

# 1. Analiza wymagań

Poniżej przedstawiony zostanie opis świata rzeczywistego. Został on sporządzony w oparciu o wizytę w lecznicy weterynaryjnej, podczas której dowiedzieliśmy się co jest niezbędne do funkcjonowania systemu w takim przedsięwzięciu. Po scharakteryzowaniu środowiska zostaną przedstawieni aktorzy, którzy odzwierciedlają rzeczywiste role oraz funkcje jakie system powinien spełniać. Funkcje te zostaną przedstawione za pomocą diagramów przypadków użycia. Ze względu na większą przejrzystość diagram przypadków użycia został podzielony według odpowiadających im ról.

## 1.1.1. Opis środowiska

Celem projektu jest utworzenie systemu służącego do zarządzania lecznicą weterynaryjną, świadcząca usługi z zakresu opieki zdrowotnej nad zwierzętami. Tworzony system ma wspomagać pracowników lecznicy w ich pracy. Główne funkcje, które wspomagać ma system to zapisywanie klientów wraz z ich zwierzętami na wizyty do wybranych lekarzy oraz zarządzanie przebiegiem tych wizyt. Dodatkowe funkcje, które powinien spełniać system to zarządzanie sprzedażą leków oraz pomoc w prowadzeniu ewidencji magazynu. System zapewni również odpowiednie zaplanowanie pracy poprzez sporządzanie terminarzy dla pracowników.

W celu poprawnego zaprojektowania systemu powinniśmy również spojrzeć na działanie lecznicy z punktu widzenia klienta. Podstawowym krokiem do podjęcia współpracy klienta z kliniką jest kontakt między nimi. Klient może przyjść osobiście, skontaktować się telefonicznie lub umówić się poprzez formularz internetowy. Pośrednikiem w świadczeniu usług jest zawsze pracownik recepcji. Następnie w zależności od potrzeb klienta może on umówić na wizytę, uregulować zaległe płatności lub sprzedać odpowiednie produkty. W przypadku umawiania wizyty pracownik recepcji ustala termin a następnie klient stawia się w wyznaczonym czasie. Po przeprowadzonej wizycie klient ponownie skierowany zostaje do recepcji w celu uregulowania należności lub odebrania odpowiednich lekarstw.

Możliwość zapewnienia wyżej wymienionych funkcjonalności jest związana z potrzebą przechowywania danych o klientach oraz ich zwierzętach. Niezbędna jest również poprawna organizacja pracy magazynu, która może mieć kluczowe znaczenie w sprawnym funkcjonowaniu kliniki.

## **2. Wymaganie funkcjonalne i нефункционаłne systemu**

### **2.1. Wymagania funkcjonalne**

1. Przyjmowanie klient6w
  - 1.1. Przyjęcie klienta na podstawie um6wionej wizyty
  - 1.2. Wizyta u klienta na podstawie wyznaczonego terminu.
2. Zapisywanie na wizyty
  - 2.1. Zgłoszenie przez formularz internetowy
  - 2.2. Osobiste um6wienie się na wizytę (telefonicznie lub osobiście)
3. Świadczenie usług medycznych
4. Sprzedaż leków
  - 4.1. Sprzedaż leków wypisanych przez lekarza.
  - 4.2. Sprzedaż leków podanych podczas wizyty.
  - 4.3. Niezależna sprzedaż leków.
5. Prowadzenie karty klienta
  - 5.1. Założenie karty klienta
  - 5.2. Rejestrowanie zmian w karcie klienta
    - 5.2.1. Przypisanie zwierzęcia
    - 5.2.2. Edycja danych osobowych lub kontaktowych
  - 5.3. Powiadomienia
    - 5.3.1. Powiadomienia o usługach okresowych
    - 5.3.2. Powiadomienia o wizytach
    - 5.3.3. Rozliczenia miesięczne
    - 5.3.4. Przypomnienia o konieczności zamówienia leku
6. Prowadzenie karty zwierzęcia
  - 6.1. Założenie karty
  - 6.2. Rejestrowanie zmian
    - 6.2.1. Monitorowanie zmian w historii leczenia
    - 6.2.2. Dodawanie predefiniowanych zabieg6w
    - 6.2.3. Zmiana właściciela
  - 6.3. Dezaktywacja karty zwierzęcia
7. Zarządzanie magazynem
  - 7.1. Przyjmowanie dostaw
  - 7.2. Ewidencja przechowywanych pozycji magazynowych

### 7.3. Zarządzanie wydawaniem towarów

- 8. Zarządzanie kliniką
  - 8.1. Składanie zamówień
  - 8.2. Zarządzanie kadrami
  - 8.3. Organizacja pracy
  - 8.4. Zarządzanie usługami
- 9. Logowanie do systemu
  - 9.1. Podział na role w systemie
  - 9.2. Definicja uprawnień
  - 9.3. Zarządzanie uprawnieniami

## 2.2 Wymagania нефunkcjonalne

1. Aplikacja powinna zostać przygotowana w wersji graficznej
2. Baza danych powinna być archiwizowana
3. Czas odtwarzania systemu z kopii zapasowej nie powinien przekraczać 15 minut

## 3. Specyfikacja wymagań funkcjonalnych za pomocą diagramów

### 3.1 Aktorzy

**Lekarz** – osoba pracująca w lecznicy. Głównym zadaniem lekarza jest wykonywanie usług na rzecz zwierząt podczas wcześniej ustalonych z klientami wizyt, prowadzenie kart zwierząt oraz sprzedaż leków na receptę.

**Pracownik recepcji** – osoba pracująca w lecznicy. Główne zadanie pracownika recepcji to zapisywanie nowych wizyt zgodnych z grafikiem lekarza, tworzenie nowych kart klientów, przyjmowanie płatności od klientów oraz sprzedaż leków bez recepty.

**Kierownik** – najważniejsza osoba w lecznicy. Zarządza pracownikami oraz godzinami ich pracy

**Klient** – osoba korzystająca z usług lecznicy na rzecz swoich zwierząt.

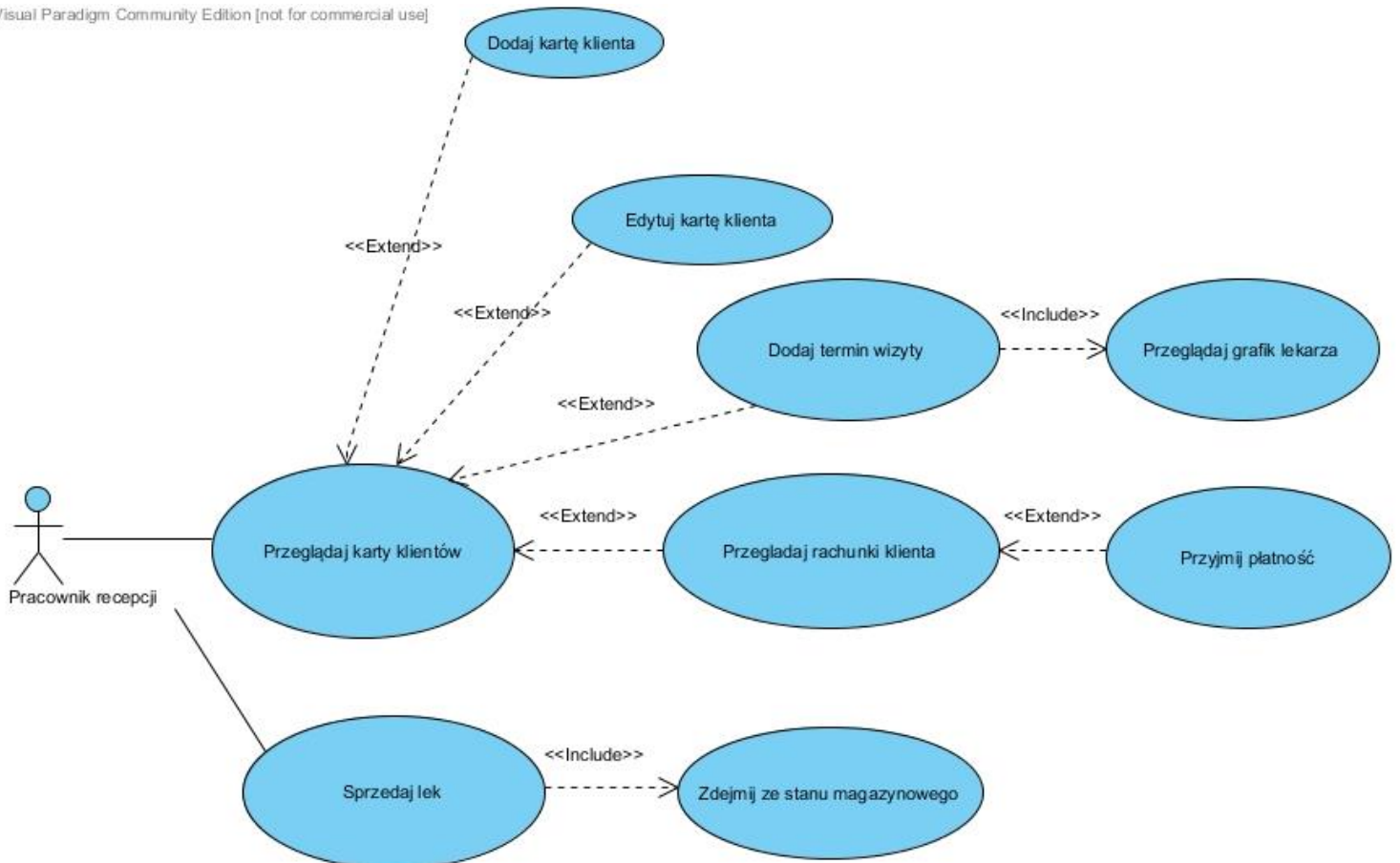
**Magazynier** – osoba zajmująca się przyjmowaniem towaru i zarządzaniem magazynem.

