TIMESTAMP WITHOUT TIME ZONE NOT NULL, → czas utworzenia encji → ograniczenie not null
    modification_date  TIMESTAMP WITHOUT TIME ZONE, → czas modyfikacji encji
    created_by    BIGINT NOT NULL, → klucz obcy powiązany z kontem odpowiedzialnego za stworzenie encji → ograniczenie not null
    modified_by   BIGINT, → klucz obcy powiązany z kontem odpowiedzialnego za ostatnią modyfikację encji
    login         VARCHAR(255) NOT NULL, → login użytkownika → ograniczenie not null oraz unique
    email         VARCHAR(255) NOT NULL, → adres e-mail użytkownika → ograniczenie not null oraz unique
    password      VARCHAR(255) NOT NULL, → hash hasła użytkownika → ograniczenie not null
    active        BOOLEAN NOT NULL, -> wartość logiczna określająca, czy dane konto jest aktywne → ograniczenie not null
    confirmed     BOOLEAN DEFAULT FALSE NOT NULL, → wartość logiczna określająca, czy dane konto jest potwierdzone → ograniczenie not null,
domyślna wartość false
    language     VARCHAR(255) NOT NULL, → wybrany język → ograniczenie not null
    last_positive_login  TIMESTAMP WITHOUT TIME ZONE, → czas ostatniego poprawnego zalogowania
    last_negative_login  TIMESTAMP WITHOUT TIME ZONE, → czas ostatniego niepoprawnego zalogowania
    logical_address  VARCHAR(255), → adres logiczny, z którego nastąpiła ostatnia próba logowania
    CONSTRAINT pk_account PRIMARY KEY (id)
);

ALTER TABLE account
    ADD CONSTRAINT uc_account_email UNIQUE (email);

ALTER TABLE account
    ADD CONSTRAINT uc_account_login UNIQUE (login);

ALTER TABLE account
    ADD CONSTRAINT FK_account_ON_CREATED_BY FOREIGN KEY (created_by) REFERENCES account(id);

ALTER TABLE account
    ADD CONSTRAINT FK_account_ON_MODIFIED_BY FOREIGN KEY (modified_by) REFERENCES account(id);

CREATE TABLE admin_data
(
    id BIGINT NOT NULL, → klucz główny → ograniczenie not null
    CONSTRAINT pk_admin_data PRIMARY KEY (id)
);
```

```
ALTER TABLE admin_data
  ADD CONSTRAINT FK_ADMINDATA_ON_ID FOREIGN KEY (id) REFERENCES access_level (id);
```

```
CREATE TABLE chemist_data
(
  id          BIGINT    NOT NULL, → klucz główny → ograniczenie not null
  license_number VARCHAR(255) NOT NULL, → numer licencji aptekarza → ograniczenie not null oraz unique
  CONSTRAINT pk_chemist_data PRIMARY KEY (id)
);
```

```
ALTER TABLE chemist_data
  ADD CONSTRAINT FK_chemist_data_ON_ID FOREIGN KEY (id) REFERENCES access_level(id);
```

```
CREATE TABLE patient_data
(
  id          BIGINT    NOT NULL, → klucz główny → ograniczenie not null

  pesel      VARCHAR(255) NOT NULL, → numer PESEL pacjenta → ograniczenie not null oraz unique
  first_name  VARCHAR(255) NOT NULL, → imię pacjenta → ograniczenie not null
  last_name   VARCHAR(255) NOT NULL, → nazwisko pacjenta → ograniczenie not null
  phone_number VARCHAR(255) NOT NULL, → numer telefonu pacjenta → ograniczenie not null
  NIP         VARCHAR(255) NOT NULL, → numer NIP pacjenta → ograniczenie not null
  CONSTRAINT pk_patient_data PRIMARY KEY (id)
);
```

```
ALTER TABLE patient_data
  ADD CONSTRAINT FK_patient_data_ON_ID FOREIGN KEY (id)
```

```
CREATE TABLE medication
(
  id          BIGINT GENERATED BY DEFAULT AS IDENTITY NOT NULL, → klucz główny → ograniczenie not null, generowany automatycznie z użyciem
  strategii GenerationType.IDENTITY
  version     BIGINT, → pole wersji
  creation_date  TIMESTAMP WITHOUT TIME ZONE NOT NULL, → czas utworzenia encji ograniczenie not null
  modification_date  TIMESTAMP WITHOUT TIME ZONE, → czas modyfikacji encji
  created_by    BIGINT NOT NULL, → klucz obcy powiązany z kontem odpowiedzialnego za stworzenie encji → ograniczenie not null
  modified_by   BIGINT, → klucz obcy powiązany z kontem odpowiedzialnego za ostatnią modyfikację encji
  medication_name  VARCHAR(255) NOT NULL, → nazwa leku → ograniczenie not null oraz unique
  stock         INTEGER NOT NULL, → ilość leku w magazynie → ograniczenie not null
  price         DECIMAL NOT NULL, → cena za 1 sztukę leku → ograniczenie not null
  category_id   BIGINT NOT NULL, → klucz obcy powiązany z tabelą category → ograniczenie not null
  CONSTRAINT pk_medication PRIMARY KEY (id)
);
```

```
ALTER TABLE medication
  ADD CONSTRAINT uc_medication_name UNIQUE (name);
```

```
ALTER TABLE medication
  ADD CONSTRAINT FK_MEDICATION_ON_CATEGORY FOREIGN KEY (category_id) REFERENCES category (id);
```

```
ALTER TABLE medication
  ADD CONSTRAINT FK_medication_ON_CREATED_BY FOREIGN KEY (created_by) REFERENCES account (id);
```

```
ALTER TABLE medication
  ADD CONSTRAINT FK_medication_ON_MODIFIED_BY FOREIGN KEY (modified_by) REFERENCES account (id);
```

```

CREATE TABLE category
(
    id          BIGINT GENERATED BY DEFAULT AS IDENTITY NOT NULL, → klucz główny → ograniczenie not null, generowany automatycznie z użyciem
strategii GenerationType.IDENTITY
    version     BIGINT, → pole wersji
    creation_date  TIMESTAMP WITHOUT TIME ZONE NOT NULL, → czas utworzenia encji ograniczenie not null
    modification_date  TIMESTAMP WITHOUT TIME ZONE, → czas modyfikacji encji
    created_by    BIGINT NOT NULL, → klucz obcy powiązany z kontem odpowiedzialnego za stworzenie encji → ograniczenie not null
    modified_by   BIGINT, → klucz obcy powiązany z kontem odpowiedzialnego za ostatnią modyfikację encji
    category_name VARCHAR(255) NOT NULL, → nazwa kategorii → ograniczenie not null oraz unique
    is_on_prescription BOOLEAN NOT NULL, → wartość logiczna określająca, czy dany lek jest na receptę → ograniczenie not null
    CONSTRAINT pk_category PRIMARY KEY (id)
);

ALTER TABLE category
    ADD CONSTRAINT uc_category_category_name UNIQUE (category_name);

ALTER TABLE category
    ADD CONSTRAINT FK_CATEGORY_ON_CREATED_BY FOREIGN KEY (created_by) REFERENCES account (id);

ALTER TABLE category
    ADD CONSTRAINT FK_CATEGORY_ON_MODIFIED_BY FOREIGN KEY (modified_by) REFERENCES account (id);

CREATE TABLE prescription
(
    id          BIGINT GENERATED BY DEFAULT AS IDENTITY NOT NULL, → klucz główny → ograniczenie not null, generowany automatycznie z użyciem
strategii GenerationType.IDENTITY
    version     BIGINT, → pole wersji
    creation_date  TIMESTAMP WITHOUT TIME ZONE NOT NULL, → czas utworzenia encji → ograniczenie not null
    modification_date  TIMESTAMP WITHOUT TIME ZONE, → czas modyfikacji encji
    created_by    BIGINT NOT NULL, → klucz obcy powiązany z kontem odpowiedzialnego za stworzenie encji → ograniczenie not null
    modified_by   BIGINT, → klucz obcy powiązany z kontem odpowiedzialnego za ostatnią modyfikację encji
    patient_data_id BIGINT NOT NULL, → klucz obcy powiązany z tabelą patient_data → ograniczenie not null oraz unique (razem z polem prescription_number)

    prescription_number VARCHAR(255) NOT NULL, → numer recepty → ograniczenie not null oraz unique (razem z polem patient_data_id)
    CONSTRAINT pk_prescription PRIMARY KEY (id)
);

ALTER TABLE prescription
    ADD CONSTRAINT uc_b9d0a510ae29e1a3792d63b06 UNIQUE (patient_data_id, prescription_number);

ALTER TABLE prescription
    ADD CONSTRAINT FK_prescription_ON_CREATED_BY FOREIGN KEY (created_by) REFERENCES account (id);

ALTER TABLE prescription
    ADD CONSTRAINT FK_prescription_ON_MODIFIED_BY FOREIGN KEY (modified_by) REFERENCES account (id);

ALTER TABLE prescription
    ADD CONSTRAINT FK_prescription_ON_PATIENT_DATA FOREIGN KEY (patient_data_id) REFERENCES patient_data (id);

CREATE UNIQUE INDEX patient_data_index ON prescription (patient_data_id);

CREATE TABLE patient_order
(
    id          BIGINT GENERATED BY DEFAULT AS IDENTITY NOT NULL, → klucz główny → ograniczenie not null, generowany automatycznie z użyciem
strategii GenerationType.IDENTITY
    version     BIGINT, → pole wersji
    creation_date  TIMESTAMP WITHOUT TIME ZONE NOT NULL, → czas utworzenia encji → ograniczenie not null
    modification_date  TIMESTAMP WITHOUT TIME ZONE, → czas modyfikacji encji
    created_by    BIGINT NOT NULL, → klucz obcy powiązany z kontem odpowiedzialnego za stworzenie encji → ograniczenie not null
    modified_by   BIGINT, → klucz obcy powiązany z kontem odpowiedzialnego za ostatnią modyfikację encji
    in_queue     BOOLEAN NOT NULL, → wartość logiczna określająca, czy dane zamówienie znajduje się kolejce → ograniczenie not null
    order_date    TIMESTAMP WITHOUT TIME ZONE NOT NULL, → data złożenia zamówienia → ograniczenie not null
    prescription_id BIGINT, → klucz obcy powiązany z tabelą prescription
    patient_data_id BIGINT NOT NULL, → klucz obcy powiązany z tabelą patient_data (pacjent składa zamówienie → ograniczenie not null)
    chemist_data_id BIGINT, → klucz obcy powiązany z tabelą chemist_data (aptekarsz opcjonalnie potwierdzającego zamówienie)
    CONSTRAINT pk_patient_order PRIMARY KEY (id)
);

```

```

ALTER TABLE patient_order
  ADD CONSTRAINT FK_patient_order_ON_CHEMIST_DATA FOREIGN KEY (chemist_data_id) REFERENCES chemist_data (id);

CREATE UNIQUE INDEX chemist_data_index ON patient_order (chemist_data_id);

ALTER TABLE patient_order
  ADD CONSTRAINT FK_patient_order_ON_CREATED_BY FOREIGN KEY (created_by) REFERENCES account (id);

ALTER TABLE patient_order
  ADD CONSTRAINT FK_patient_order_ON_MODIFIED_BY FOREIGN KEY (modified_by) REFERENCES account (id);

ALTER TABLE patient_order
  ADD CONSTRAINT FK_patient_order_ON_PATIENT_DATA FOREIGN KEY (patient_data_id) REFERENCES patient_data (id);

CREATE UNIQUE INDEX patient_data_index ON patient_order (patient_data_id);

ALTER TABLE patient_order
  ADD CONSTRAINT FK_patient_order_ON_prescription FOREIGN KEY (prescription_id) REFERENCES prescription (id);

CREATE UNIQUE INDEX prescription_index ON patient_order (prescription_id);

CREATE TABLE shipment
(
  id          BIGINT GENERATED BY DEFAULT AS IDENTITY NOT NULL, → klucz główny → ograniczenie not null, generowany automatycznie z użyciem
strategii GenerationType.IDENTITY
  version     BIGINT, → pole wersji
  creation_date  TIMESTAMP WITHOUT TIME ZONE  NOT NULL, → czas utworzenia encji
  modification_date  TIMESTAMP WITHOUT TIME ZONE, → czas modyfikacji encji
  created_by    BIGINT  NOT NULL, → klucz obcy powiązany z kontem odpowiedzialnego za stworzenie encji
  modified_by    BIGINT, → klucz obcy powiązany z kontem odpowiedzialnego za ostatnią modyfikację encji
  shipment_date  TIMESTAMP WITHOUT TIME ZONE  NOT NULL, → data przyjęcia dostawy → ograniczenie not null
  CONSTRAINT pk_shipment PRIMARY KEY (id)
);

