System obsługi kliniki weterynaryjnej

Kierunek: Informatyka i Ekonometria

Spis treści

[Opis ogólny systemu 1](#_Toc198303912)

[Cel 1](#_Toc198303913)

[Zakres 1](#_Toc198303914)

[Kontekst 2](#_Toc198303915)

[Wymagania funkcjonalne: 2](#_Toc198303916)

[Modele 3](#_Toc198303917)

[Model pojęciowy 3](#_Toc198303918)

[Model logiczny 3](#_Toc198303919)

[Model implementacyjny 3](#_Toc198303920)

[Opis zaimplementowanych funkcji systemu 7](#_Toc198303921)

[Moduły systemu: 8](#_Toc198303922)

[Formularz Menu 8](#_Toc198303923)

[Formularz Wizyty 9](#_Toc198303924)

[Formularz Rachunki 10](#_Toc198303925)

[Formularz Klienci 11](#_Toc198303926)

[Formularz Zwierzęta 12](#_Toc198303927)

[Formularz Weterynarze 13](#_Toc198303928)

[Formularz Usługi\_w\_wizycie 14](#_Toc198303929)

[Raport Rachunki 14](#_Toc198303930)

# 

# Opis ogólny systemu

## Cel

System ma za zadanie ułatwić organizacje pracy w klinice weterynaryjnej poprzez zarządzanie danymi posiadanymi przez klinikę, a także umożliwia wprowadzenie nowych i aktualizowanie danych.

## Zakres

System zapewnia sprawniejsze prowadzenie rejestracji klientów i ich zwierząt. Pozwala na tworzenie wizyt, zawierających listę usług, które mają być wykonane w jej ramach oraz wybranie zwierzęcia, dla którego umawiana jest wizyta. Pozwala również na dodanie rachunku na podstawie wykonanych usług oraz wygenerowanie recepty gotowej do wydruku i przekazania do klienta. Umożliwia także sprawne wprowadzanie do systemu nowych danych dotyczących weterynarzy, asystentów oraz usług.

## Kontekst

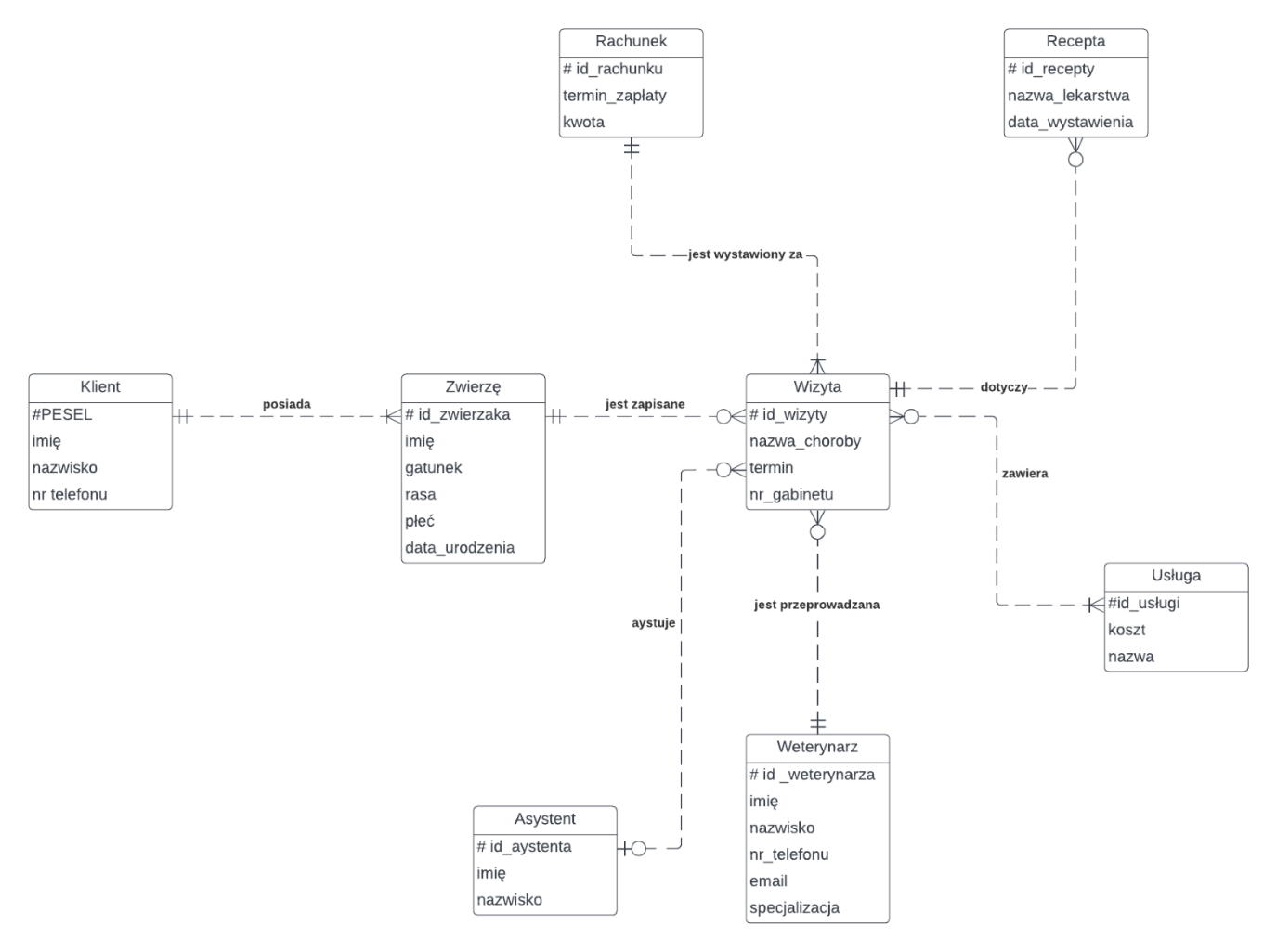
System przeznaczony jest dla biurowego pracownika kliniki weterynaryjnej który musi zadbać o sprawne funkcjonowanie kliniki oraz aktualizować dane.