## 3.2 Diagramy przypadków użycia

Ze względu na większą przejrzystość diagram przypadków użycia został podzielony według odpowiadających im ról.

### 3.2.1 Diagram przypadków użycia dla pracownika recepcji

Visual Paradigm Community Edition [not for commercial use]



- Przeglądaj karty klientów** – możliwość przeglądanie kart klientów zapisanych do lecznicy.
- Dodaj kartę klienta** – możliwość dodania karty klienta wraz z jego danymi osobowymi.
- Edytuj kartę klienta** – możliwość edycji karty klienta i jego danych osobowych.
- Dodaj termin wizyty** – przypisanie nowej wizyty do karty klienta i wybranego lekarza.
- Przeglądaj grafik lekarza** – przeglądanie możliwych dyżurów i wolnych terminów lekarza.
- Przeglądaj rachunki klienta** – możliwość wyświetlanie wszystkich rachunków za wykonane usługi i sprzedane leki.
- Przyjmij płatność** – przyjęcie pieniędzy za wykonane usługi i zmiana stanu

rachunku na „uregulowany”.

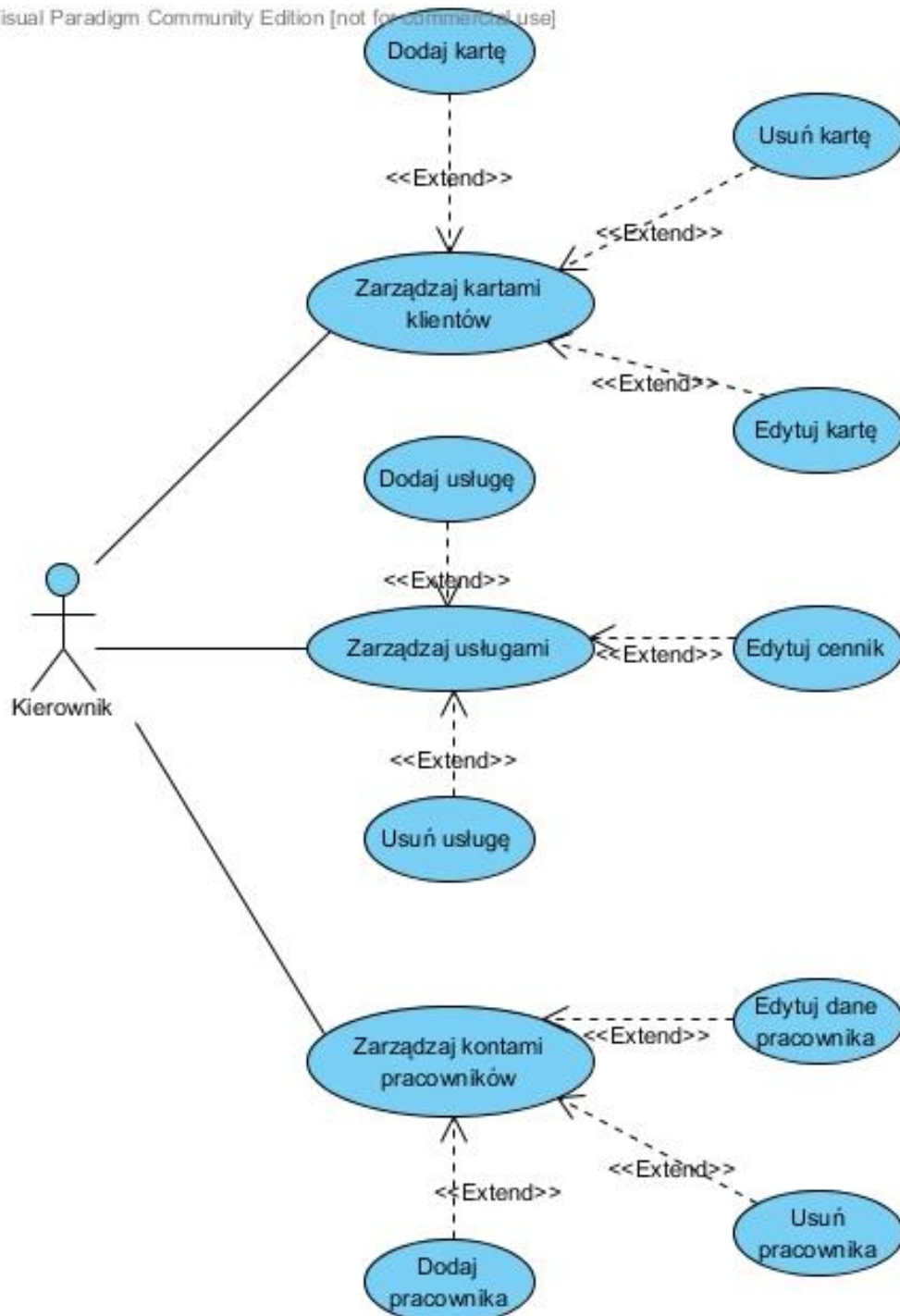
**Sprzedaj lek**

- sprzedaż leku znajdującego się w magazynie. Umożliwia sprzedaż na rachunek i bez rachunku.