ALTER TABLE shipment
  ADD CONSTRAINT FK_shipment_ON_CREATED_BY FOREIGN KEY (created_by) REFERENCES account (id);

ALTER TABLE shipment
  ADD CONSTRAINT FK_shipment_ON_MODIFIED_BY FOREIGN KEY (modified_by) REFERENCES account (id);

CREATE TABLE order_medication
(
  id          BIGINT GENERATED BY DEFAULT AS IDENTITY NOT NULL, → klucz główny → ograniczenie not null, generowany automatycznie z użyciem
strategii GenerationType.IDENTITY
  version     BIGINT, → pole wersji
  creation_date  TIMESTAMP WITHOUT TIME ZONE  NOT NULL, → czas utworzenia encji
  modification_date  TIMESTAMP WITHOUT TIME ZONE, → czas modyfikacji encji
  created_by    BIGINT  NOT NULL, → klucz obcy powiązany z kontem odpowiedzialnego za stworzenie encji
  modified_by    BIGINT, → klucz obcy powiązany z kontem odpowiedzialnego za ostatnią modyfikację encji
  patient_order_id    BIGINT  NOT NULL, → klucz obcy powiązany z tabelą patient_order → ograniczenie not null
  medication_id    BIGINT  NOT NULL, → klucz obcy powiązany z tabelą medication → ograniczenie not null
  quantity         INTEGER  NOT NULL, → liczba sztuk danego leku występująca w zamówieniu → ograniczenie not null
  CONSTRAINT pk_order_medication PRIMARY KEY (id)
);

ALTER TABLE order_medication
  ADD CONSTRAINT FK_ORDER_MEDICATION_ON_CREATED_BY FOREIGN KEY (created_by) REFERENCES account (id);

ALTER TABLE order_medication
  ADD CONSTRAINT FK_ORDER_MEDICATION_ON_MEDICATION FOREIGN KEY (medication_id) REFERENCES medication (id);

CREATE UNIQUE INDEX medication_index ON order_medication (medication_id);

ALTER TABLE order_medication
  ADD CONSTRAINT FK_ORDER_MEDICATION_ON_MODIFIED_BY FOREIGN KEY (modified_by) REFERENCES account (id);

ALTER TABLE order_medication
  ADD CONSTRAINT FK_ORDER_MEDICATION_ON_PATIENT_ORDER FOREIGN KEY (patient_order_id) REFERENCES patient_order (id);

```

```
CREATE UNIQUE INDEX order_index ON order_medication (patient_order_id);
```

```
CREATE TABLE shipment_medication
```

```
(  
    id          BIGINT GENERATED BY DEFAULT AS IDENTITY NOT NULL, → klucz główny → ograniczenie not null, generowany automatycznie z użyciem  
    strategii GenerationType.IDENTITY  
    version     BIGINT,  
    creation_date  TIMESTAMP WITHOUT TIME ZONE NOT NULL, → czas utworzenia encji  
    modification_date  TIMESTAMP WITHOUT TIME ZONE, → czas modyfikacji encji  
    created_by    BIGINT NOT NULL, → klucz obcy powiązany z kontem odpowiedzialnego za stworzenie encji  
    modified_by   BIGINT, → klucz obcy powiązany z kontem odpowiedzialnego za ostatnią modyfikację encji  
    shipment_id   BIGINT NOT NULL, → klucz obcy powiązany z tabelą shipment → ograniczenie not null  
    medication_id BIGINT NOT NULL, → klucz obcy powiązany z tabelą medication → ograniczenie not null  
    quantity     INTEGER NOT NULL, → liczba sztuk danego leku przyjęta w dostawie → ograniczenie not null  
    CONSTRAINT pk_shipment_medication PRIMARY KEY (id)  
);
```

```
ALTER TABLE shipment_medication
```

```
ADD CONSTRAINT FK_SHIPMENT_MEDICATION_ON_CREATED_BY FOREIGN KEY (created_by) REFERENCES account (id);
```

```
ALTER TABLE shipment_medication
```

```
ADD CONSTRAINT FK_SHIPMENT_MEDICATION_ON_MEDICATION FOREIGN KEY (medication_id) REFERENCES medication (id);
```

```
CREATE UNIQUE INDEX medication_index ON shipment_medication (medication_id);
```

```
ALTER TABLE shipment_medication
```

```
ADD CONSTRAINT FK_SHIPMENT_MEDICATION_ON_MODIFIED_BY FOREIGN KEY (modified_by) REFERENCES account (id);
```

```
ALTER TABLE shipment_medication
```

```
ADD CONSTRAINT FK_SHIPMENT_MEDICATION_ON_SHIPMENT FOREIGN KEY (shipment_id) REFERENCES shipment (id);
```

```
CREATE UNIQUE INDEX shipment_index ON shipment_medication (shipment_id);
```

Dane inicjujące

```
insert into account(id, version, creation_date, created_by, login, email, password, active, confirmed, language)
```

values (1, 1, now(), 1, 'admin', 'admin01@ssbd.pl', '0dcd26826306868961dfe990e851832c0d475fcafb8a5b319e884b4b711c0b84', true, true, 'PL'); - przeprowadzenie operacji wstawienia do tabeli *account* użytkownika typu admin

insert into access_level(id, version, creation_date, created_by, access_level_role, active, account_id) values (1, 1, now(), 1, 'Admin', true, 1); insert into admin_data(id) values (1); - przeprowadzenie operacji wstawienia do tabeli *access_level* użytkownika ww. użytkownika od identyfikatorze 1.