# Wymagania funkcjonalne:

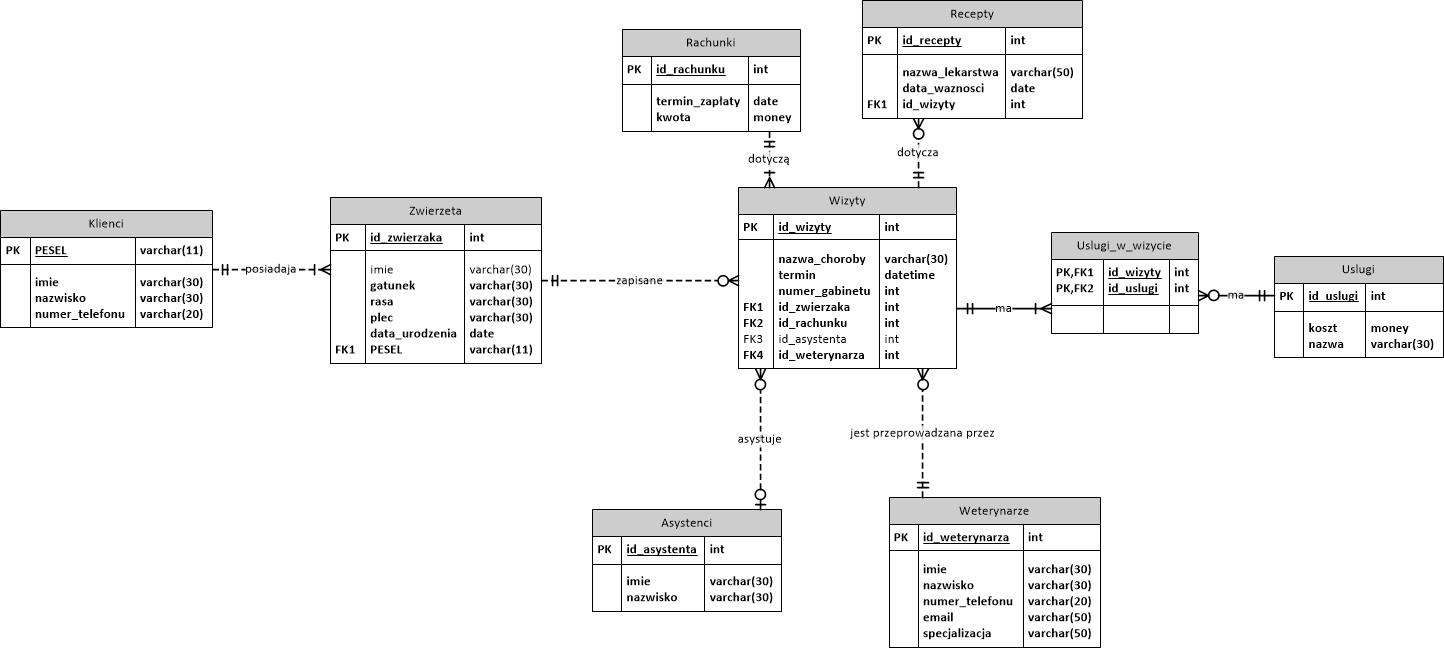
* Rejestracja klientów (PESEL, imię, nazwisko, nr telefonu, email) w systemie wraz z ich zwierzętami (imię, gatunek, rasa, płeć, data urodzenia). Klient musi posiadać co najmniej jedno zwierzę. Jedno zwierzę ma tylko jednego właściciela.
* Zarządzanie katalogiem weterynarzy (imię, nazwisko,nr\_telefonu,email, specjalizacja) w systemie: dodawanie, usuwanie, edycja i wyszukiwanie weterynarzy. Weterynarz może przeprowadzić jedną lub więcej wizyt.
* Zarządzanie katalogiem wykonywanych usług(nazwa, koszt): dodawanie, usuwanie, edycja i wyszukiwanie usług. Przyjmowanie usług do wykonania podczas wizyty. Na wizycie można wykonać jedną lub więcej usług.
* Rejestracja wizyt(nazwa\_choroby,termin,nr\_gabinetu): wizyta powinna być powiązana tylko z jednym zwierzęciem.
* Rejestracja asystentów(imie,nazwisko):do wizyty można powiązać zero lub wiele asystentów.
* Podczas wizyty, zostaje wypisana recepta(nazwa\_lekarstwa, data wystawienia). Receptę wypisuje jeden i tylko jeden lekarz. Lekarz może przepisać zero lub więcej recept.
* Generowanie rachunków za zrealizowaną wizytę(termin\_zapłaty, kwota): ustalenie terminu zapłaty oraz kwoty do zapłacenia. Jeden rachunek dotyczy tylko jednej wizyty.