**Zdejmij ze stanu magazynowego** – zmniejszenie ilości leków w magazynie.

**3.2.2 Diagram przypadków użycia dla kierownika lecznicy**

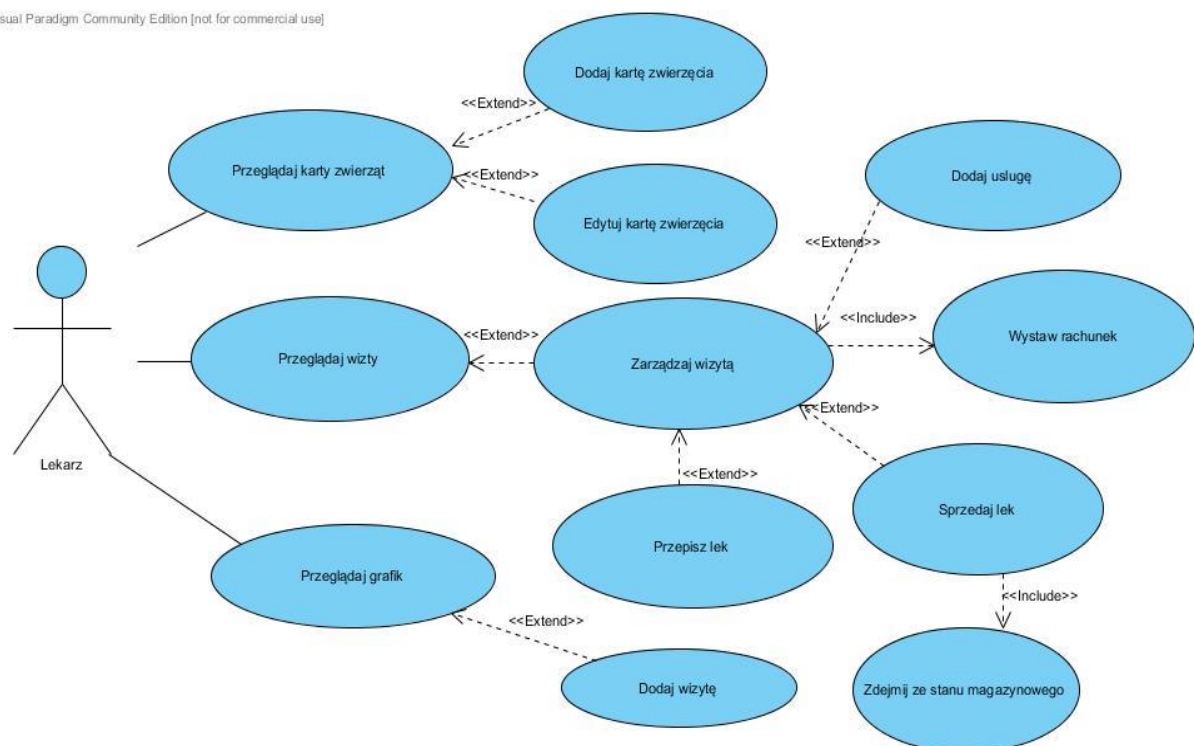
Visual Paradigm Community Edition [not for commercial use]



- Zarządzaj usługami** – przypadek użycia umożliwiający obejrzenie listy usług.
- Dodaj usługę** – możliwość zdefiniowania nowej usługi oferowanej przez lecznicę oraz określenie jej ceny.
- Edytuj cennik** - możliwość zmiany ceny za wybrane usługi
- Usuń usługę** - usunięcie usługi z listy usług oferowanych przez lecznicę.
- Zarządzaj kontami pracowników** – przypadek użycia umożliwiający przeglądanie kont pracowników.
- Dodaj pracownika** – możliwość dodania nowego pracownika do lecznicy wraz z jego danymi osobowymi i terminami dyżurów.
- Edytuj pracownika** – edycja danych osobowych pracownika oraz terminu dyżurów.
- Usuń pracownika** – usunięcia pracownika z listy pracowników lecznicy

### 3.2.3 Diagram przypadków użycia dla lekarza

Visual Paradigm Community Edition [not for commercial use]



**Przeglądaj karty zwierząt** – przypadek użycia umożliwiający wyświetlenie i oglądanie wszystkich kart zwierząt leczonych w klinice.

**Dodaj kartę zwierzęcia** – przypadek użycia umożliwiający dodanie karty nowego zwierzęcia oraz jego danych.

**Edytuj kartę zwierzęcia** – przypadek użycia umożliwiający edycję karty zwierzęcia.

**Przeglądaj wizyty** – możliwość przeglądanie wszystkich wizyt , zarówno tych odbytych oraz nadchodzących.

**Zarządzaj wizytą** – przypadek użycia wyświetlający konkretnie wybraną wizytę i umożliwiający przeprowadzenie czynności związanych z wizytą.

**Dodaj usługę** – przypisanie usługi do wizyt wraz z jej ceną

**Przepisz lek** – przypisanie leku dla zwierzęcia

**Sprzedaj lek** – sprzedanie leku i dopisanie go do rachunku klienta

**Zdejmij ze stanu magazynowego** – zmniejszenie ilości leków w magazynie.

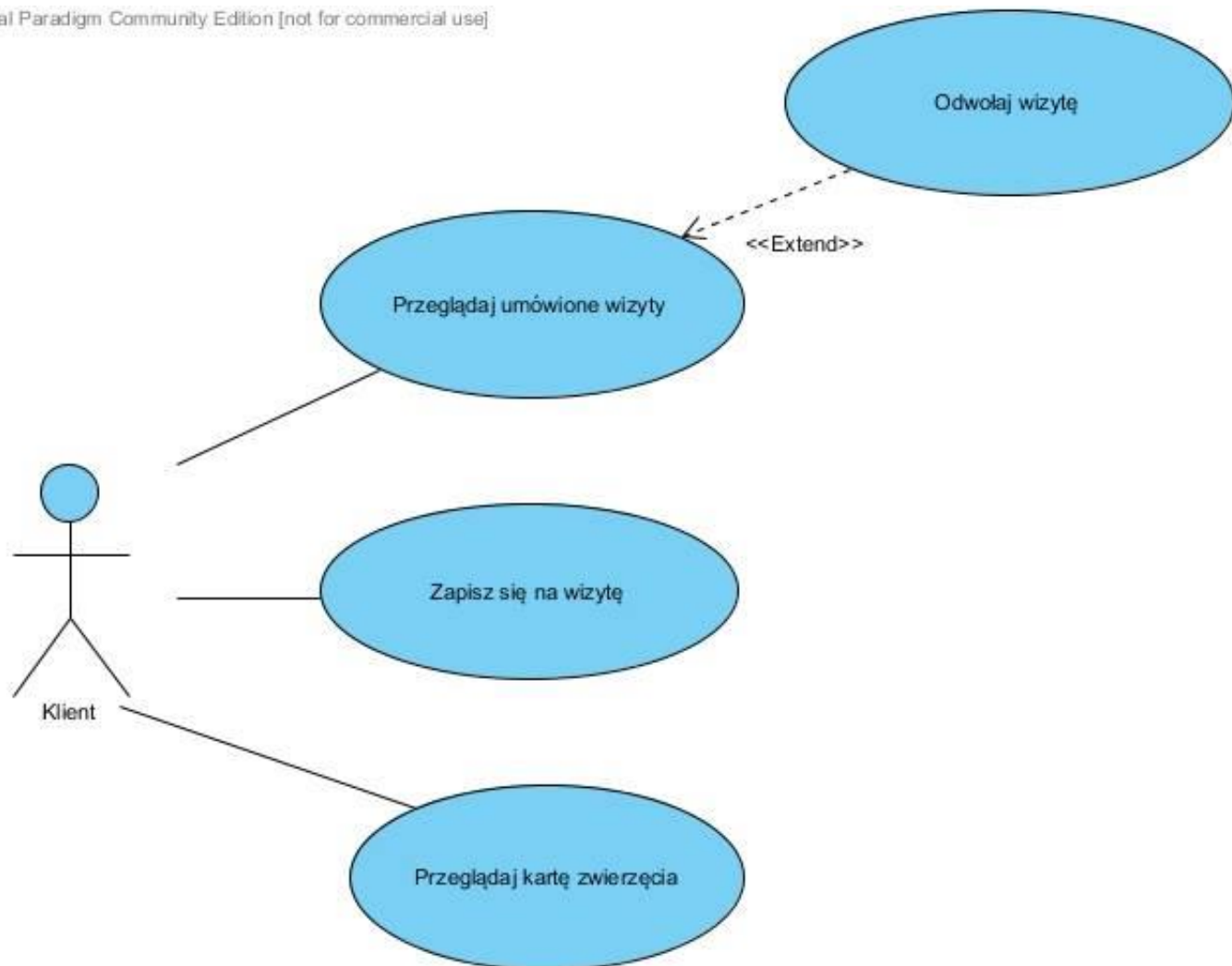
**Wystaw rachunek** – przypisanie rachunku za daną wizytę do klienta wraz z łączną kwotą.