```
CREATE VIEW glassfish_auth_view AS SELECT account.login,
```

```
account.password,
```

```
al.access_level_role
```

```
FROM access_level AS al
```

```
JOIN account ON al.account_id = account.id
```

```
WHERE account.confirmed = true AND account.active = true AND al.active = true;
```

Utworzony widok (*view*) służy do odczytywania informacji potrzebnych serwerowi *glassfish* do procesu uwierzytelniania i autoryzacji.

4 Użytkownicy bazodanowi 2 pkt

Lista użytkowników bazodanowych

Użytkownik	Hasło	Moduł	Opis
postgres	postgrespassword	-	Użytkownik z prawami administratora.
ssbd01admin	admin	mok, moa	Użytkownik, który jest właścicielem wszystkich tabel w bazie danych.
ssbd01mok	mokpassword	mok	Konto stworzone na potrzeby modułu obsługi kont (mok).
ssbd01moa	moapassword	moa	Konto stworzone na potrzeby modułu obsługi apteki (moa).
ssbd01glassfish	glassfishpassword	-	Użytkownik stworzony na potrzeby utworzenia puli połączeń w celu odczytania widoku i przeprowadzenia przez serwer aplikacyjny procesu uwierzytelnienia.

Tabela uprawnień użytkowników bazodanowych

Nazwa	Typ	postgres	ssbd01admin	ssbd01mok	ssbd01moa	ssbd01glassfish
account	tabela	a,r,w,d	a,r,w,d	a,r,w,d	r	
access_level	tabela	a,r,w,d	a,r,w,d	a,r,w	r	
admin_data	tabela	a,r,w,d	a,r,w,d	a,r,w	r	
category	tabela	a,r,w,d	a,r,w,d	r	a,r,w,d	
chemist_data	tabela	a,r,w,d	a,r,w,d	a,r,w	r	
glassfish_auth_view	widok	a,r,w,d	a,r,w,d			r
medication	tabela	a,r,w,d	a,r,w,d	r	a,r,w,d	
order_medication	tabela	a,r,w,d	a,r,w,d	r	a,r,w,d	
patient_data	tabela	a,r,w,d	a,r,w,d	a,r,w,d	r	
patient_order	tabela	a,r,w,d	a,r,w,d	r	a,r,w,d	
prescription	tabela	a,r,w,d	a,r,w,d	r	a,r,d	
shipment	tabela	a,r,w,d	a,r,w,d	r	a,r,w,d	
shipment_medication	tabela	a,r,w,d	a,r,w,d	r	a,r,w,d	

Legenda:

r – Select, prawo do odczytu

a – Insert, prawo do wstawiania wierszy

w – Update, prawo do aktualizacji wierszy

d – Delete, prawo do usuwania wierszy

Puste pole oznacza brak uprawnień.

5 Identyfikacja obiektów encji 2 pkt

Zestawienie klas encji

Informacje o polach klas encyjnych przedstawione są w następującym formatowaniu:

atrybut klasy encyjnej modelu obiektowego → oznaczenie odpowiadającej kolumny w tabeli → ograniczenia i dodatkowe informacje

Klasa abstrakcyjna *AbstractEntity*:

Jest to klasa, która jest oznaczona adnotacją *@MappedSuperclass* i służy jako baza dla wszystkich encji w systemie. Ta klasa nie reprezentuje żadnej encji, ale każda encja dziedziczy po niej, aby zawierać jej atrybuty.

- a. *private Long version* → *bigint version* → numer wersji zarządzany przez JPA, pole oznaczone adnotacją *@Version*
- b. *private Date creationDate* → *TIMESTAMP WITHOUT TIME ZONE creation_date* → ograniczenie *not null*, ograniczenie JPA *updatable = false*
- c. *private Date modificationDate* → *TIMESTAMP WITHOUT TIME ZONE modification_date*
- d. *private Account createdBy* → *bigint created_by* => ograniczenie *not null*, ograniczenie JPA *updatable = false*, odwołuje się w ramach klucza obcego do kolumny *id* z tabeli *accounts*, związek jeden do jednego
- e. *private Account modifiedBy* → *bigint altered_by* => ograniczenie *not null*, odwołuje się w ramach klucza obcego do kolumny *id* z tabeli *accounts*, związek jeden do jednego

Klasa *Account* → tabela *account* :

Klasa *Account* reprezentuje konto użytkownika w systemie. Przechowuje ona informacje takie jak login, hasło, adres email, poziomy dostępu oraz informacje dotyczące logowania. Celem klasy jest umożliwienie zarządzania kontami użytkowników w systemie.

- a. *Set<AccessLevel> accessLevels* → jest klucz obcy dla kolumny *BIGINT account_id* w tabeli *access_levels*, związek jeden do wielu, relacja dwukierunkowa