# Modele

## Model pojęciowy



## Model logiczny



## 

## Model implementacyjny

|  |
| --- |
| IF EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[Recepty]') AND type = N'U')  DROP TABLE [dbo].[Recepty]  GO  CREATE TABLE [dbo].[Recepty] (  [id\_recepty] INT NOT NULL,  [nazwa\_lekarstwa] VARCHAR(50) NOT NULL,  [data\_waznosci] DATE NOT NULL,  [id\_wizyty] INT NOT NULL  )  GO  ALTER TABLE [dbo].[Recepty] ADD CONSTRAINT [Recepty\_PK] PRIMARY KEY CLUSTERED ([id\_recepty])  GO  IF EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[Klienci]') AND type = N'U')  DROP TABLE [dbo].[Klienci]  GO  CREATE TABLE [dbo].[Klienci] (  [PESEL] VARCHAR(11) NOT NULL,  [imie] VARCHAR(30) NOT NULL,  [nazwisko] VARCHAR(30) NOT NULL,  [numer\_telefonu] VARCHAR(20) NOT NULL  )  GO  ALTER TABLE [dbo].[Klienci] ADD CONSTRAINT [Klienci\_PK] PRIMARY KEY CLUSTERED ([PESEL])  GO  IF EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[Zwierzeta]') AND type = N'U')  DROP TABLE [dbo].[Zwierzeta]  GO  CREATE TABLE [dbo].[Zwierzeta] (  [id\_zwierzaka] INT NOT NULL,  [imie] VARCHAR(30) NULL,  [gatunek] VARCHAR(30) NOT NULL,  [rasa] VARCHAR(30) NOT NULL,  [plec] VARCHAR(30) NOT NULL,  [data\_urodzenia] DATE NOT NULL,  [PESEL] VARCHAR(11) NOT NULL  )  GO  ALTER TABLE [dbo].[Zwierzeta] ADD CONSTRAINT [Zwierzeta\_PK] PRIMARY KEY CLUSTERED ([id\_zwierzaka])  GO  IF EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[Asystenci]') AND type = N'U')  DROP TABLE [dbo].[Asystenci]  GO  CREATE TABLE [dbo].[Asystenci] (  [id\_asystenta] INT NOT NULL,  [imie] VARCHAR(30) NOT NULL,  [nazwisko] VARCHAR(30) NOT NULL  )  GO  ALTER TABLE [dbo].[Asystenci] ADD CONSTRAINT [Asystenci\_PK] PRIMARY KEY CLUSTERED ([id\_asystenta])  GO  IF EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[Weterynarze]') AND type = N'U')  DROP TABLE [dbo].[Weterynarze]  GO  CREATE TABLE [dbo].[Weterynarze] (  [id\_weterynarza] INT NOT NULL,  [imie] VARCHAR(30) NOT NULL,  [nazwisko] VARCHAR(30) NOT NULL,  [numer\_telefonu] VARCHAR(20) NOT NULL,  [email] VARCHAR(50) NOT NULL,  [specjalizacja] VARCHAR(50) NOT NULL  )  GO  ALTER TABLE [dbo].[Weterynarze] ADD CONSTRAINT [Weterynarze\_PK] PRIMARY KEY CLUSTERED ([id\_weterynarza])  GO  IF EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[Wizyty]') AND type = N'U')  DROP TABLE [dbo].[Wizyty]  GO  CREATE TABLE [dbo].[Wizyty] (  [id\_wizyty] INT NOT NULL,  [nazwa\_choroby] VARCHAR(30) NOT NULL,  [termin] DATETIME NOT NULL,  [numer\_gabinetu] INT NOT NULL,  [id\_zwierzaka] INT NOT NULL,  [id\_rachunku] INT NOT NULL,  [id\_asystenta] INT NULL,  [id\_weterynarza] INT NOT NULL  )  GO  ALTER TABLE [dbo].