**Przeglądaj grafik** – przeglądaj wolne i zajęte terminy oraz terminy dyżurów.

**Dodaj wizytę** – wyznaczenie terminu nowej wizyty .

### 3.2.4 Diagram przypadków użycia dla klienta

Visual Paradigm Community Edition [not for commercial use]



**Przeglądaj umówione wizyty** – przeglądanie wszystkich wizyt, na które zapisany jest klient

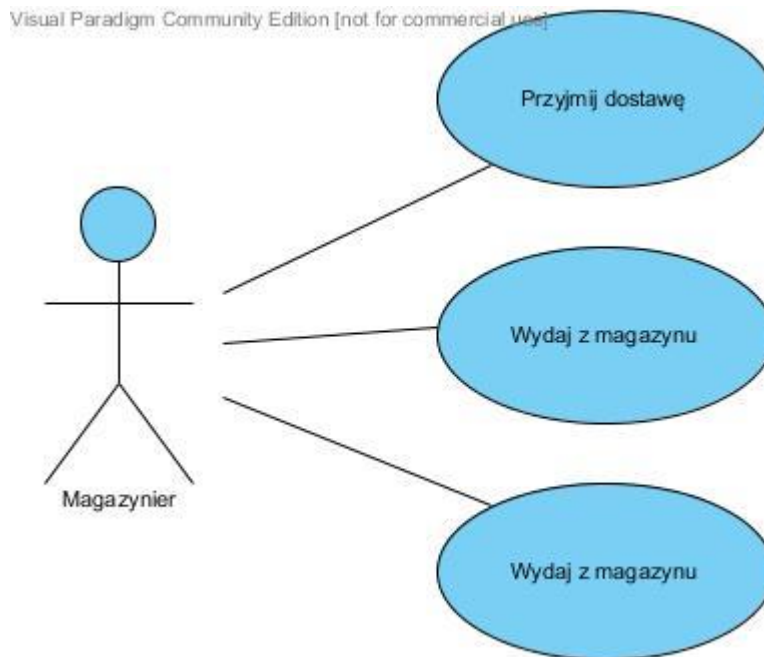
**Odwołaj wizytę** – odwołanie wcześniej umówionej wizyty

**Zapisz się na wizytę** – zgłoszenie chęci umówienia się na wizytę

**Przeglądaj kartę zwierzęcia** – przeglądanie karty zwierzęcia wraz z przepisanyimi lekami i wykonanymi usługami.



### 3.2.5 Diagram przypadków użycia dla magazyniera

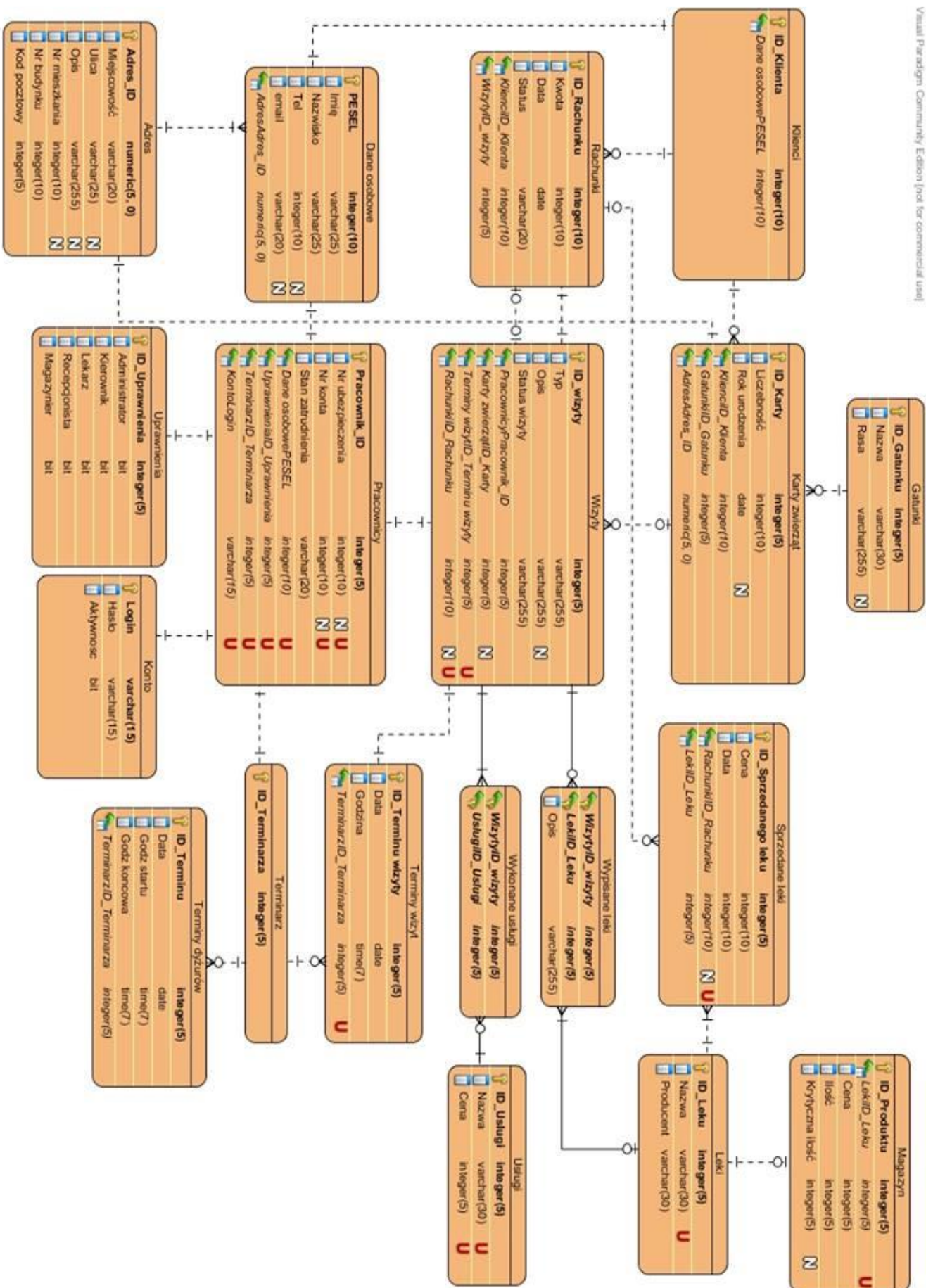


**Przyjmij dostawę** – dodanie nowych leków na stan magazynowy.

**Wydaj towar** – zdejmij ze stanu magazynowego

**Oznacz jako stratę** – przypadek użycia umożliwiający usunięcie towaru z magazynu bez jego sprzedaży i dodanie go do strat.

## 4. Diagram ERD



## Etap II : Projekt bazy danych

### 5. Projekt bazy danych

W tym rozdziale przedstawiony zostanie projekt bazy danych wraz odpowiednimi analizami umożliwiającymi jej sprawne działanie.

#### 5.1 Analiza liczby instancji dla każdej encji

Nazwa encji	Spodziewana l. instancji po roku użytkowania systemu	Opis
Adres	240	Encja zawierająca dane adresowe klientów, pracowników oraz zwierząt. Maksymalna liczba instancji jest sumą liczb instancji encji „Dane osobowe” oraz „Karty zwierząt”.
Dane osobowe	207	Encja zawierająca dane osobowe klientów oraz pracowników. Liczba instancji jest sumą liczb instancji encji „Pracownicy” oraz „Klienci”.
Gatunki	40	Encja zawierająca informacje na temat gatunków zwierząt przyjmowanych w lecznicy. Liczba instancji jest uzależniona od różnorodności przyjmowanych zwierząt. Maksymalnie może być równa liczbie instancji encji „Karty zwierząt”. W praktyce przypadek taki jest jednak bardzo mało prawdopodobny. Liczba instancji tej encji została określona na podstawie analizy dotychczasowej działalności lecznicy.
Karty zwierząt	250	Encja zawierająca informacje na temat zwierzęcia lub stada. Liczba instancji jest co najmniej równa liczbie instancji encji „Klient”. Z danych lecznicy wynika, że na jednego klienta przypada średnio 1,25 zwierzęcia.
Klienci	200	Encja zawierająca dane osobowe klienta oraz jego identyfikator przypisywany do rachunków oraz kart zwierząt. Liczba instancji wynika z ruchu w lecznicy.
Konta	7	Encja zawiera dane do logowania dla pracowników. Liczba instancji odpowiada liczbie pracowników
Leki	300	Encja zawierająca informacje na temat leków znajdujących się na rynku. Liczba instancji jest liczbą leków wprowadzonych do systemu.
Magazyn	240	Encja zawierająca informacje o lekach znajdujących się w magazynie. Liczba instancji jest równa liczbie typów leków dostępnych w lecznicy.
Pracownicy	7	Encja zawierająca dane osobowe pracownika, jego uprawnienia oraz terminy. Liczba instancji jest liczbą zatrudnionych pracowników.
Rachunki	1700	Encja zawierająca koszt wizyty lub sprzedanych leków. Liczba instancji encji w najgorszym przypadku jest równa sumie liczb instancji encji „Wizyty” oraz „Sprzedane leki”.