- b. *private Long id* → *id BIGINT GENERATED BY DEFAULT AS IDENTITY* → ograniczenie *not null* → klucz główny, generowany automatycznie z użyciem strategii *GenerationType.IDENTITY*
- c. *private String login* → *login VARCHAR(255)* → ograniczenie *not null*, ograniczenie *unique*
- d. *private String email* → *email VARCHAR(255)* → ograniczenie *not null*, ograniczenie *unique*, ograniczenie typu wyrażenie regularne *"^[A-Za-z0-9+_.-]+@(.+)\$"*
- e. *private String password* → *password VARCHAR(255)* → ograniczenie *not null*, ograniczenie typu wyrażenie regularne *"^[0-9a-f]{64}\$"*
- f. *private Boolean active* → *active BOOLEAN* → ograniczenie *not null*
- g. *private Boolean confirmed* → *confirmed BOOLEAN DEFAULT FALSE* → ograniczenie *not null*, domyślna wartość *false*
- h. *private Locale language* → *language VARCHAR(255)* → ograniczenie *not null*
- i. *private Date lastPositiveLogin* → *last_positive_login TIMESTAMP WITHOUT TIME ZONE*
- j. *private Date lastNegativeLogin* → *last_negative_login TIMESTAMP WITHOUT TIME ZONE*
- k. *private String logicalAddress* → *logical_address VARCHAR(255)*

Klasa abstrakcyjna *AccessLevel* → tabela *access_level* :

Klasa bazowa służąca do implementacji różnych poziomów dostępu w systemie. W kontekście bazy danych, ww. klasa pełni rolę tabeli pośredniczącej pomiędzy kontami użytkowników a poziomami dostępu. Wykorzystuje się do tego strategię dziedziczenia JPA *InheritanceType.JOINED*, co oznacza, że każdy poziom dostępu posiada swoją osobną tabelę, a dane związane z kontami użytkowników są przechowywane w tabeli głównej, a informacje o dostęпах użytkowników są przechowywane w tabelach poziomów dostępu, z którymi są powiązane kluczami obcymi.

- a. *private Long id* → *id BIGINT GENERATED BY DEFAULT AS IDENTITY* → ograniczenie *not null* → klucz główny, generowany automatycznie z użyciem strategii *GenerationType.IDENTITY*
- b. *private Role role* → *access_level_role VARCHAR(255)* → kolumna dyskryminatora, jako wartość przyjmuje nazwę klasy danego poziomu dostępu → ograniczenie *not null*, ograniczenie JPA *updatable = false*, do mapowania wykorzystano adnotację *@Enumerated(EnumType.STRING)*
- c. *private Boolean active* → *active BOOLEAN DEFAULT TRUE* → ograniczenie *not null*, domyślna wartość *true*
- d. *private Account account* → *account_id BIGINT* → odwołuje się w ramach klucza obcego do kolumny *id* z tabeli *accounts*, ograniczenie JPA *updatable = false*

Klasa *AdminData* → tabela *admin_data*:

Klasa, która reprezentuje poziom dostępu administratora w systemie. Klasa ta dziedziczy atrybuty i relacje z klasy *AccessLevel*

- a. *private Long id* → *id BIGINT GENERATED BY DEFAULT AS IDENTITY* → ograniczenie *not null* → klucz główny, generowany automatycznie z użyciem strategii *GenerationType.IDENTITY*, klucz dostępny poprzez dziedziczenie z klasy *AccessLevel*

Klasa *ChemistData* → tabela *chemist_data*:

Klasa, która reprezentuje poziom dostępu aptekarza w systemie. Klasa ta dziedziczy atrybuty i relacje z klasy *AccessLevel*

- a. *private Long id* → *id BIGINT GENERATED BY DEFAULT AS IDENTITY* → ograniczenie *not null* → klucz główny, generowany automatycznie z użyciem strategii *GenerationType.IDENTITY*, klucz dostępny poprzez dziedziczenie z klasy *AccessLevel*
- b. *private String licenseNumber* → *license_number VARCHAR(255)* → ograniczenie *not null*

Klasa *PatientData* → tabela *patient_data*:

Klasa, która reprezentuje poziom dostępu pacjenta w systemie. Klasa ta dziedziczy atrybuty i relacje z klasy *AccessLevel*