[Wizyty] ADD CONSTRAINT [Wizyty\_PK] PRIMARY KEY CLUSTERED ([id\_wizyty])  GO  IF EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[Uslugi]') AND type = N'U')  DROP TABLE [dbo].[Uslugi]  GO  CREATE TABLE [dbo].[Uslugi] (  [id\_uslugi] INT NOT NULL,  [koszt] MONEY NOT NULL,  [nazwa] VARCHAR(30) NOT NULL  )  GO  ALTER TABLE [dbo].[Uslugi] ADD CONSTRAINT [Uslugi\_PK] PRIMARY KEY CLUSTERED ([id\_uslugi])  GO  IF EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[Uslugi\_w\_wizycie]') AND type = N'U')  DROP TABLE [dbo].[Uslugi\_w\_wizycie]  GO  CREATE TABLE [dbo].[Uslugi\_w\_wizycie] (  [id\_wizyty] INT NOT NULL,  [id\_uslugi] INT NOT NULL  )  GO  ALTER TABLE [dbo].[Uslugi\_w\_wizycie] ADD CONSTRAINT [Uslugi\_w\_wizycie\_PK] PRIMARY KEY CLUSTERED (  [id\_wizyty], [id\_uslugi]  )  GO  IF EXISTS (SELECT \* FROM sys.objects WHERE object\_id = OBJECT\_ID(N'[dbo].[Rachunki]') AND type = N'U')  DROP TABLE [dbo].[Rachunki]  GO  CREATE TABLE [dbo].[Rachunki] (  [id\_rachunku] INT NOT NULL,  [termin\_zapłaty] DATE NOT NULL,  [kwota] MONEY NOT NULL  )  GO  ALTER TABLE [dbo].[Rachunki] ADD CONSTRAINT [Rachunki\_PK] PRIMARY KEY CLUSTERED ([id\_rachunku])  GO  -- Relacje (klucze obce)  ALTER TABLE [dbo].[Recepty] WITH CHECK ADD CONSTRAINT [Wizyty\_Recepty\_FK1] FOREIGN KEY ([id\_wizyty])  REFERENCES [dbo].[Wizyty] ([id\_wizyty])  GO  ALTER TABLE [dbo].[Zwierzeta] WITH CHECK ADD CONSTRAINT [Klienci\_Zwierzeta\_FK1] FOREIGN KEY ([PESEL])  REFERENCES [dbo].[Klienci] ([PESEL])  GO  ALTER TABLE [dbo].[Wizyty] WITH CHECK ADD CONSTRAINT [Asystenci\_Wizyty\_FK1] FOREIGN KEY ([id\_asystenta])  REFERENCES [dbo].[Asystenci] ([id\_asystenta])  ALTER TABLE [dbo].[Wizyty] WITH CHECK ADD CONSTRAINT [Weterynarze\_Wizyty\_FK1] FOREIGN KEY ([id\_weterynarza])  REFERENCES [dbo].[Weterynarze] ([id\_weterynarza])  ALTER TABLE [dbo].[Wizyty] WITH CHECK ADD CONSTRAINT [Zwierzeta\_Wizyty\_FK1] FOREIGN KEY ([id\_zwierzaka])  REFERENCES [dbo].[Zwierzeta] ([id\_zwierzaka])  ALTER TABLE [dbo].[Wizyty] WITH CHECK ADD CONSTRAINT [Rachunki\_Wizyty\_FK1] FOREIGN KEY ([id\_rachunku])  REFERENCES [dbo].[Rachunki] ([id\_rachunku])  GO  ALTER TABLE [dbo].[Uslugi\_w\_wizycie] WITH CHECK ADD CONSTRAINT [Wizyty\_Uslugi\_w\_wizycie\_FK1] FOREIGN KEY ([id\_wizyty])  REFERENCES [dbo].[Wizyty] ([id\_wizyty])  ALTER TABLE [dbo].[Uslugi\_w\_wizycie] WITH CHECK ADD CONSTRAINT [Uslugi\_Uslugi\_w\_wizycie\_FK1] FOREIGN KEY ([id\_uslugi])  REFERENCES [dbo].[Uslugi] ([id\_uslugi])  GO |