Nazwa encji	Spodziewana l. instancji po roku użytkowania systemu	Opis
Sprzedane leki	2000	Encja zawierająca informacje na temat sprzedanych leków, łączy ona tabele „Rachunki” oraz „Leki”. Liczbę instancji oszacowano na podstawie dotychczasowej sprzedaży w lecznicy.
Terminarz	7	Encja łącząca tabele „Terminy wizyt” oraz „Terminy dyżurów”. Przypisywana jest ona do pracownika. Liczba instancji jest równa liczbie pracowników.
Terminy dyżurów	700	Encja zawierająca informacje na temat dyżurów poszczególnego pracownika w danym dniu. Terminy dyżurów mogą być dodawane z dwumiesięcznym wyprzedzeniem oraz przechowywane są przez okres dwóch miesięcy. Liczba instancji w przybliżeniu wynosić będzie 100 na każdego pracownika.
Terminy wizyt	1200	Encja zawierająca terminy umówionych oraz odbytych wizyt. Instancja zawiera odwołanie do terminarza lekarza oraz przypisywana jest do konkretnej wizyty. Liczba instancji jest równa liczbie instancji encji „Wizyty”.
Uprawnienia	7	Encja zawierająca informacje na temat uprawnień konkretnego pracownika. Liczba instancji równa jest liczbie pracowników.
Usługi	20	Encja zawierająca informacje na temat oferowanych przez lecznicę usług. Liczba instancji wynika z dotychczasowej oferty.
Wizyty	1200	Encja zawierająca informacje na temat umówionych oraz odbytych wizyt. Instancja zawiera informacje o typie, statusie oraz terminie. Zawiera również odwołanie do lekarza, karty zwierzęcia oraz rachunku. Liczba instancji uzależniona jest od ilości świadczonych wizyt przez lecznicę.
Wykonane usługi	1800	Encja łączy tabele „Wizyty” oraz „Usługi”. Z danych lecznicy wynika, że na jedną wizytę przypada średnio 1,5 usługi.
Wypisane leki	900	Encja łączy tabele „Wizyty” oraz „Leki”. Z danych lecznicy wynika, że na jedną wizytę przypadają średnio 0,75 wypisane leki.

## 5.2 Analiza użycia identyfikująca poszczególne transakcje

### 1. Dodanie nowego klienta

- dodanie nowych danych osobowych
- dodanie nowego adresu i przypisanie go danych osobowych
- dodanie nowej krotki w tabeli „Klienci” i przypisanie danych osobowych do unikalnego numeru klienta.

## **2. Dodanie karty zwierzęcia**

- dodanie nowej krotki w tabeli „Karty zwierząt”
- wyszukanie lub dodanie nowego gatunku i przypisanie go do kart zwierzęcia
- wyszukanie klienta i przypisanie go do karty zwierzęcia

## **3. Umówienie wizyty**

- wyświetlenie wszystkich terminów dyżurów w interesującym nas dniu
- utworzenie nowej krotki w tabeli „Wizyty” z polem Status Wizyty – „Umówiona”
- wyszukanie interesującej nas karty zwierzęcia i przypisanie jej do wizyty
- wyszukanie interesującego nas pracownika z uprawnieniami lekarza i przypisanie go do wizyty
- utworzenie nowej krotki w tabeli „Terminy Wizyt”
- przypisanie terminu wizyty do wizyty oraz do terminarza lekarza

## **4. Sprzedaż leku na rachunek**

- wyszukanie interesującego nas klienta
- utworzenie nowej krotki w tabeli „Rachunki”.
- wyszukanie interesującego nas leku w tabeli „Magazyn” i „Leki”.
- utworzenie nowej krotki w tabeli „Sprzedane leki” z polem cena równym cenie pozycji znalezionej w magazynie i przypisanie leku do rachunku

## **5. Sprzedaż leku bez rachunku**

- wyszukanie interesującego nas leku w tabeli „Magazyn” i „Leki”.
- utworzenie nowej krotki w tabeli „Sprzedane leki” z polem cena równym cenie pozycji znalezionej w magazynie i przypisanie leku do rachunku

## **6. Przeprowadzenie wizyty**

- wyszukanie interesującej nas wizyty
- modyfikacja pola Status Wizyty na „ Odbyta”
- wyszukanie przeprowadzonej usługi (możliwe kilkakrotnie)
- utworzenie nowej krotki w tabeli „ Wykonie Usługi” i przypisanie jej do wizyty (możliwe kilkakrotnie)
- wyszukanie interesującego nas leku (możliwe kilkakrotnie)
- utworzenie nowej krotki w tabeli „Wypisane leki” i przypisanie jej do wizyty (możliwe kilkakrotnie)

## **7. Wystawianie rachunku za wizytę**

- wyszukanie interesującego nas klienta
- utworzenie nowej krotki w tabeli „Rachunki”.
- wyszukanie wykonanych usług podczas wizyty
- modyfikacja pola kwota w tabeli „Rachunki”

## **8. Dodanie nowego leku do Magazynu**

- wyszukanie interesującego nas leku w tabeli „Leki”
- utworzenie nowej krotki w tabeli „Magazyn” lub modyfikacja istniejącej

## **9. Dodanie nowej usługi**

- utworzenie nowej krotki w tabeli „Usługi”

## **10. Modyfikacja cennika usług**

- wyszukanie interesującej nas usługi w tabeli „Usługi”
- modyfikacja pola cena

## **11. Zatrudnienie pracownika**

- dodanie nowych danych osobowych
- dodanie nowego adresu i przypisanie go danych osobowych
- utworzenie nowej krotki w tabeli „Uprawnienia”
- utworzenie nowej krotki w tabeli „Terminarz”
- utworzenie nowej krotki w tabeli „Pracownicy”

## **5.3 Sformułowanie wymagań dotyczących dostępu – określenie częstości wykonania operacji na danych.**

### **Adres**

Rejestrowanie klientów będzie wymagało dodanie nowej instancji do tej tabeli. Przewidujemy jednak, że skala na jaką działa nasza lecznica nie spowoduje dużego przyrostu tej tabeli. Dlatego też operacje wstawiania określamy tutaj jako rzadką. Zakłada się, że w ciągu tygodnia przybywa kilku klientów. W systemie dane klientów nie będą usuwane a modyfikacja będzie się zdarzała sporadycznie w przypadku wykrycia błędu. Ze względu na fakt, że czynności takie jak umówienie wizyty czy wypisanie rachunku wiązały się będą z dostępem do danych osobistych najczęstszą operacją dokonywaną na tej tabeli będzie wyszukiwanie, dlatego czynność tą określiliśmy jako bardzo częstą.

## **Dane osobowe**

Tabela spełnia podobną funkcję co tabela „Adres”, ponieważ przechowuje dane klientów oraz pracowników firmy. Dlatego częstość wykonywania na niej operacji określono w ten sam sposób co w poprzedniej tabeli. Wstawianie będzie wykonywane kilka razy w tygodniu, modyfikacja będzie operacją jeszcze rzadszą natomiast usuwanie nie powinno mieć miejsca. Najczęstszą operacją będzie wyszukiwanie, ponieważ będzie się ono łączyło z głównymi funkcjonalnościami systemu.

## **Gatunki**

Za każdym razem kiedy będzie tworzona karta zwierzęcia trzeba będzie określić jego gatunek. Jeżeli przyjmowane zwierzę jest pierwszym z danego gatunku, trzeba będzie zaktualizować dane w tej tabeli. Zatem operacja wstawiania będzie z początku dosyć częsta lecz z upływem czasu gatunki zwierząt będą się powtarzały, więc operacja wstawiania została określana na bardzo rzadką. Zakładamy, że wprowadzone gatunki nie będą modyfikowane oraz usuwane. Natomiast wyszukiwanie będzie występowało codziennie, ponieważ z każdą wizytą będzie wiązało się określenie gatunku przyjmowanego zwierzęcia.