- a. *private Long id* → *id BIGINT GENERATED BY DEFAULT AS IDENTITY* → ograniczenie *not null* → klucz główny, generowany automatycznie z użyciem strategii *GenerationType.IDENTITY*, klucz dostępny poprzez dziedziczenie z klasy *AccessLevel*
- b. *private String pesel* → *pesel VARCHAR(255)* → ograniczenie *not null*, ograniczenie JPA *updatable = false*, ograniczenie typu wyrażenie regularne *"^[0-9]{11}\$"*
- c. *private String firstName* → *first_name VARCHAR(255)* → ograniczenie *not null*
- d. *private String lastName* → *last_name VARCHAR(255)* → ograniczenie *not null*
- e. *private String phoneNumber* → *phone_number VARCHAR(255)* → ograniczenie *not null*, ograniczenie typu wyrażenie regularne *"^(\d{48})? [0-9]{3} [0-9]{3} [0-9]{3}\$"*
- f. *private String NIP* → *nip VARCHAR(255)* → ograniczenie *not null*, ograniczenie typu wyrażenie regularne *"^[0-9]{3}-[0-9]{3}-[0-9]{2}-[0-9]{2}\$"*

Klasa *Medication* → tabela *medication* :

Klasa *medication* reprezentuje fizyczny obiekt leku. Przechowuje ona informacje takie jak nazwa leku, dostępną ilość w magazynie, cenę za sztukę. Jest powiązana z encją *category* informującą o kategorii leku.

- a. *private Long id* → *id BIGINT GENERATED BY DEFAULT AS IDENTITY* → ograniczenie *not null* → klucz główny, generowany automatycznie z użyciem strategii *GenerationType.IDENTITY*
- b. *private String name* → *medication_name VARCHAR(255)* → ograniczenie *not null*, ograniczenie *unique*
- c. *private Integer stock* → *stock INTEGER* → ograniczenie *not null*, ograniczenie dla minimalnej wartości jako zero
- d. *private BigDecimal price* → *price DECIMAL* → ograniczenie *not null*, ograniczenie dla minimalnej wartości jako zero, ograniczenie dopuszczalnych wartości liczbowych do 10 cyfr przed kropką i 2 cyfr po kropce
- e. *private Category category* → *category_id BIGINT* → odwołuje się w ramach klucza obcego do kolumny *id* z tabeli *category*, ograniczenie *not null*

Klasa *Category* → tabela *category*:

Klasa *Category* reprezentuje kategorię dla danego leku. Oprócz nazwy kategorii zawiera także informację, czy leki o danej kategorii są do kupienia tylko na receptę.

- a. *private Long id* → *id BIGINT GENERATED BY DEFAULT AS IDENTITY* → ograniczenie *not null* → klucz główny, generowany automatycznie z użyciem strategii *GenerationType.IDENTITY*
- b. *private String name* → *category_name VARCHAR(255)* → ograniczenie *not null* oraz *unique*
- c. *private boolean isOnPrescription* → *is_on_prescription BOOLEAN* → ograniczenie *not null*

Klasa *Shipment* → tabela *shipment*:

Klasa *Shipment* reprezentuje dostawę leków i zawiera informacje o dacie wysyłki oraz listę leków przypisanych do przesyłki.

- a. *private Long id* → *id BIGINT GENERATED BY DEFAULT AS IDENTITY* → ograniczenie *not null* → klucz główny, generowany automatycznie z użyciem strategii *GenerationType.IDENTITY*
- b. *private Date shipmentDate* → *shipment_date TIMESTAMP WITHOUT TIME ZONE* → ograniczenie *not null*, ograniczenie JPA *updatable = false*
- c. *private List<ShipmentMedication> shipmentMedications* (relacja zmapowana po stronie klasy *ShipmentMedication*)

Klasa *ShipmentMedication* → tabela *shipment_medication*:

Klasa *ShipmentMedication* stanowi klasę encyjną i nie ma bezpośredniego odzwierciedlenia w logicznym modelu, za to zawiera informacje o jednym leku w jednej dostawie.

- a. *private Long id* → *id BIGINT GENERATED BY DEFAULT AS IDENTITY* → ograniczenie *not null* → klucz główny, generowany automatycznie z użyciem strategii *GenerationType.IDENTITY*