# .

.

.

.

.

# Opis zaimplementowanych funkcji systemu

* Dodawanie/edycja/usuwanie klientów do/z bazy danych.
* Dodawanie/edycja/usuwanie zwierząt do/z bazy danych. Łączenie zwierząt z właścicielami.
* Dodawanie/edycja/usuwanie weterynarzy pracujących w warsztacie do/z bazy danych.
* Dodawanie/edycja/usuwanie asystentów pracujących w warsztacie do/z bazy danych.
* Dodawanie/edycja/usuwanie usług wykonywanych w klinice do/z bazy danych.
* Dodawanie/edycja/usuwanie usług do wykonania w ramach pojedynczej wizyty do/z bazy danych. - Dodawanie/edycja/usuwanie nowych wizyt do/z bazy danych z użyciem danych klienta, zwierzęcia i usług do wykonania.
* Dodawanie/edycja/usuwanie rachunków utworzonych na podstawie wykonanych usług do/z bazy danych.
* Generowanie rachunków do wystawienia w formie utworzonych raportów.

# Moduły systemu:

## Formularz Menu



Pozwala użytkownikowi aplikacji na nawigowanie pomiędzy formularzami dostępnymi w aplikacji. Naciśnięcie przycisku powoduje przejście do danego formularza.

Każdy z tych formularzy zawiera następujące przyciski:  
Do operowania na rekordach:

* Nowy rekord: Dodaje nowy, czysty rekord do którego użytkownik wprowadza dane
* Usuń rekord: Usuwa aktualnie wybrany rekord z bazy danych
* Zapisz rekord: Zapisuje wprowadzone dane w wybranym rekordzie