## **Karty zwierząt**

Zazwyczaj z każdym nowym klientem wiąże się nowa karta zwierzęcia. Wobec tego wstawianie w tej tabeli będzie czynnością powtarzaną kilka razy w tygodniu. Modyfikacja będzie operacją sporadyczną a usuwanie nie będzie miało miejsca. Ze względu na fakt, że z każdą wizytą wiąże się jakaś karta zwierząt wyszukiwanie będzie operacją bardzo częstą.

## **Klienci**

Podobnie jak w przypadku kart zwierząt częstość wstawiania została określona na operację rzadką. Tutaj również najczęstszym przypadkiem będzie wyszukiwanie, ponieważ będzie się ono odbywało codziennie, podczas wystawiania rachunków bądź umawiania wizyt.

## **Leki**

Zakładamy, że przy otwieraniu działalności trzeba będzie uzupełnić dane leków dostępnych na rynku. Z czasem niektóre leki są wycofywane a inne wprowadzane na rynek. Dlatego operacje wstawiania oraz usuwania określiliśmy na bardzo rzadką. Lek który został

dodany do tabeli nie będzie modyfikowany. Wyszukiwanie danych z tej tabeli będzie operacją wykonywaną codziennie ze względu na usługę sprzedawania leków oraz wystawiania ich podczas wizyty.

### **Magazyn**

Zakup nowych leków odbywa się raz bądź kilkakrotnie w miesiącu, w zależności od zapotrzebowania oraz możliwości finansowych firmy. Kiedy dany lek osiągnie ilość krytyczną będzie wysyłane powiadomienie o uzupełnienie stanu danego produktu. Zatem wstawianie oraz usuwanie w tej tabeli będzie operacją rzadką. Z każdym kolejnym dniem konieczna będzie aktualizacja tabeli a więc modyfikacja określona została jako operacja wykonywana często, podobnie jak wyszukiwanie leku.

### **Pracownicy**

Podobnie jak w tabeli „Leki” podczas otwierania działalności koniecznym jest zatrudnienie personelu. Jednak w czasie normalnego funkcjonowania lecznicy zmian dokonywanych będzie bardzo mało i dotyczyć będą ewentualną zmianą uprawnień bądź terminarza. Zatrudnianie lub zwalnianie pracowników odbywać się będzie tylko w uzasadnionych przypadkach, na przykład w momencie nie wypełniania swoich obowiązków. Dlatego też operacje wstawiania, modyfikacji oraz usuwania określono jak bardzo rzadkie. Natomiast z każdą wizytą wiązało się będzie wyszukiwanie a więc będzie miało ono miejsce codziennie.

### **Rachunki**

Codziennie będzie wystawiany jakiś rachunek więc wstawianie w tej tabeli jest na porządku dziennym. Ze względu na fakt, że rachunek posiada swój status modyfikacja zawartości tej tabeli będzie również operacją wykonywaną często. Założyliśmy jednak, że usuwanie z tej tabeli nie będzie miało miejsca ze względu na konieczność przechowywania dokumentacji związanej z działalnością lecznicy.

### **Sprzedane leki**

Konieczność wystawienia rachunku za sprzedane leki sprawia, że wyszukiwanie w tej tabeli będzie operacją bardzo częstą. Lecznicy prowadzi też sprzedaż leków jako osobną usługę,



co powoduje, że nawet kilkakrotnie w ciągu dnia będzie konieczność wstawiania nowych wierszy do tabeli. Zakładamy, że modyfikacja oraz usuwanie nie będzie tutaj potrzebna.

### **Terminarz**

Jest to tabela która łączy dwie bardzo ważne elementy systemu a więc terminy wizyt oraz terminy dyżurów. Niezbędnym elementem do sprawnego działania przedsiębiorstwa jest odpowiednie zarządzanie jego czasem. Jednak do jednego pracownika będzie przypisywany jeden terminarz a więc wstawianie, modyfikacja czy usuwanie w tej tabeli będą sporadyczne. Wyszukiwanie natomiast będzie się odbywało codziennie a więc zostało określone jako operacja wykonywana bardzo często.

### **Terminy dyżurów**

Jest to jedna z tabel na których wszystkie operacje będą wykonywane często. Założyliśmy, że dyżury będą wyznaczane maksymalnie dwa miesiące do przodu, natomiast dyżury które miały już miejsce będą usuwane dwa miesiące po ich upływie. Dlatego operacje wstawiania oraz usuwania będą często wykonywane, podobnie w przypadku modyfikowania oraz wyszukiwania, ponieważ będą się one wiązały z wizytami.

### **Terminy wizyt**

Sytuacja z tą tabelą ma się podobnie jak w poprzedniej, a więc będzie ona modyfikowana kilkakrotnie w ciągu dnia. Dane o przeprowadzonych wizytach również będą usuwane z ubiegiem określonego czasu. Wyszukiwanie będzie się odbywało za każdym razem kiedy klient będzie się chciał umówić na wizytę a kiedy znaleziony zostanie odpowiadający termin trzeba będzie wstawić kolejny wiersz w tej tabeli. To wszystko sprawia, że ruch generowany w tej tabeli będzie jednym z większych w naszym systemie.

### **Uprawnienia**

Wszelkie operacje dokonywane w systemie będą wymagały odpowiednich uprawnień. Uprawnienia są przydzielane do pracowników wraz z ich zatrudnieniem, podczas pracy w lecznicy można zmieniać uprawnienia danego pracownika. Wstawianie, modyfikacja oraz usuwanie będą zatem wykonywane rzadko. Wyszukiwanie natomiast będzie wykonywane podczas większości operacji wykonywanych w systemie.

## **Usługi**

Tabela pełni funkcję cennika. Dlatego też musi ona zostać uzupełniona po otwarciu działalności. Wraz z biegiem czasu proponowane usługi oraz ich ceny mogą się zmieniać jednak nie będzie to zbyt częsta sytuacja, ponieważ mogłaby ona doprowadzić do dezorientacji klientów. Zakładamy więc, że oferta lecznicy nie będzie się często zmieniała dlatego operacje wstawiania, modyfikowania oraz usuwania zostały określone jako operacje bardzo rzadkie. Jednak za każdym wystawianiem rachunku oraz przeprowadzaniem wizyty trzeba będzie określić wykonaną usługę, co prowadzi do konieczności wyszukiwania w tej tabeli. Wyszukiwanie określono jako operacje wykonywaną często.

## **Wizyty**

Wszelkie informacje na temat wizyt będą zapisywane w tej tabeli. Dlatego też będzie ona modyfikowana nawet kilkanaście razy dziennie. Odnosi się to zarówno do wstawiania jak i modyfikowania tabeli. Zakładamy, że informacje na temat przeprowadzonych wizyt nie będą usuwane. Z każdą wizytą czy wystawianiem rachunku będzie się wiązało przeszukiwanie tabeli tak więc operacja ta również została jako częsta.

## **Wykonane usługi**

Tabela ta tworzy połączenie pomiędzy cennikiem a określoną wizytą. Dzięki niej będzie możliwe wyliczenie należności naliczonych podczas wizyty. Dlatego będzie ona modyfikowana nawet kilka-kilkanaście razy dziennie. Zakłada się, że w razie nieprawidłowo naliczonych opłat bądź innej pomyłki istnieje możliwość usunięcia wpisanej usługi. Natomiast ich modyfikacja nie powinna być umożliwiana. Wyszukiwanie będzie występowało za każdym razem kiedy będzie wyliczana opłata za usługę oraz podczas prowadzenia różnego rodzaju raportów.

## **Wypisane leki**

Podobnie jak tabela opisana wyżej tworzy ona połączenie pomiędzy lekami a wizytą na której zostały one wypisane z tą różnicą, że nie będą z tego powodu naliczane żadne opłaty. Dzięki tej tabeli będzie wiadomo, jakie leki zostały wypisane na określonej wizycie. Wstawianie oraz wyszukiwanie będzie operacją częstą, natomiast usuwanie oraz modyfikacja będą sporadycznie wykonywane w razie pomyłki lub zmiany opisu.