- b. *private Shipment shipment* → *shipment_id BIGINT* → ograniczenie *not null*, odwołuje się w ramach klucza obcego do kolumny *id* z tabeli *shipment*, związek wiele do jednego
- c. *private Medication medication* → *medication_id BIGINT* → ograniczenie *not null*, odwołuje się w ramach klucza obcego do kolumny *id* z tabeli *medication*, związek wiele do jednego
- d. *private Integer quantity* → *INTEGER* → ograniczenie *not null*, ograniczenie dla minimalnej wartości jako jeden

Klasa *Order* → tabela *patient_order*:

Klasa *Order* reprezentuje zamówienie pacjenta w aptece. Zamówienie zawiera informacje o kupowanych lekach, danych pacjenta oraz o ewentualnych receptach i danych przypisanego aptekarza do weryfikacji recepty.

- a. *private Long id* → *id BIGINT GENERATED BY DEFAULT AS IDENTITY* → ograniczenie *not null* → klucz główny, generowany automatycznie z użyciem strategii *GenerationType.IDENTITY*
- b. *private Boolean inQueue* → *in_queue BOOLEAN* → ograniczenie *not null*
- c. *private Date orderDate* → *order_date TIMESTAMP WITHOUT TIME ZONE* → ograniczenie *not null*
- d. *private List<OrderMedication> patient_order_medications* (relacja zmapowana po stronie klasy *OrderMedication*)
- e. *private Prescription prescription* → *prescription_id BIGINT* → odwołuje się w ramach klucza obcego do kolumny *id* z tabeli *prescription*, opcjonalny związek jeden do jednego
- f. *private PatientData patientData* → *patient_data_id BIGINT* → ograniczenie *not null*, odwołuje się w ramach klucza obcego do kolumny *id* z tabeli *patient_data*, związek wiele do jednego
- g. *private ChemistData chemistData* → *chemist_data_id BIGINT* → odwołuje się w ramach klucza obcego do kolumny *id* z tabeli *chemist_data*, opcjonalny związek wiele do jednego

Klasa *OrderMedication* → tabela *order_medication*:

Klasa *OrderMedication* stanowi klasę encyjną i nie ma bezpośredniego odzwierciedlenia w logicznym modelu, za to zawiera informacje o jednym leku w jednym zamówieniu.

- a. *private Long id* → *id BIGINT GENERATED BY DEFAULT AS IDENTITY* → ograniczenie *not null* → klucz główny, generowany automatycznie z użyciem strategii *GenerationType.IDENTITY*
- b. *private Order order* → *patient_order_id BIGINT* → odwołuje się w ramach klucza obcego do kolumny *id* z tabeli *patient_order*, związek wiele do jednego
- c. *private Medication medication* → *medication_id BIGINT* → odwołuje się w ramach klucza obcego do kolumny *id* z tabeli *medication*, związek wiele do jednego
- d. *private Integer quantity* → *quantity INTEGER* → ograniczenie *not null*, ograniczenie dla minimalnej wartości jako jeden

Klasa *Prescription* → tabela *prescription*:

Klasa *Prescription* reprezentuje obiekt recepty, niezbędnej do kupna leków o określonej kategorii. Numer recepty (*prescription_number*) jest unikalny tylko w połączeniu z ID pacjenta, dlatego w tej klasie znajduje się związek wiele do jednego z encją *patient_data*.

- a. *private Long id* → *id BIGINT GENERATED BY DEFAULT AS IDENTITY* → ograniczenie *not null* → klucz główny, generowany automatycznie z użyciem strategii *GenerationType.IDENTITY*
- b. *private PatientData patientData* → *patient_data_id BIGINT* → ograniczenie *not null* oraz *unique* (razem z polem *prescription_number*), odwołuje się w ramach klucza obcego do kolumny *id* z tabeli *patient_data*, związek wiele do jednego
- c. *private String prescriptionNumber* → *prescription_number VARCHAR(255)* → ograniczenie *not null* oraz *unique* (razem z polem *patient_data_id*)

Związki między obiektami klas encji

		Klasa podstawowa										
		AbstractEntity	Account	AccessLevel	AdminData	ChemistData	PatientData	Medication	Shipment	ShipmentMedication	Order	
Klasa powiązana	AbstractEntity											
	Account	<ul style="list-style-type: none"> createdBy; 1-1; PR; NM modifiedBy; 1-1; ; M 		*-1; ; NM								
	AccessLevel		1-*; PR, MR, RM; M									
	AdminData											
	ChemistData										*-1; ; NM	
	PatientData										*-1; ; NM	
	Medication									*-1; PR; NM		*-1
	Shipment									*-1; ; NM		

	ShipmentMedication								1-*; PR, RM; M			
	Order											*-1
	OrderMedication										1-*; PR, RM, MR; M	
	Perscription										1-1; PR; NM	
	Category							*-1; PR, RF; M				

Legenda:

Pole przedstawia związek pomiędzy "klasą podstawową" a "klasą powiązaną", gdzie właścicielem związku jest klasa podstawowa i związek przebiega od klasy podstawowej do klasy powiązanej.

Zawartość pola opisuje charakter związku za pomocą następującego schematu:

1. Liczność związku:

- 1-1 - związek jeden do jednego,
- 1-* - związek jeden do wielu,
- *-1 - związek wiele do jednego.

2. Operacje kaskadowe:

- PR - CascadeType.PERSIST,
- RM - CascadeType.REMOVE,
- MR - CascadeType.MERGE,
- RF - CascadeType.REFRESH.

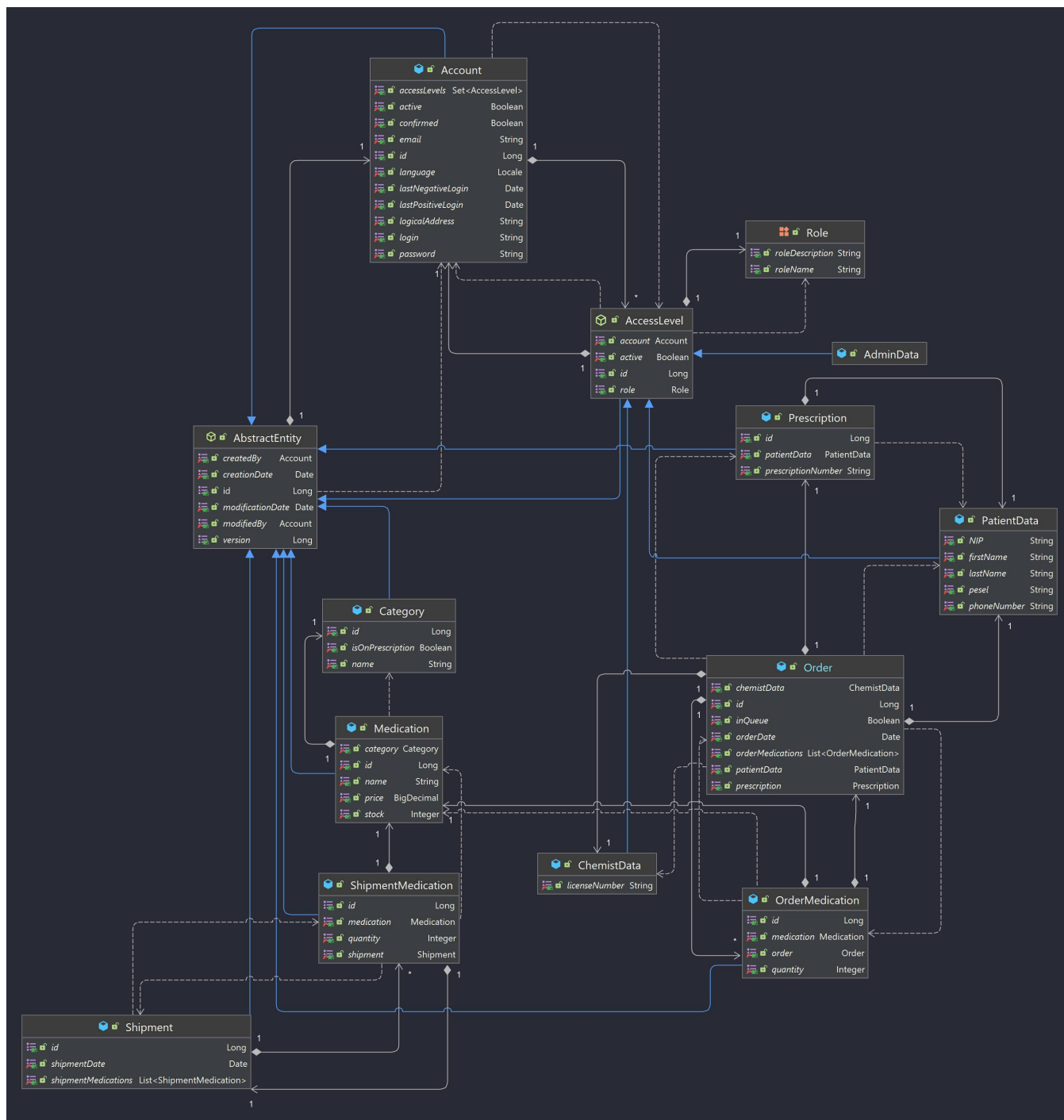
3. Czy relacja może ulec zmianie po utworzeniu:

- M - tak
- NM - nie

W przypadku występowania w klasie podstawowej wielu związków do tej samej klasy powiązanej takowe związki zostały wypunktowane oraz dodano nazwę atrybutu.

6 Diagram klas encji 1 pkt

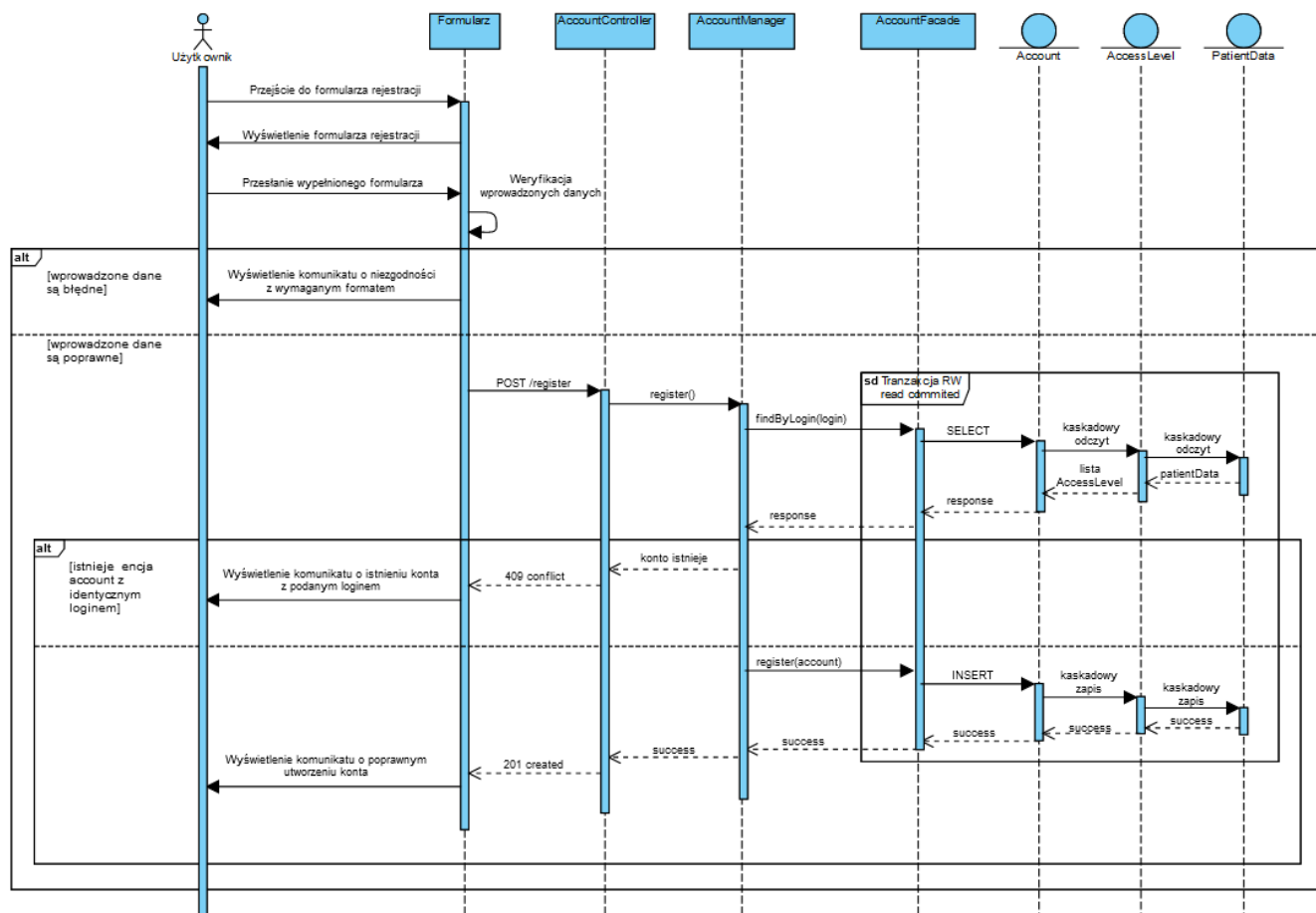
Diagram klas encji



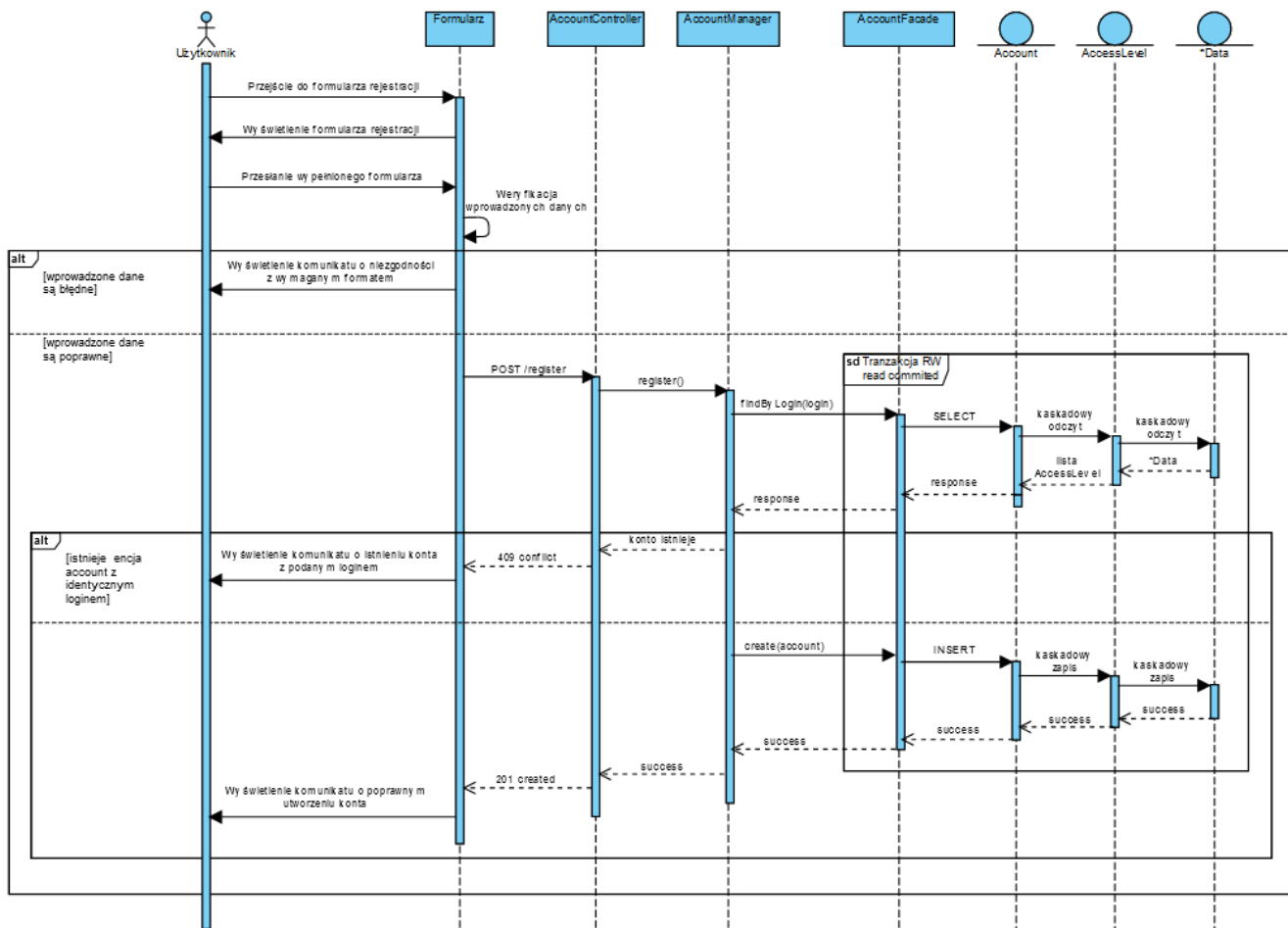
7 Diagramy sekwencji oraz identyfikacja transakcji bazodanowych 3 pkt

Na diagramach, żeby zaoszczędzić powtórzeń, skorzystano z notacji *, która oznacza odwołanie do jednej klasy lub encji spośród: PatientData, ChemistData, AdminData. W nazwach funkcji oznacza odwołanie do jednej z 3 metod zawierających zamiast znaku * 'Patient', 'Chemist' lub 'Admin'.

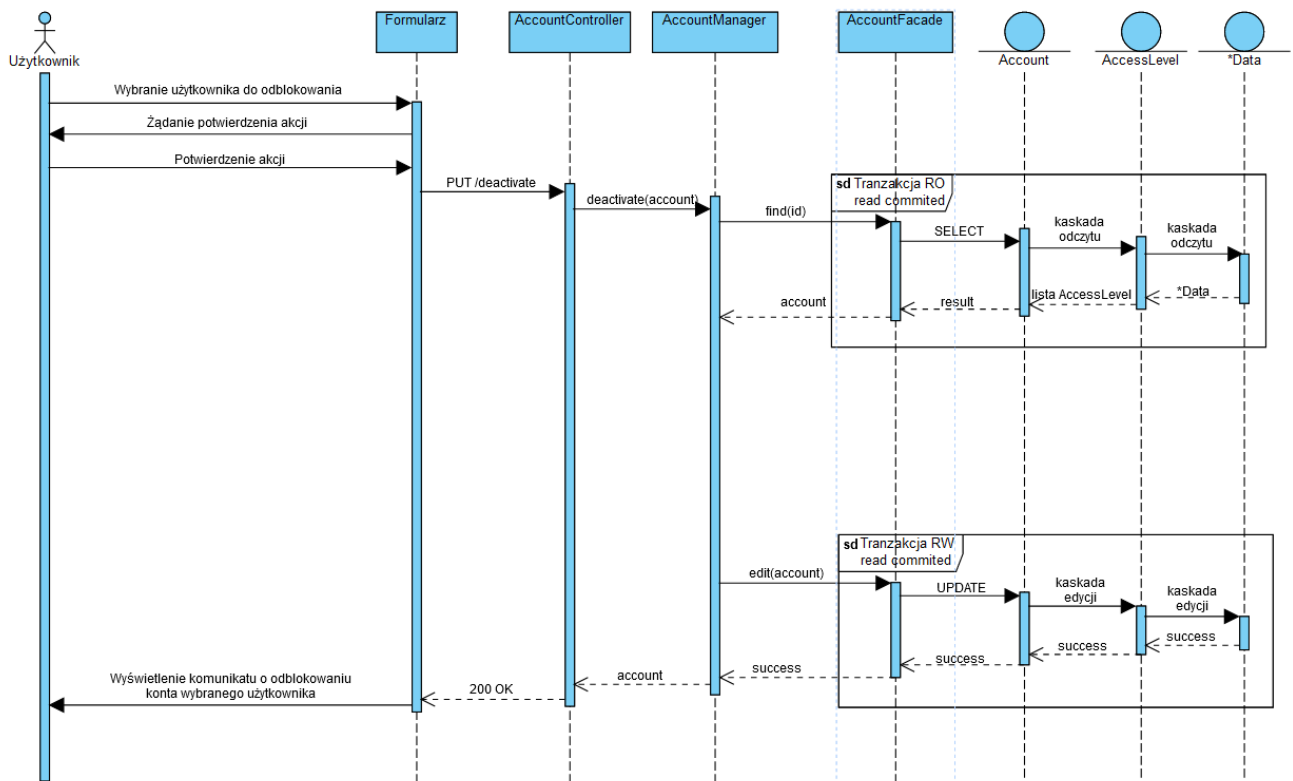
Formularz na diagramach to określenie na aplikację SPA.



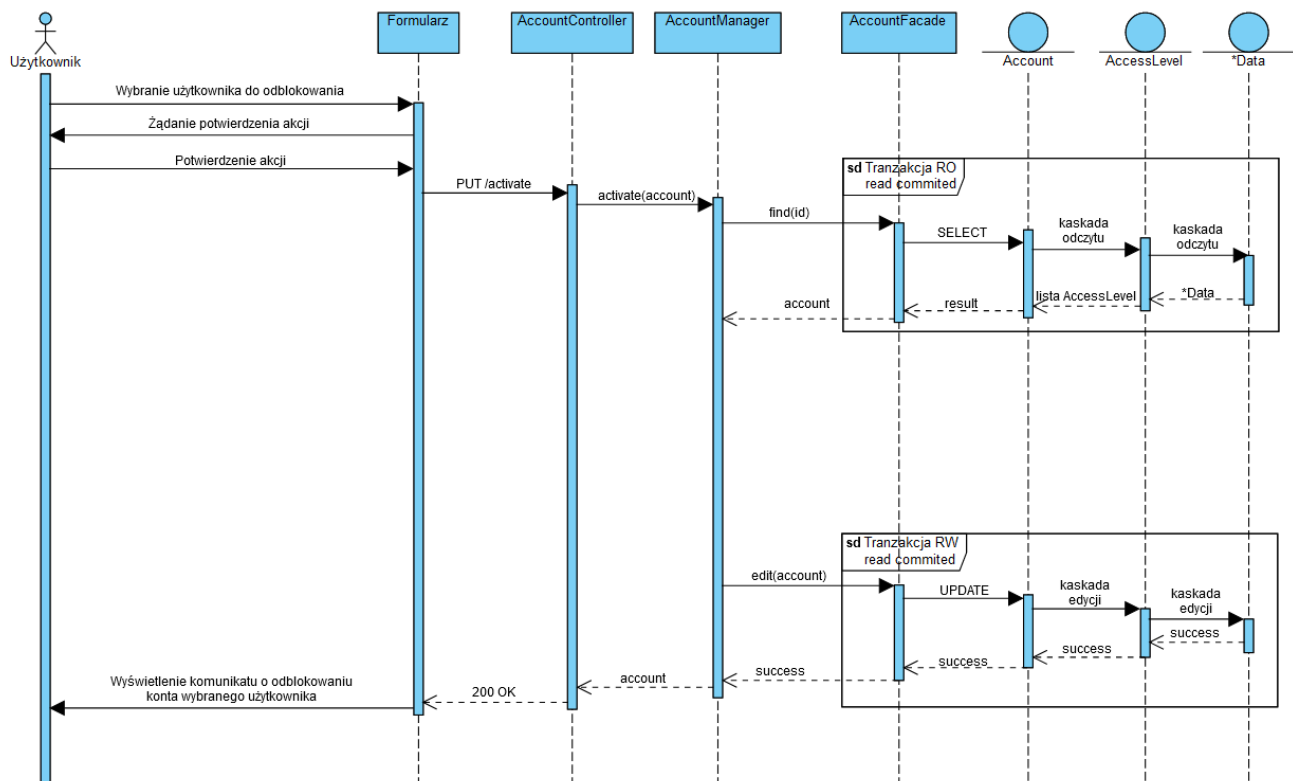
1.4.1 Diagram sekwencji dla przypadku użycia: MOK.1 Zarejestruj



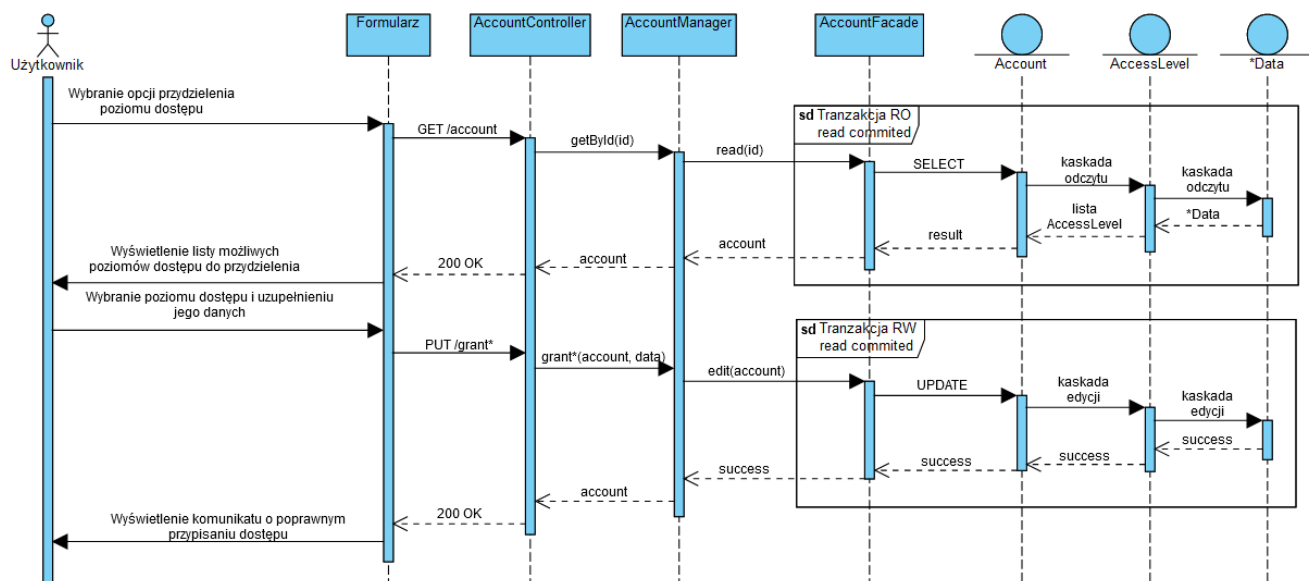
1.4.2 Diagram sekwencji dla przypadku użycia: MOK.2 Utwórz konto



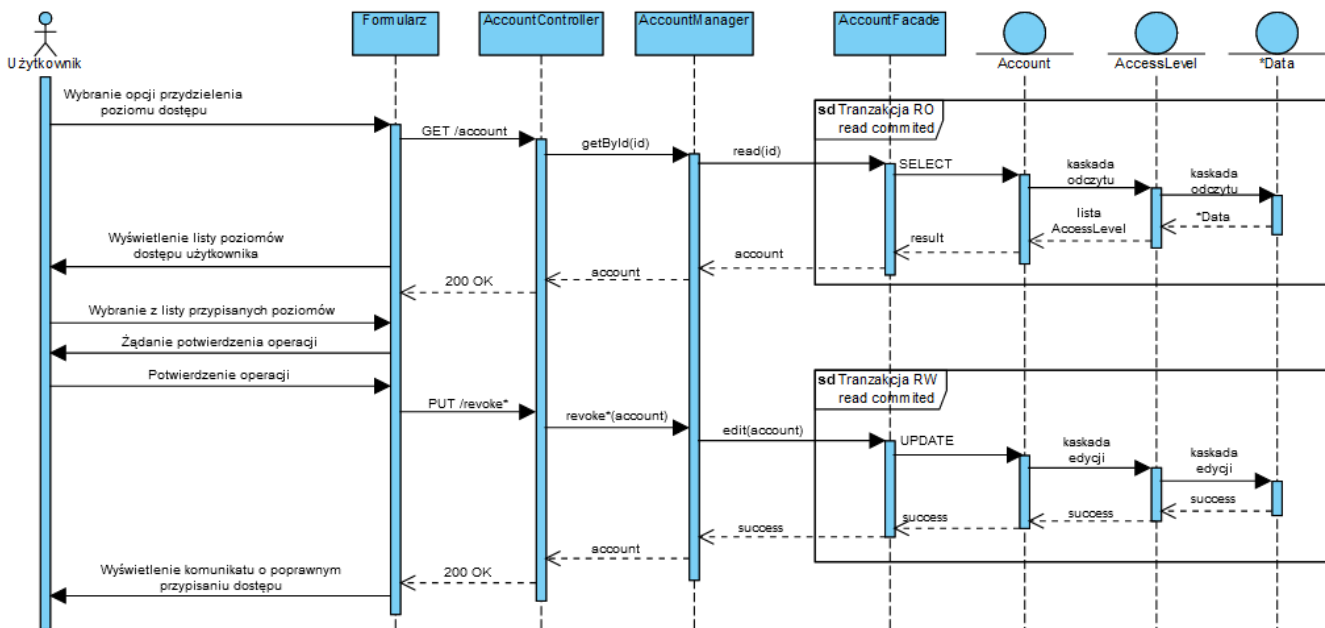
1.4.3 Diagram sekwencji dla przypadku użycia: MOK 3 Zablokuj konto



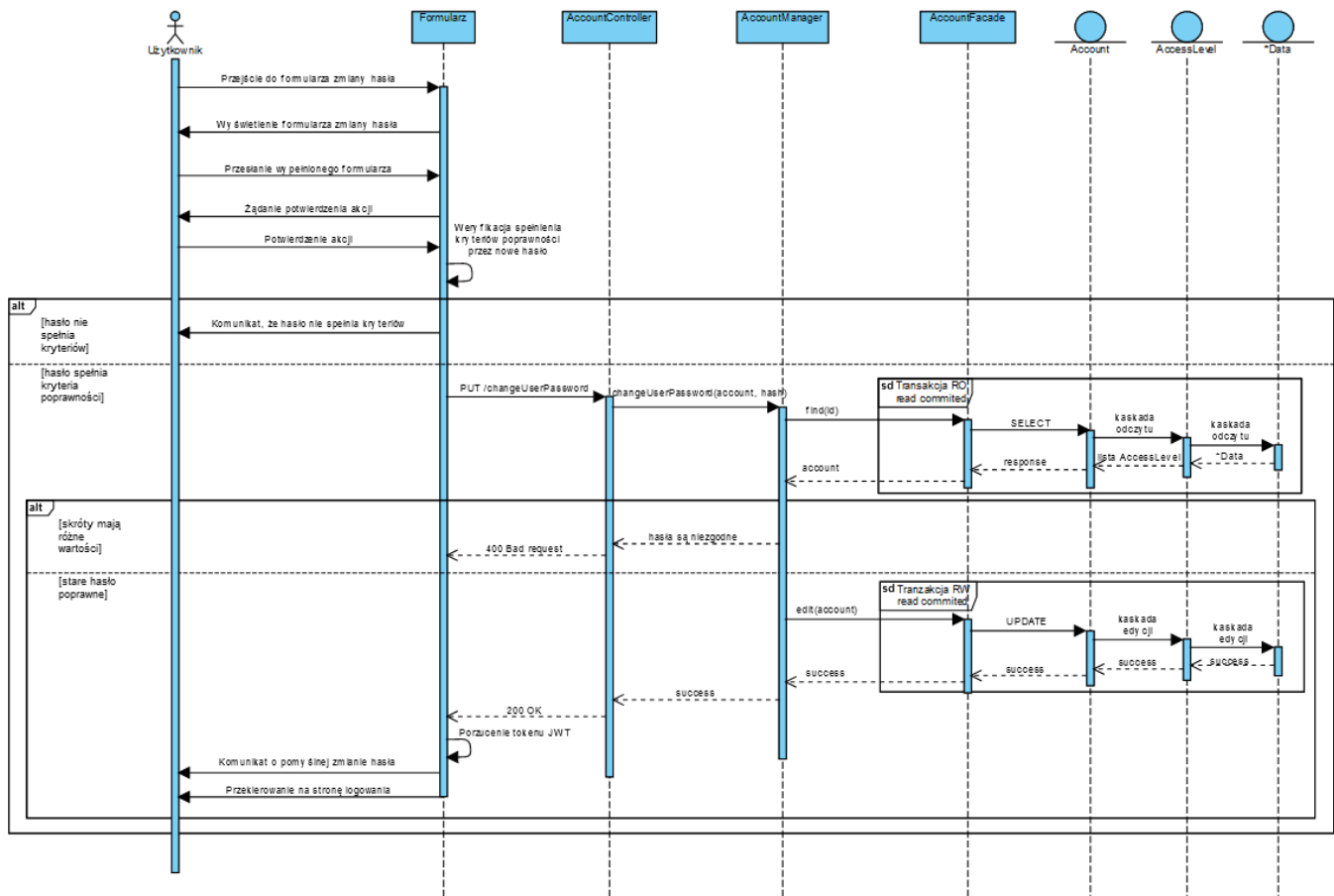
1.4.4 Diagram sekwencji dla przypadku użycia: MOK 4 Odblokuj konto



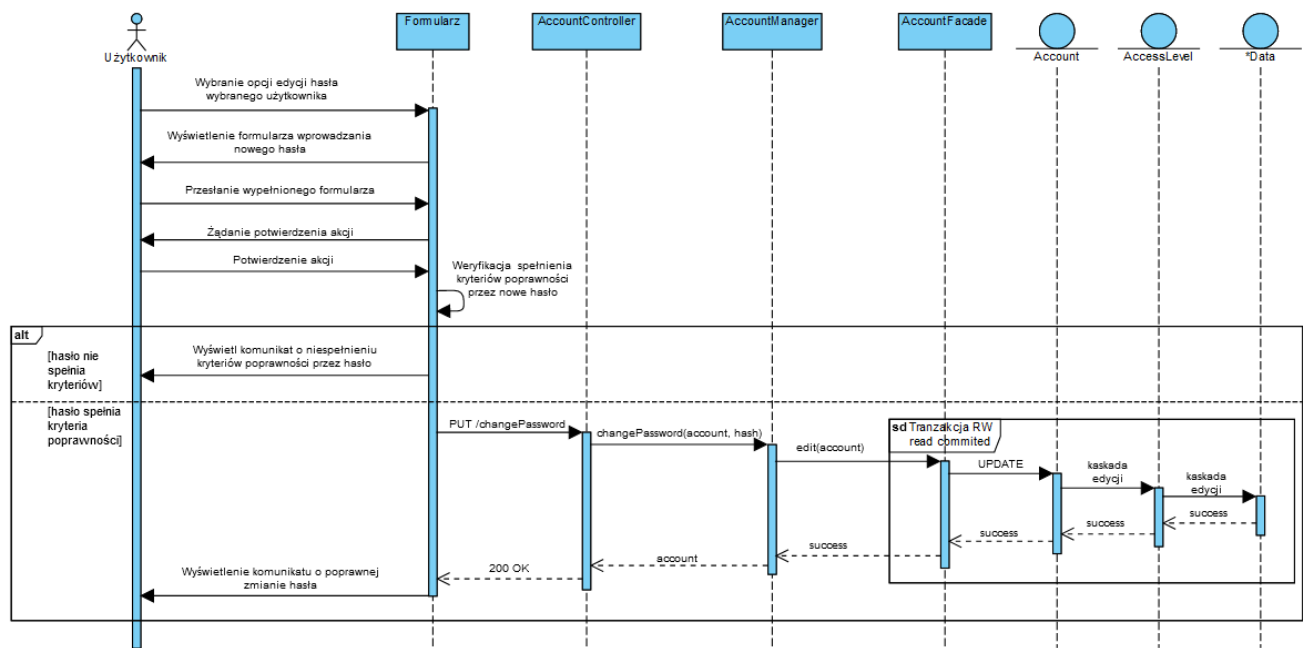
1.4.5 Diagram sekwencji dla przypadku użycia: MOK 5 Dołącz poziom dostępu do konta



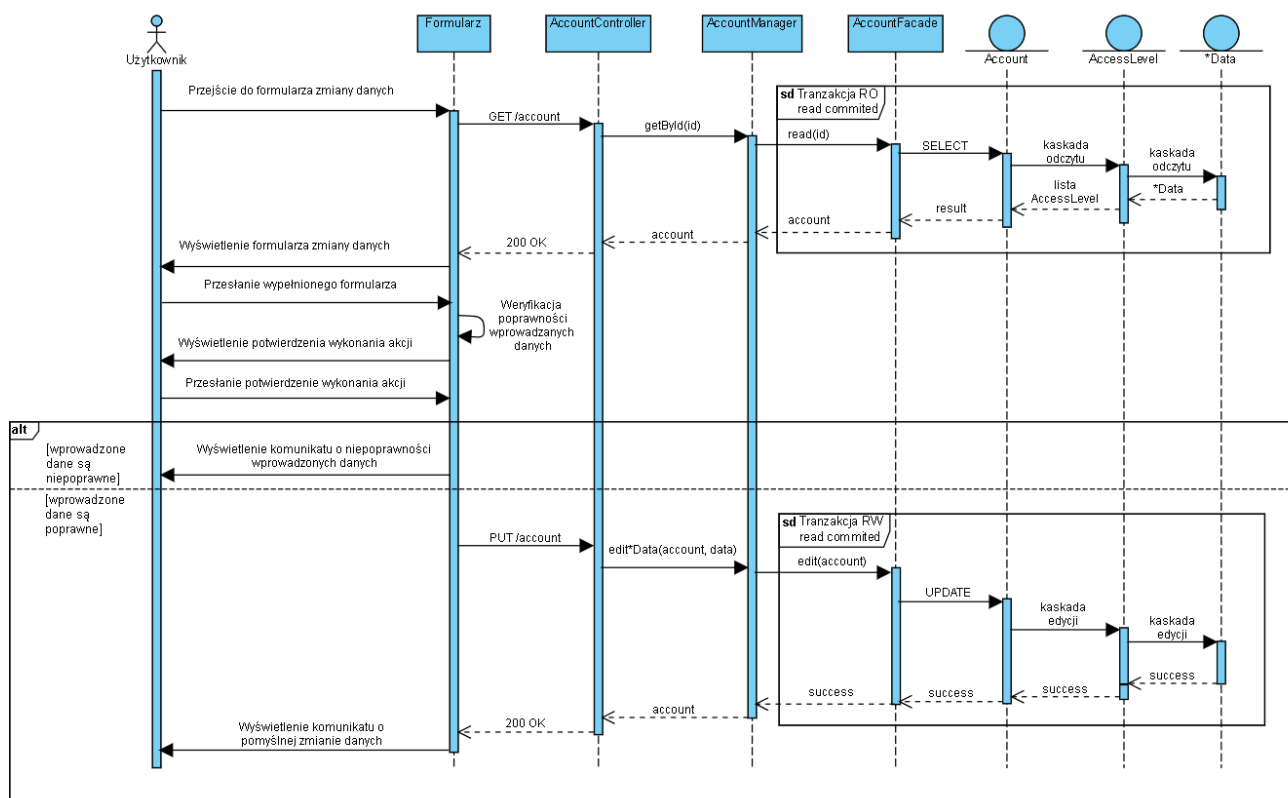
1.4.6 Diagram sekwencji dla przypadku użycia: MOK 6 Odłącz poziom dostępu od konta



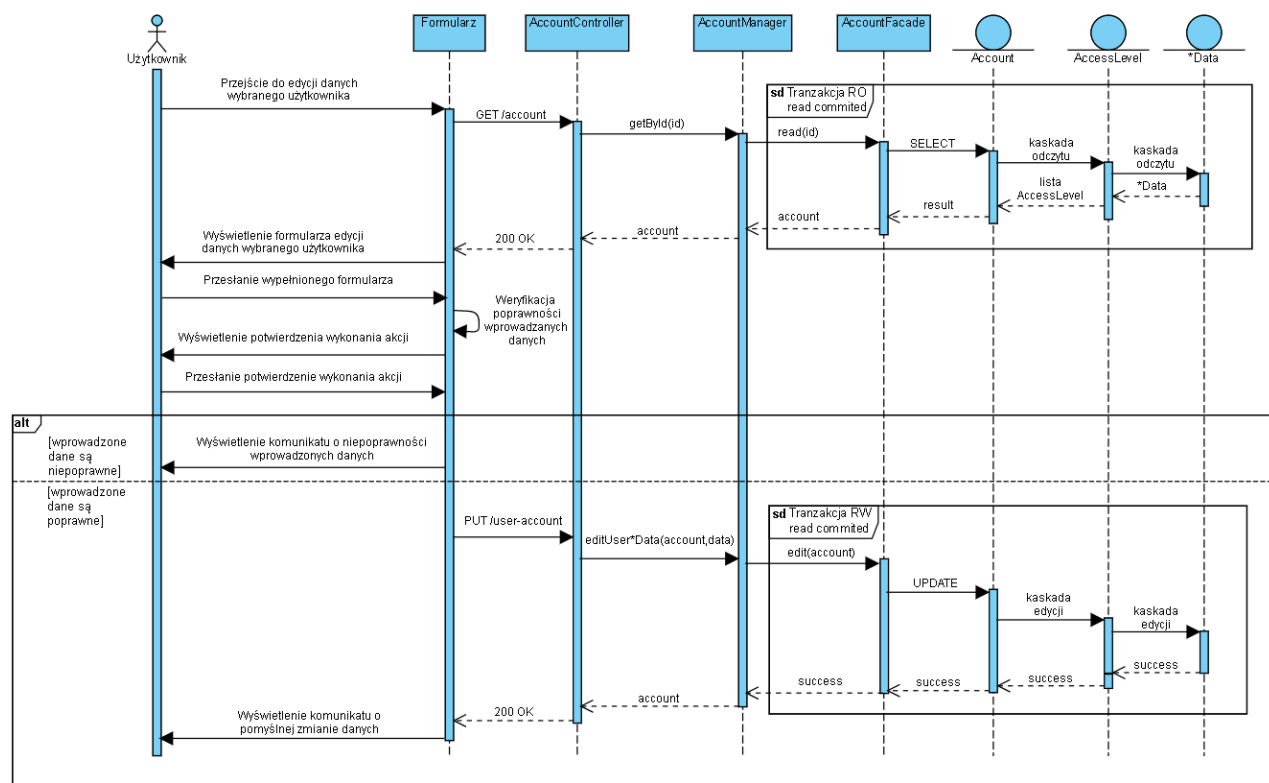
1.4.7 Diagram sekwencji dla przypadku użycia: MOK 7 Zmień własne hasło



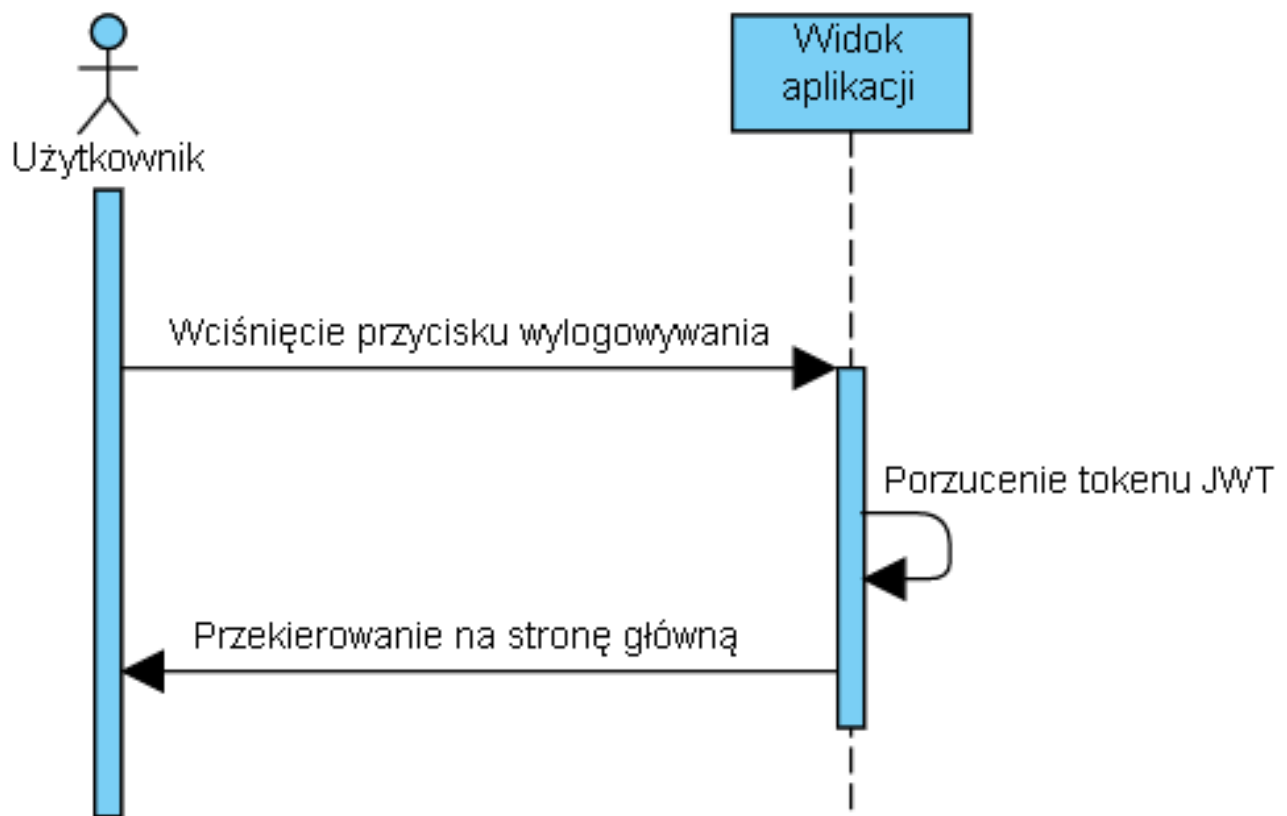
1.4.8 Diagram sekwencji dla przypadku użycia: MOK 8 Zmień hasło innego użytkownika



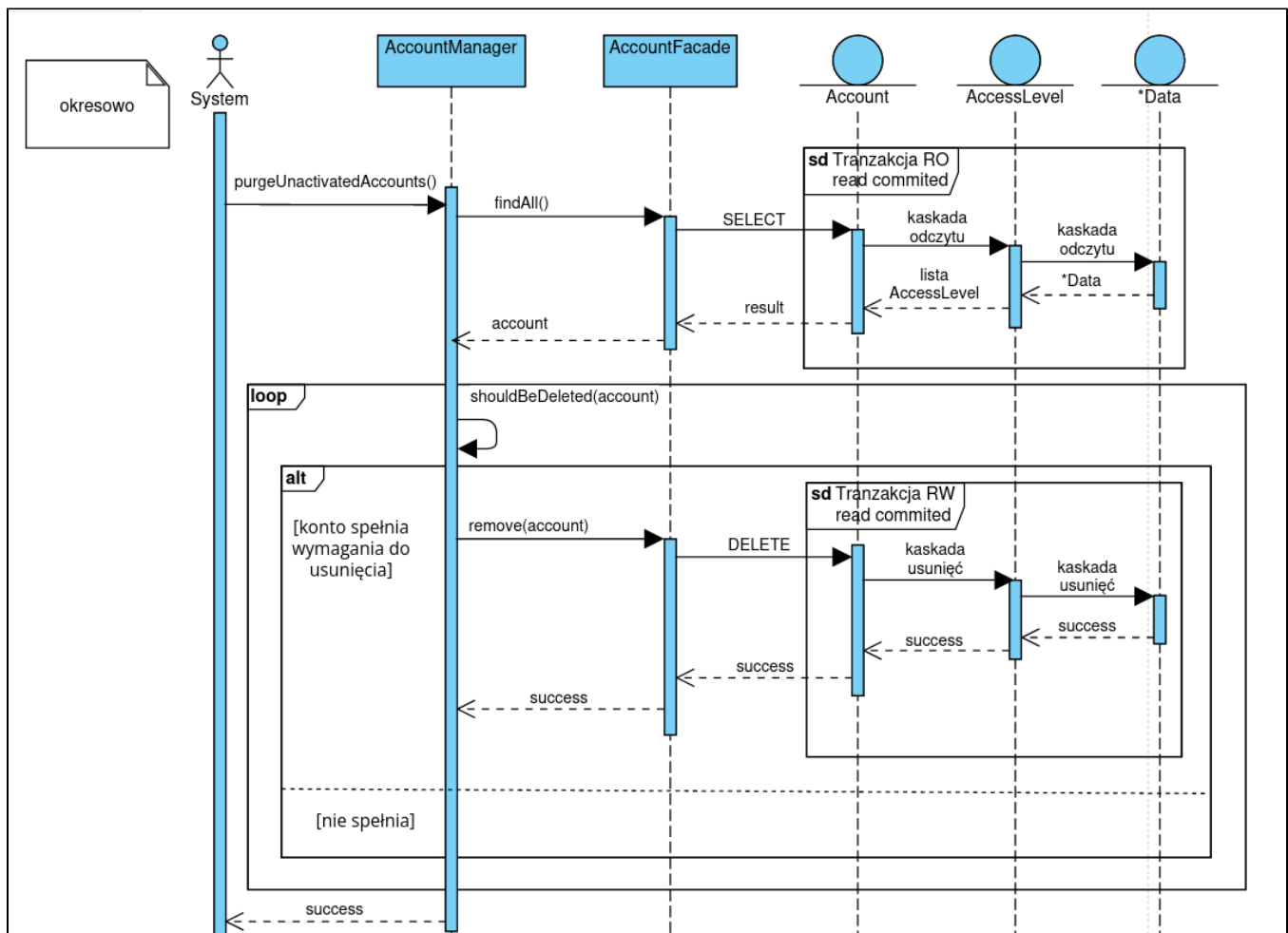
1.4.9 Diagram sekwencji dla przypadku użycia: MOK 9 Edytuj dane własnego konta



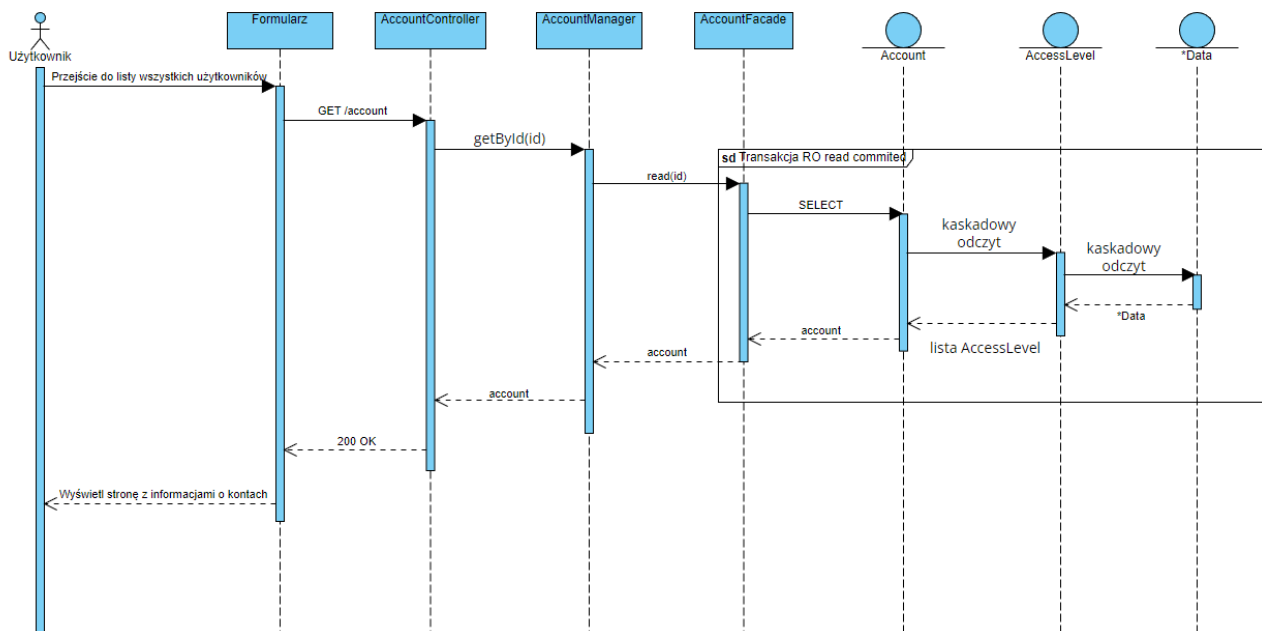
1.4.10 Diagram sekwencji dla przypadku użycia: MOK 10 Edytuj dane innego konta użytkownika



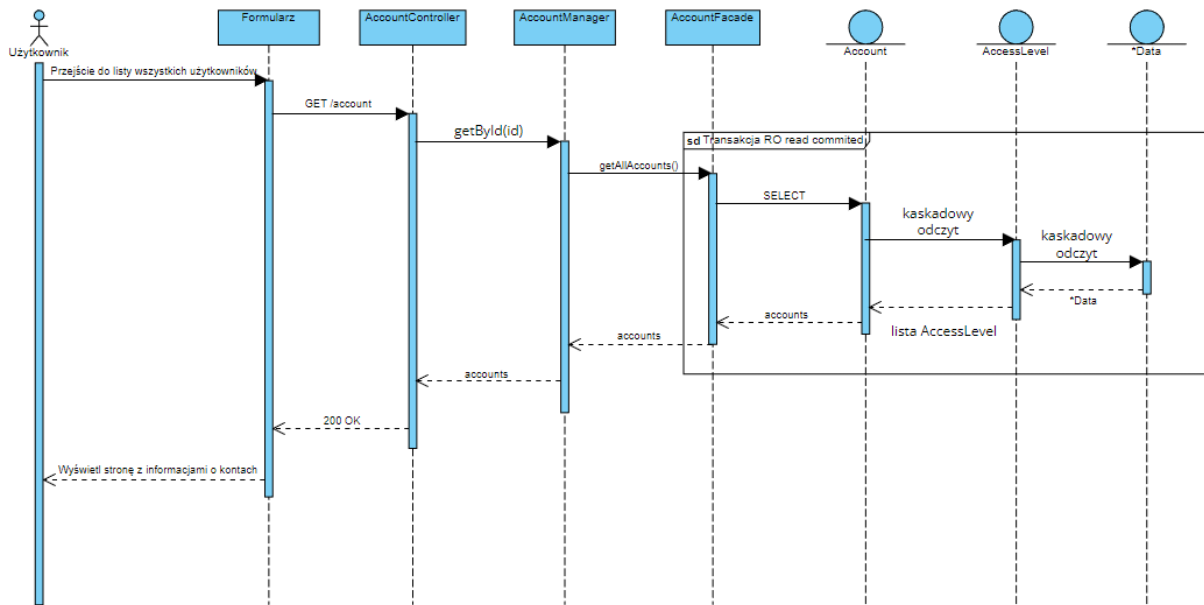
1.4.11 Diagram sekwencji dla przypadku użycia: MOK 11 Wyloguj



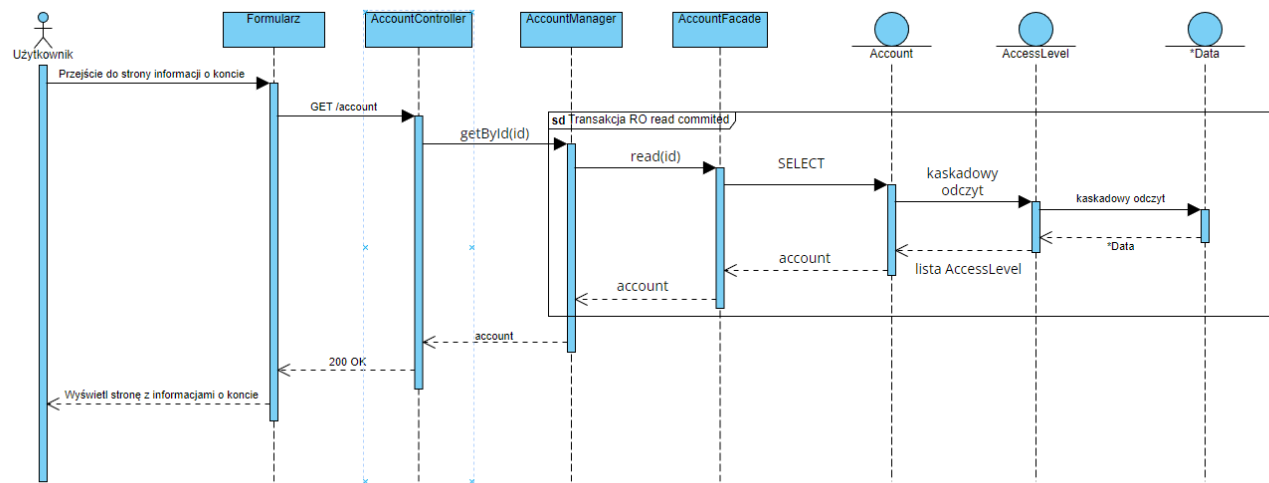
1.4.12 Diagram sekwencji dla przypadku użycia: MOK 12 Usui konto w przypadku braku aktywacji



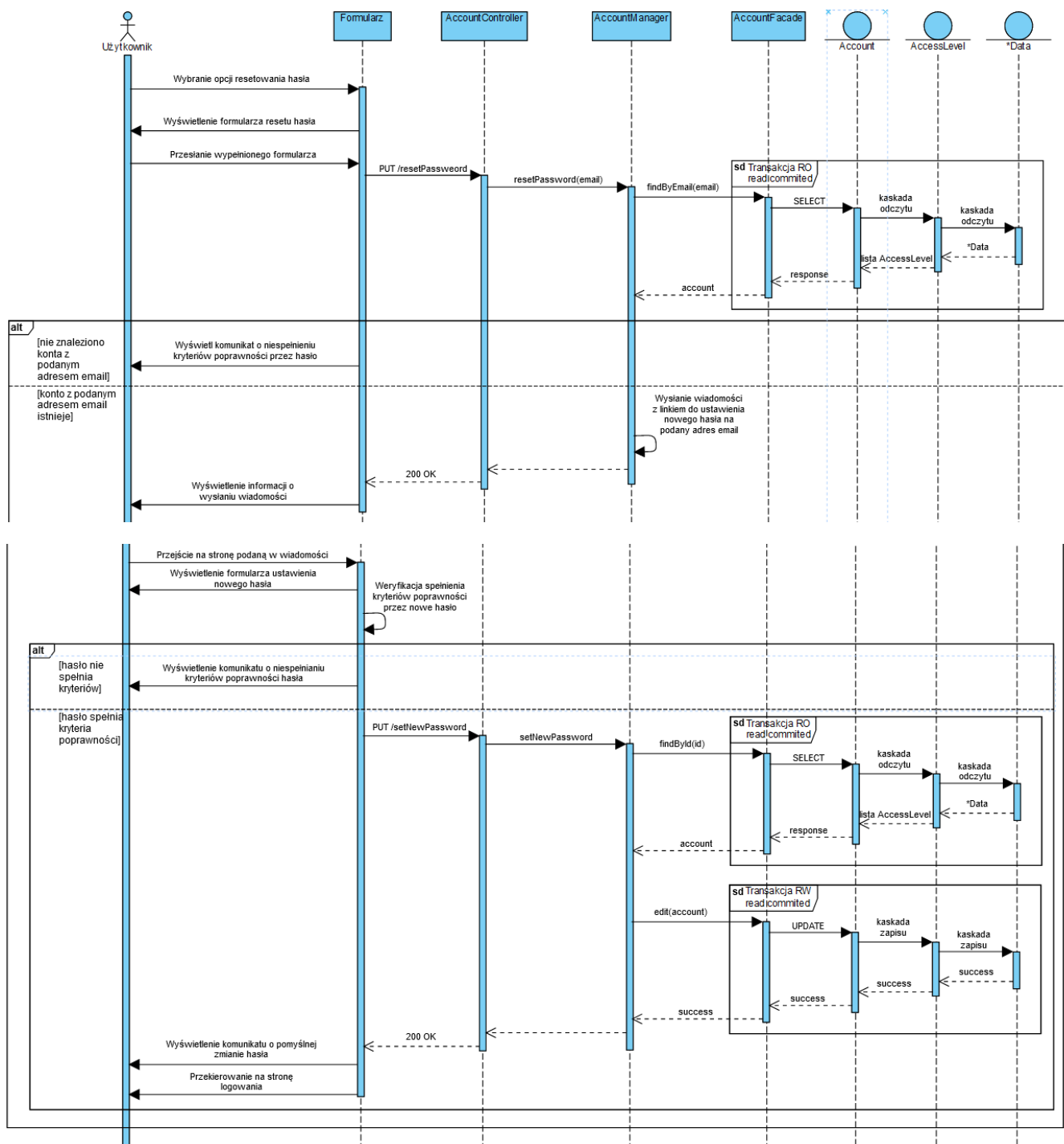
1.4.13 Diagram sekwencji dla przypadku użycia: MOK 13 Wyświetl informacje o koncie



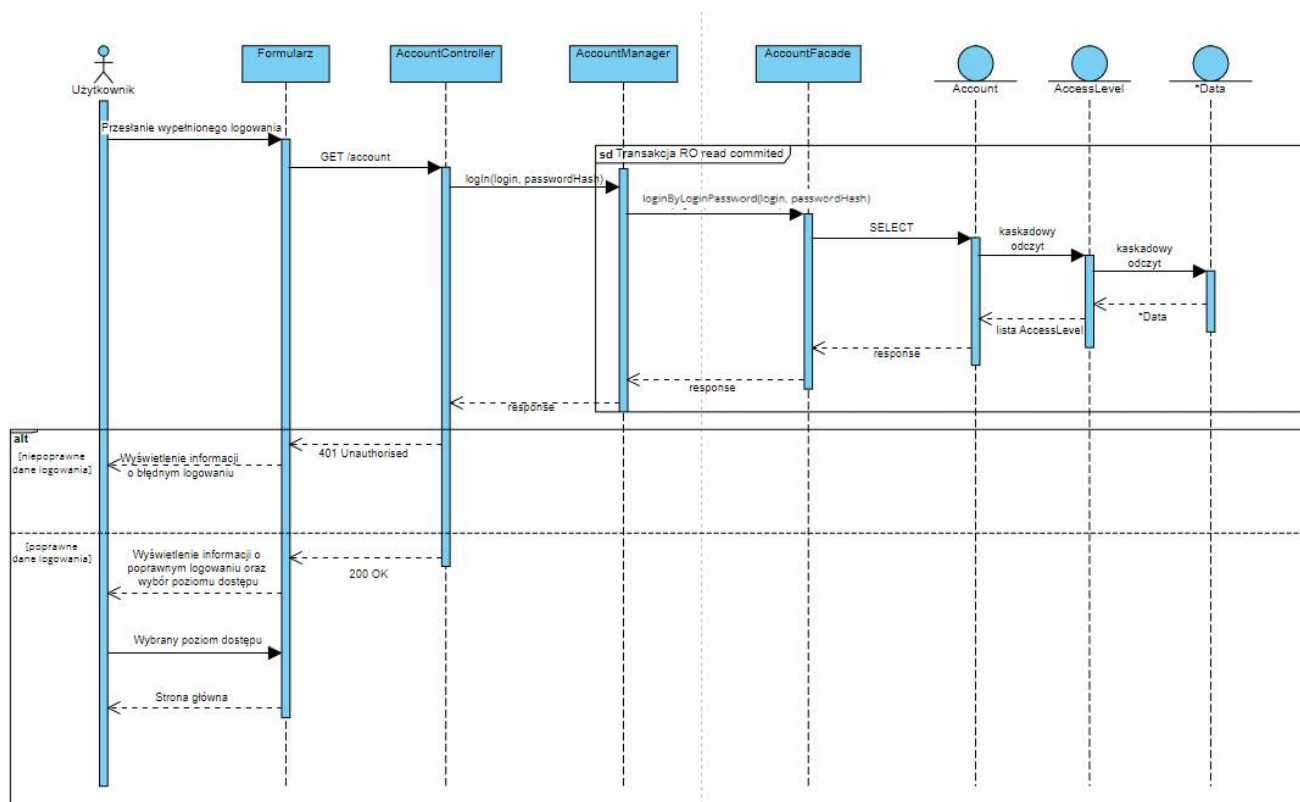