## Podsumowanie

Podsumowaniem powyższych opisów jest tabela w której zebrano częstotliwość wykonywania podstawowych operacji na tabelach znajdujących się w systemie.

Nazwa encji	Wstawianie	Modyfikacja	Usuwanie	Wyszukiwanie
Adres	Rzadko	<u>Bardzo rzadko</u>	Brak	<u>Bardzo często</u>
Dane osobowe	Rzadko	<u>Bardzo rzadko</u>	Brak	<u>Bardzo często</u>
Gatunki	<u>Bardzo rzadko</u>	Brak	Brak	<u>Bardzo często</u>
Karty zwierząt	Rzadko	<u>Bardzo rzadko</u>	Brak	<u>Bardzo często</u>
Klienci	Rzadko	<u>Bardzo rzadko</u>	Brak	<u>Bardzo często</u>
Leki	<u>Bardzo rzadko</u>	Brak	<u>Bardzo rzadko</u>	Często
Magazyn	Rzadko	<u>Bardzo często</u>	<u>Bardzo rzadko</u>	Często
Pracownicy	<u>Bardzo rzadko</u>	<u>Bardzo rzadko</u>	<u>Bardzo rzadko</u>	Często
Rachunki	<u>Bardzo często</u>	Często	Brak	Często
Sprzedane leki	<u>Bardzo często</u>	Brak	Brak	Często
Terminarz	<u>Bardzo rzadko</u>	Brak	<u>Bardzo rzadko</u>	<u>Bardzo często</u>
Terminy dyżurów	Często	Często	Często	Często
Terminy wizyt	Często	Często	Często	Często
Uprawnienia	<u>Bardzo rzadko</u>	<u>Bardzo rzadko</u>	<u>Bardzo rzadko</u>	<u>Bardzo często</u>
Usługi	<u>Bardzo rzadko</u>	<u>Bardzo rzadko</u>	<u>Bardzo rzadko</u>	Często
Wizyty	Często	Często	Brak	Często
Wykonane usługi	Często	Brak	<u>Bardzo rzadko</u>	Często
Wypisane leki	Często	<u>Bardzo rzadko</u>	<u>Bardzo rzadko</u>	Często

## 5.4 Analiza integralności

Integralność łączy w sobie formalną poprawność bazy danych i procesów przetwarzania, poprawność fizycznej organizacji danych, zgodność ze schematem bazy danych, zgodność z ograniczeniami integralności oraz z regułami dostępu.

Jednym ze sposobów zachowania integralności danych są więzy integralności, czyli warunki, które powinny być spełnione przez określony podzbiór danych z bazy. Spełnianie tych warunków świadczy, że baza danych jest w stanie spójnym.

- **Integralność encji**

Odnosi się do pojedynczej tabeli, w której powinien istnieć klucz pierwotny. Jeżeli danej kolumnie nałożyliśmy warunek *PRIMARY KEY*, to automatycznie zostaną nałożone jej warunki *NOT NULL* oraz *UNIQUE*.

W naszych tabelach umieściliśmy takie atrybuty, które jednoznacznie charakteryzują klucz pierwotny danej tabeli. W większości przypadków są to identyfikatory ID lecz w niektórych przypadkach są to inne kolumny jak np. PESEL w tabeli Dane osobowe.

- **Integralność krotki**

Zakłada się, że każda krotka opisuje jeden obiekt świata rzeczywistego a wartość krotki powinna odpowiadać elementowi świata rzeczywistego. Na wartości przyjmowane przez krotki można nałożyć niezależne więzy, które muszą być spełnione przez wszystkie krotki niezależnie. Więzy te to:

- zawężenie dziedziny atrybutu poprzez podanie przedziału wartości, listy możliwych zawartości,

- podanie formatu wartości, np. *VARCHAR(20)*,

- zadeklarowanie konieczności występowania jakiejś wartości - *NOT NULL*,

- zdefiniowanie niepowtarzalnych wartości atrybutu - *UNIQUE*

- **Integralność odwołań**

Dotyczy modelowania powiązań pomiędzy obiektami w rzeczywistości. W bazach relacyjnych realizowane jest to z wykorzystaniem klucza pierwotnego i odpowiadającego mu wartością klucza obcego Tworząc klucz obcy, definiujemy związek między tabelą klucza

pierwotnego i tabelą klucza obcego. Związek taki powstaje podczas połączenia kolumn tych samych typów danych z każdej tabeli.

W relacyjnych bazach danych integralność odwołań dotyczy sytuacji, kiedy tabela A zawiera klucz obcy będący równocześnie kluczem pierwotnym tabeli B. Warunek integralności odwołań ustala, że dla każdego wiersza tabeli A musi istnieć taki wiersz w tabeli B, że wartość klucza obcego i pierwotnego są jednakowe. Integralność ta może zostać naruszona poprzez usunięcie wiersza w tabeli, której klucz główny jest kluczem obcym w innej tabeli. W celu uniknięcia sytuacji odwoływania się do nieistniejących elementów zdecydowaliśmy się na różne rozwiązania. Jednym z nich będzie wpisywanie wartości *NULL* zamiast odwołania do konkretnego wiersza w innej tabeli. Mimo wszystko w naszym systemie operacja usuwania nie zdarza się często, dlatego liczba takich sytuacji jest mocno ograniczona.

- **Asercje**

Asercje to ograniczenia występujące w schemacie jak niezależne od tabel obiekty, służą do kontroli wartości wprowadzanych do tabel. Stosuje się je przede wszystkim do pól, których wartości musi być ograniczona oraz wartość ta powinna być sprawdzona pod kryterium spełniania określonych wymagań. Dla systemu bazy lecznicy asercje można efektywnie wykorzystać przy kontroli wpisywanych cen czy też kosztów:

- w tabeli Magazyn do kontroli ceny leków - pole to nie może wynosić ani *NULL*, ani wartość nie może być ujemna. Osiągniemy to dzięki poleceniu:

```
CREATE ASSERTION sprawdzenie ceny CHECK(magazyn.cena >=0);
```

W asercji nie trzeba już dodatkowo sprawdzać czy pole wynosi *NULL*, ze względu na klauzule '*NOT NULL*' przy tworzeniu tego atrybutu, ale oczywiście istnieje taka możliwość.

- w tabeli Rachunki do sprawdzenia cen wykonanych usług
- w tabeli Usługi przy weryfikowaniu poprawności kosztu odpowiedniej usługi
- w tabeli Karty zwierząt w celu sprawdzenia poprawności liczebności

W ramach potrzeby istnieje możliwość dodawania kolejnych asercji np. w razie wystąpienia błędów czy w celu usprawnienia działania bazy.

Po przeanalizowaniu integralności nasze tabele będą miały następującą postać.

<b>ADRES</b>	
ID_ADRES	INTEGER(5) , PK
MIEJSCOWOSC	VARCHAR2(20), NOT NULL
ULICA	VARCHAR2(20), NOT NULL
OPIS	VARCHAR2(255)
NR BUDYNKU	INTEGER(3), NOT NULL
NR MIESZKANIA	INTEGER(3)
KOD POCZTOWY	INTEGER(5),NOT NULL

<b>DANE OSOBOWE</b>	
PESEL	INTEGER(10) , PK
IMIE	VARCHAR2(25), NOT NULL
NAZWISKO	VARCHAR2(20), NOT NULL
TELEFON	INTEGER(10)
EMAIL	VARCHAR2(20)
ADRES_ID	INTEGER(5),FK

<b>KLIENCI</b>	
ID_KLIENTA	INTEGER(5) , PK
PESEL	VARCHAR2(25),FK, NOT NULL, UNIQUE

<b>RACHUNKI</b>	
ID_RACHUNKU	INTEGER(10) , PK
KWOTA	INTEGER(5)
DATA	DATE,NOT NULL
STATUS	VARCHAR2(12), NOT NULL,CHECK IN('zapłacony','niezapłacony')
ID_KLIENTA	INTEGER(5) , FK, NOT NULL
ID_WIZYTY	INTEGER(5),FK

<b>KARTY ZWIERZAT</b>	
ID_KARTY	INTEGER(5) , PK
LICZEBNOSC	INTEGER(5), NOT NULL
ROK URODZENIA	DATE
ID_ADRES	INTEGER(5) , FK
ID_KLIENTA	INTEGER(5) FK
ID_GATUNKU	INTEGER(5),FK