1.4.14 Diagram sekwencji dla przypadku użycia: MOK 14 Wyświetl listę kont



1.4.15 Diagram sekwencji dla przypadku użycia: MOK 15 Wyświetl informacje o koncie innego użytkownika



1.4.16 Diagram sekwencji dla przypadku użycia: MOK 16 Zresetuj hasło



1.4.17 Diagram sekwencji dla przypadku użycia: MOK 17 Zaloguj się

8 Konfiguracja uwierzytelniania w aplikacji 1 pkt

Należy zaprezentować wszystkie elementy niezbędne dla funkcjonowania mechanizmu uwierzytelniania użytkowników (implementacja i konfiguracja). Należy podać czas trwania sesji bezczynnego uwierzytelnionego użytkownika, po upływie którego nastąpi automatyczne zakończenie sesji użytkownika (wylogowanie). W przypadku zastosowania JWT należy przedstawić kiedy tworzony jest żeton, co zawiera i jaki ma okres ważności.

W bazie danych muszą znajdować się krotki umożliwiające przeprowadzenie procesu uwierzytelniania dla przykładowych kont użytkowników aplikacji, przy czym przynajmniej jedno z tych kont musi występować jednocześnie w wielu poziomach dostępu

Procedury zarządzania kontami

Należy zwięźle opisać planowane procedury zarządzania kontami:

- rejestracja kont - w przypadku stosowania dla różnych poziomów dostępu innych mechanizmów wymagany opis każdego z nich
- weryfikacji konta po rejestracji samodzielnej - w przypadku stosowania dla różnych poziomów dostępu innych mechanizmów wymagany opis każdego z nich.
- usuwania nieaktywnych kont po upływie limitu czasu na aktywację zarejestrowanego konta.
- blokowania i odblokowania konta
- przydzielania i odbierania poziomów dostępu

Przedstawione opisy muszą odnosić się do modyfikacji dokonywanych w krotkach bazy danych, precyzyjnie wyjaśniając jak wykonywane operacje wpływają na zmiany rekordów w poszczególnych tabelach relacyjnej bazy danych.

- rejestracja kont
 - Admin: konto jest tworzone przez istniejącego Admina poprzez podanie loginu i hasła.
 - Farmaceuta: konto jest tworzone z poziomu panelu administratora poprzez podanie loginu hasła i numeru licencji.
 - Pacjent: konto jest tworzone przez gościa poprzez podanie danych użytkownika, loginu, hasła i adresu e-mail.

Rejestracja konta skutkuje utworzeniem krotki w tabeli `account`, oraz powiązaną z nią encję odpowiedniego poziomu dostępu (krotka w tabeli `access_level` i jednej z: `patient_data`, `admin_data` i `chemist_data`. w przypadku kont tworzonych przez administratora wartość flagi `confirmed` jest od razu ustawiona na `true`, w przeciwnym razie jest to `false`

- weryfikacji konta po rejestracji samodzielnej

Na podany przez użytkownika adres e-mail wysyłany jest link aktywacyjny, skorzystanie z niego ustawia flagę `confirmed` na `true`. Można zalogować się jedynie na konto którego flaga `confirmed` jest ustawiona na `true`.

- usuwania nieaktywnych kont po upływie limitu czasu na aktywację zarejestrowanego konta.

Jeżeli pacjent nie skorzysta z linku w przeciągu 24 godzin jego konto wraz z przypisanymi do niego poziomami dostępu zostaną usunięte. (automatyczne usuwane są konta utworzone wcześniej niż 24 godziny temu (wartość pola `creationDate`) z flagą `confirmed` równą `false`)

- blokowania i odblokowania konta

Administrator może aktywować lub dezaktywować konta wszystkich użytkowników korzystając (zmieniana jest flaga `active` w tabeli `account`). Na konto z flagą `active` równą `false` nie można się zalogować.

- przydzielania i odbierania poziomów dostępu

Administrator może nadawać i odbierać poziomy dostępu kontom użytkowników (krotki w tabeli `access_level`).

Użytkownicy mogą aktywować lub dezaktywować przypisane do swojego konta poziomy dostępu (flaga `active` w tabeli `access_level`)