<b>GATUNKI</b>	
ID_GATUNKU	INTEGER(5) , PK
NAZWA	VARCHAR2(30), NOT NULL
RASA	VARCHAR2(30)

<b>WIZYTY</b>	
ID_WIZYTY	INTEGER(5) , PK
TYP	varchar2(10), NOT NULL,CHEC IN ('w lecznicy', 'u pacjenta')
OPIS	VARCHAR2 (255)
STATUS WIZYTY	varchar2(10), NOT NULL,CHECK IN ('umowiona', 'odbyta')
ID_PRACOWNIKA	INTEGER(5), FK,NOT NULL
ID_KARTY	INTEGER(5), FK
ID_RACHUNKU	INTEGER(5), FK,UNIQUE
ID_TERMINU_WIZYTY	INTEGER(5),FK, UNIQUE,NOT NULL

<b>PRACOWNICY</b>	
ID_PRACOWNIKA	INTEGER(5) , PK
NR UBEZBIECZENIA	INTEGER(10),UNIQUE
NR KONTA	INTEGER(10),UNIQUE
PESEL	INTEGER(10) , FK
ID_UPRAWNIENIA	INTEGER(5), FK,NOT NULL, UNIQUE
ID_TERMINARZA	INTEGER(5),FK, NOT NULL, UNIQUE
STAN ZATRUDNIENIA	varchar2(25), NOT NULL, CHECK IN('zatrudniony', 'zwolniony','urlop')

<b>UPRAWNIENIA</b>	
ID_UPRAWNIENIA	INTEGER(5) , PK
ADMINISTRATOR	BIT, NOT NULL
KIEROWNIK	BIT, NOT NULL
LEKARZ	BIT, NOT NULL
RECEPCJONISTA	BIT, NOT NULL
MAGAZYNIER	BIT, NOT NULL

<b>SPRZEDANE LEKI</b>	
ID_SPRZEDANEGO_LEKU	INTEGER(5) , PK
CENA	INTEGER(5), NOT NULL
DATA	DATE , NOT NULL
ID_RACHUNKU	INTEGER(5) , FK, UNIQUE
ID_LEKU	INTEGER(5) FK, UNIQUE, NOT NULL

<b>WYPISANE LEKI</b>	
ID_WIZYTY	INTEGER(5), PK,FK, NOT NULL
ID_LEKU	INTEGER(5), PK,FK, NOT NULL
OPIS	VARCHAR(255)

<b>WYKONANE USŁUGI</b>	
ID_WIZYTY	INTEGER(5), PK,FK, NOT NULL
ID_USŁUGI	INTEGER(5), PK,FK, NOT NULL
OPIS	VARCHAR(255)

<b>TERMINY WIZYT</b>	
ID_TERMINU_WIZYTY	INTEGER(5) , PK
DATA	DATE, NOT NULL
GODZINA	TIME,NOT NULL
ID_TERMINARZA	INTEGER(5),FK,UNIQUE,NOT NULL



<b>TERMINARZ</b>	
ID_TERMINARZA	INTEGER(5) , PK

<b>TERMINY DYŻURÓW</b>	
ID_TERMINU_WIZYTY	INTEGER(5) , PK
DATA	DATE, NOT NULL
GODZINA STARTU	TIME,NOT NULL
GODZINA KONCOWA	TIME,NOT NULL
ID_TERMINARZA	INTEGER(5),FK,UNIQUE,NOT NULL

<b>MAGAZYN</b>	
ID_PRODUKTU	INTEGER(5) , PK
ID_LEKU	INTEGER(10) , FK, UNIQUE,NOT NULL
CENA	INTEGER(5),NOT NULL
ILOSC	INTEGER(5) , NOT NULL
KRYTYCZNA ILOSC	INTEGER(5)

<b>LEKI</b>	
ID_LEKU	INTEGER(5) , PK
NAZWA	VARCHAR2(30),NOT NULL, UNIQUE
PRODUCENT	VARCHAR2(30),NOT NULL

<b>USŁUGI</b>	
ID_USLUGI	INTEGER(5) , PK
NAZWA	VARCHAR2(30),NOT NULL, UNIQUE
CENA	INTEGER(5),NOT NULL

## 5.5 Dostrajanie bazy danych pod względem wydajności

- **Indeksowanie**

Przeprowadzona we wcześniejszych punktach analiza umożliwiła nam na identyfikację encji dla , których najczęściej stosowane zostaną operacje wyszukiwania. Postanowiliśmy w pewnych tabelach, w których uznaliśmy to za opłacalny zabieg, wprowadzić indeksowanie.

Każda tabela w relacyjnym modelu bazy danych zawiera klucz główny. Należy zatem pamiętać, że każdy klucz główny jest także indeksem.

Szczególnie często wykonywanym zabiegiem w naszej bazie danych będzie wyszukiwanie danych osobowych po imieniu i nazwisku. Postanowiliśmy skorzystać zatem z indeksu obejmującego dwa pola, a zatem `imię` i `nazwisko`. Dodatkowo przewidujemy dodanie indeksu na samo pole `nazwisko`, ponieważ równie często wyszukiwać będziemy klientów po nazwisku. Kolejnym często wykonywanym zabiegiem w naszej bazie będzie wyszukiwanie rachunków. Szczególnie często wyświetlać będziemy je względem daty wystawienia oraz statusu rachunku. Wprowadzimy zatem indeks na pole `Data` oraz na pole `Status`. Ze względu na dużą liczbę danych przechowywanych w tabeli „Leki” oraz stosunkowo częste wyszukiwanie leków po ich nazwie nałożymy indeks w tabeli Leki na pole `Nazwa`. Podobna sytuacja do powyższej zachodzi dla tabeli „Usługi”. Tutaj również wprowadzony indeks obejmie pole „Nazwa”. Częstym zabiegiem okazuje się także wyszukiwanie wizyt po ich terminie. W tym przypadku stworzony indeks dotyczył będzie pola `Data` w tabeli „Terminy wizyt”. Analogiczna sytuacja występuje w tabeli „Terminy dyżurów”. Kolejny indeks, które zamierzamy wprowadzić do naszej bazy danych to indeks w tabeli „Sprzedane leki” na pole `Data`, który przyspieszy wyszukiwanie sprzedanych leków w danym dniu. Podczas składania zamówień na leki często przydatne okaże się sprawdzenie pozycji o niskim stanie magazynowym, dlatego opłacalne będzie wprowadzenie indeksu na pole `Ilość` w tabeli „Magazyn”. Przy zakładaniu karty zwierzęcia podawany zawsze jest jego gatunek. Operacja to zostanie przyspieszona przez wprowadzenie indeksu w tabeli „Gatunki” na pole „Nazwa”. W tabeli „Wizyty” wprowadzimy indeks na pole `Status wizyty`. Przyspieszy to wyszukiwanie umówionych wizyt.

<b>Tabela</b>	<b>Indeks</b>
Dane osobowe	(imię, nazwisko), (nazwisko)
Rachunek	(status), (data)
Sprzedane leki	(data)
Terminy wizyt	(data)
Terminy dyżurów	(data)
Leki	(nazwa)
Usługi	(nazwa)
Magazyn	(ilość)
Gatunki	(nazwa)
Wizyty	(status wizyty)

- **Denormalizacja**

Denormalizacja to proces wprowadzania kontrolowanej nadmiarowości w bazie danych i zmiana logicznego projektu bazy danych na słabszą postać normalną np. 2NF w celu uzyskania możliwości szybszego wykonywania często występujących zapytań i transakcji.

Z przeprowadzonej przez nas analizy stwierdziliśmy, że szczególnie częstym zapytaniem, które wymagałoby dużo czasochłonnych łączy tabel byłoby wyświetlanie konkretnych pozycji na rachunku. Postanowiliśmy zatem w tabeli „Sprzedane leki” wprowadzić dodatkowo pole cena, które niejako jest redundancją pola cena z tabeli magazyn. Umożliwi to dużo szybsze wyświetlenie pozycji rachunku bez dołączania tabel „Leki” oraz „Magazyn”