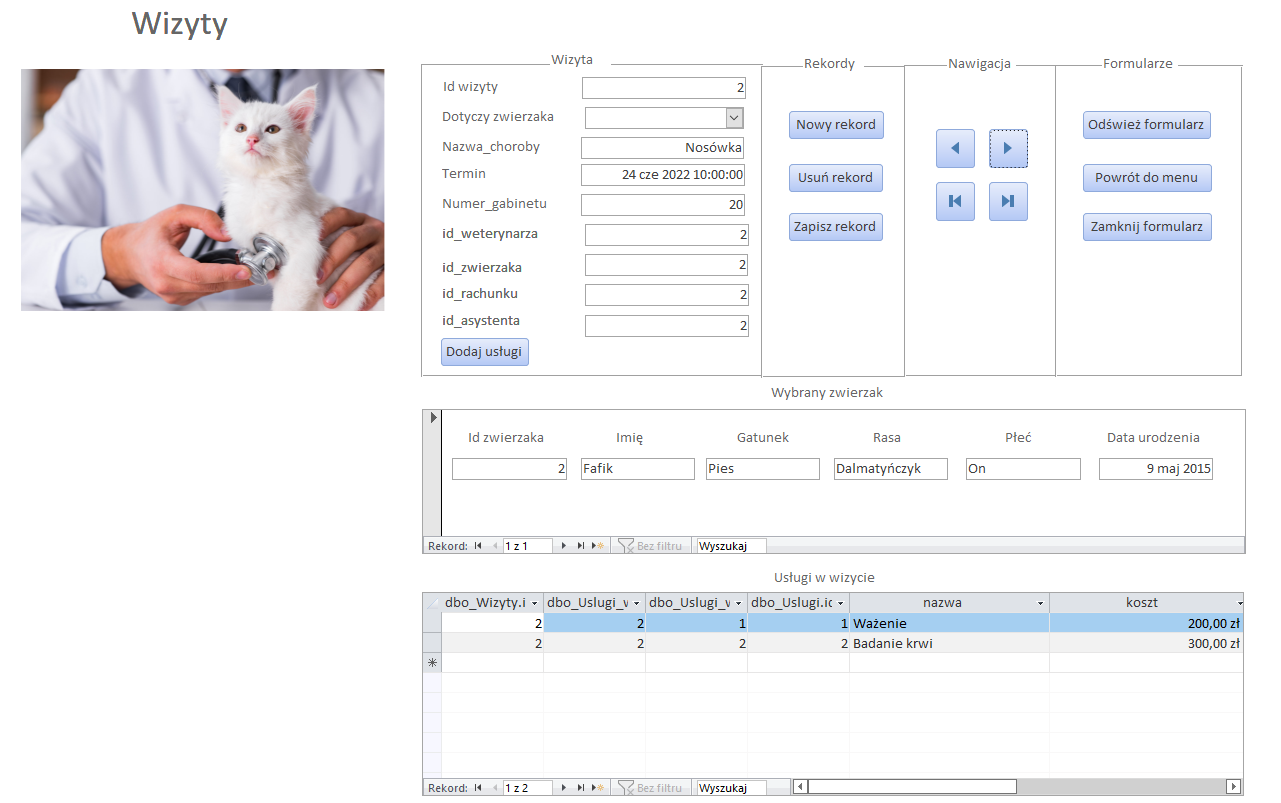
Do nawigacji na rekordach:

* Przycisk strzałki w prawo: Pokazuje kolejny rekord w bazie
* Przycisk strzałki w lewo: Pokazuje poprzedni rekord w bazie
* Ostatni rekord: Pokazuje ostatni rekord w bazie
* Pierwszy rekord: Pokazuje pierwszy rekord w bazie

Do zarządzania formularzem:

* Odśwież formularz: Odświeża otwarty formularz
* Powrót do menu: Powraca do formularza Menu
* Zamknij formularz: Zamyka formularz

## Formularz Wizyty

Po naciśnięciu przycisku „Wizyty” w formularzu Menu, zostaje otwarty formularz Wizyty. Pozwala on na dodawanie, edycje, usuwanie nowych wizyt do/z bazy danych oraz przeglądanie wcześniej dodanych wizyt.

Formualrz zawiera pola:

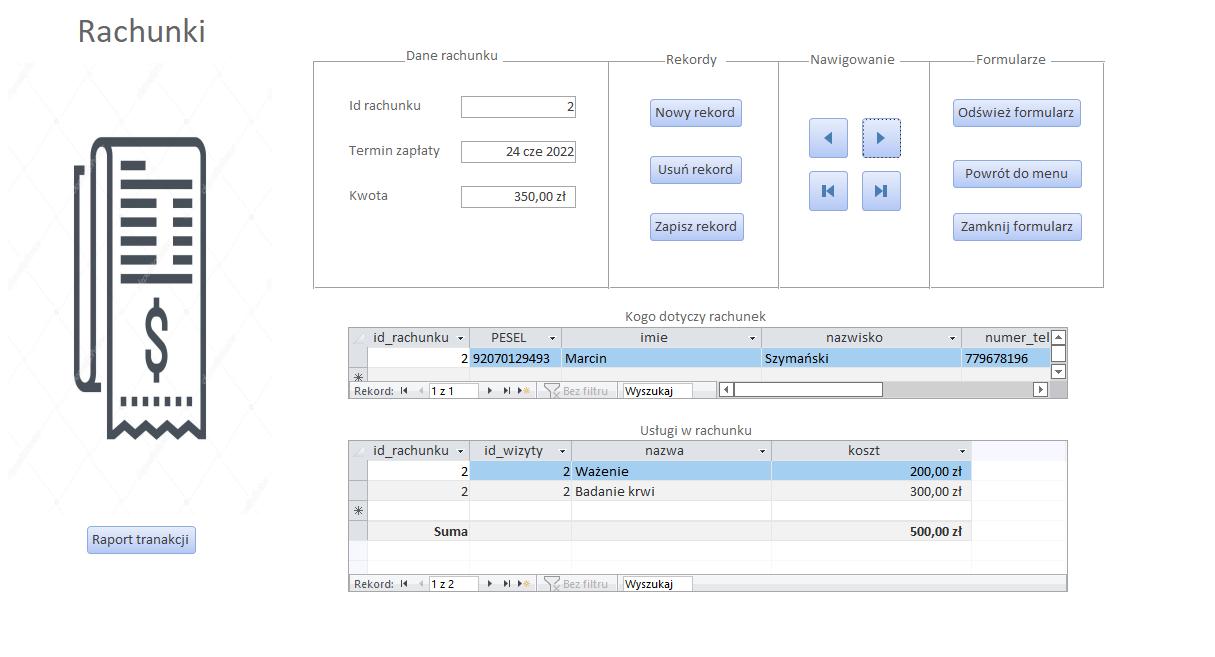
* Id wizyty: Id wizyty
* Dotyczy zwierzaka: Lista rozwijana z której można wybrać zwierzka którego dotyczyć się będzie wizyta
* Nazwa choroby: Nazwa choroby zwierzaka
* Termin: Termin wizyty
* Numer gabinetu: Numer gabinetu w którym jest wizyta
* Id weterynarza: Id weterynarza który przeprowadzi wizyte
* Id zwierzaka: Id zwierzaka zapisanego na wizyte
* Id rachunku: Id rachunku
* Id asystenta: Opcjonalny numer asystenta asystującego w wizycie

Formularz zawiera dwa podformularze:

* Informacje na temat zwierzak którego będzie dotyczyć tworzone zlecenie
* Informacje o usługach które będą wykonane w ramach wizyty

Formularz zawiera przycisk „Dodaj usługi” który otwiera formularz Usługi\_w\_wizycie, pozwalając na dodanie do wizyt usług które zostaną w ramach niej wykonane.

## Formularz Rachunki



Po naciśnięciu przycisku „Rachunki” w formularzu Menu, zostaje otwarty formularz Rachunki. Pozwala on na dodawanie, edycje, usuwanie rachunków do/z bazy danych oraz przeglądanie wcześniej dodanych rachunków.

Formularz zawiera pola:

* Id rachunku: Id rachunku
* Termin zapłaty: Termin zapłaty rachunku
* Kwota zapłaty: Kwota do zapłaty za wykonane usługi

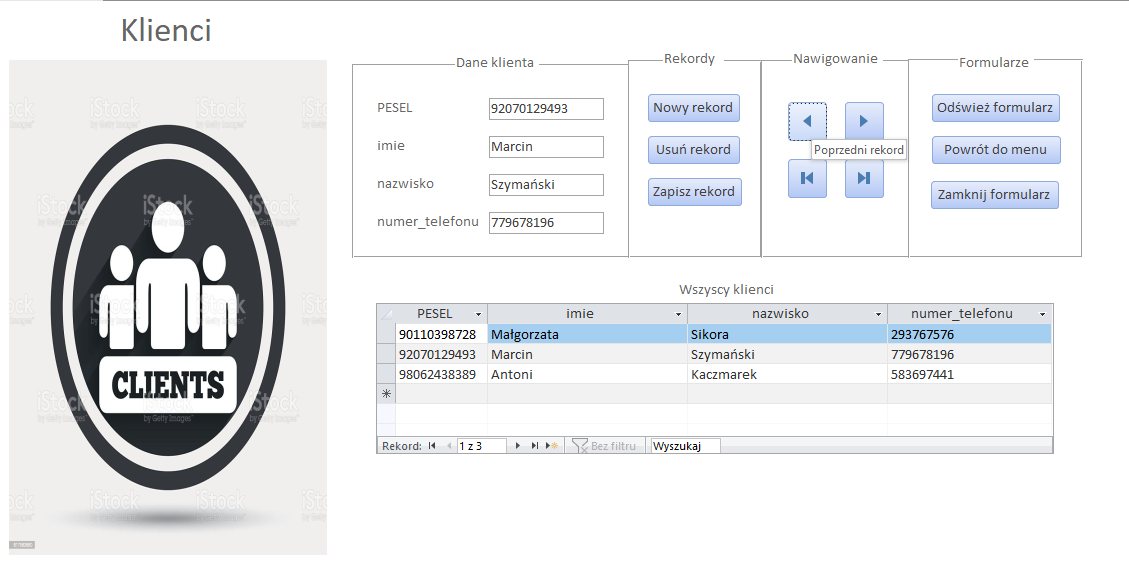
Formularz zawiera dwa podformularze:

* Informacje na temat klienta którego dotyczy wystawiony rachunek
* Informacje o usługach które zostały wykonane w ramach wizyty za które został wystawiony

rachunek oraz zsumowany koszt ich wykonania

Formularz zawiera przycisk „Raport transakcji” który wyświetla raport podsumowuący transakcie w postaci raportu gotowego do wydruku.

## Formularz Klienci



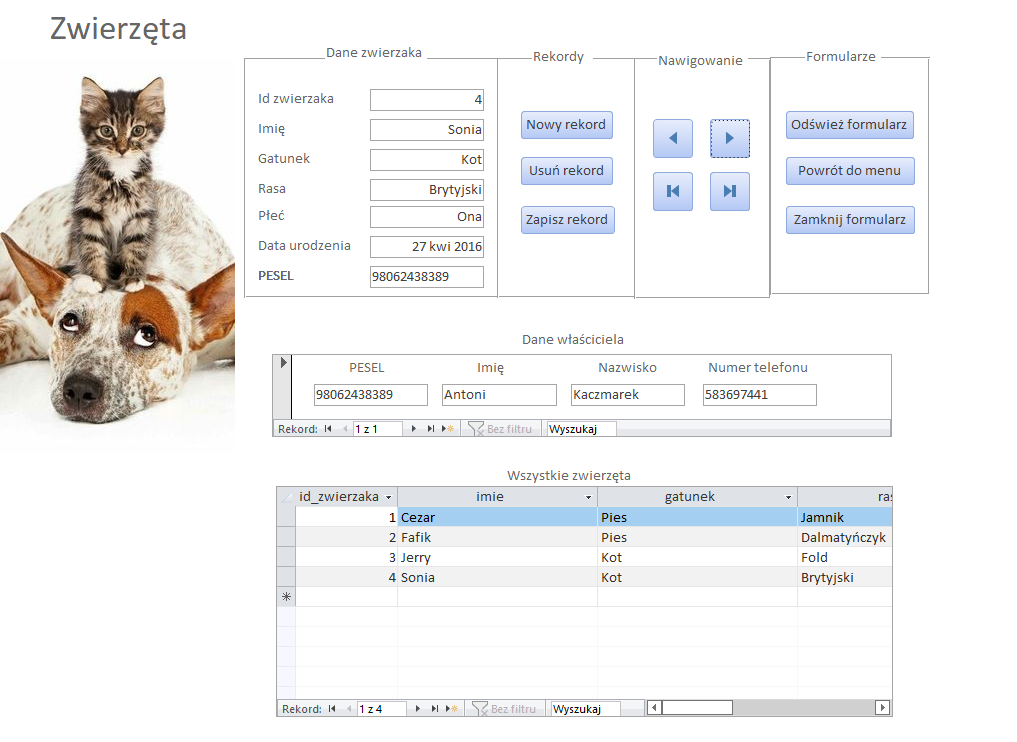
Po naciśnięciu przycisku „Klienci” w formularzu Menu, zostaje otwarty formularz Klienci. Pozwala on na dodawanie, edycje, usuwanie klientów do/z bazy danych oraz przeglądanie wcześniej dodanych klientów.

Formularz zawiera pola:

* PESEL: 11-cyfrowy unikalny PESEL klienta
* Imię: Imię Klienta
* Nazwisko: Nazwisko klienta
* Numer telefonu: Kontaktowy numer telefonu klienta

Formularz zawiera podformularz wyświetlający informacje o uprzednio dodanych klientach.

## Formularz Zwierzęta



Po naciśnięciu przycisku Zwierzęta w formularzu Menu, zostaje otwarty formularz Zwierzęta. Pozwala on na dodawanie, edycje, usuwanie zwierząt do/z bazy danych oraz przeglądanie wcześniej dodanych zwierząt.

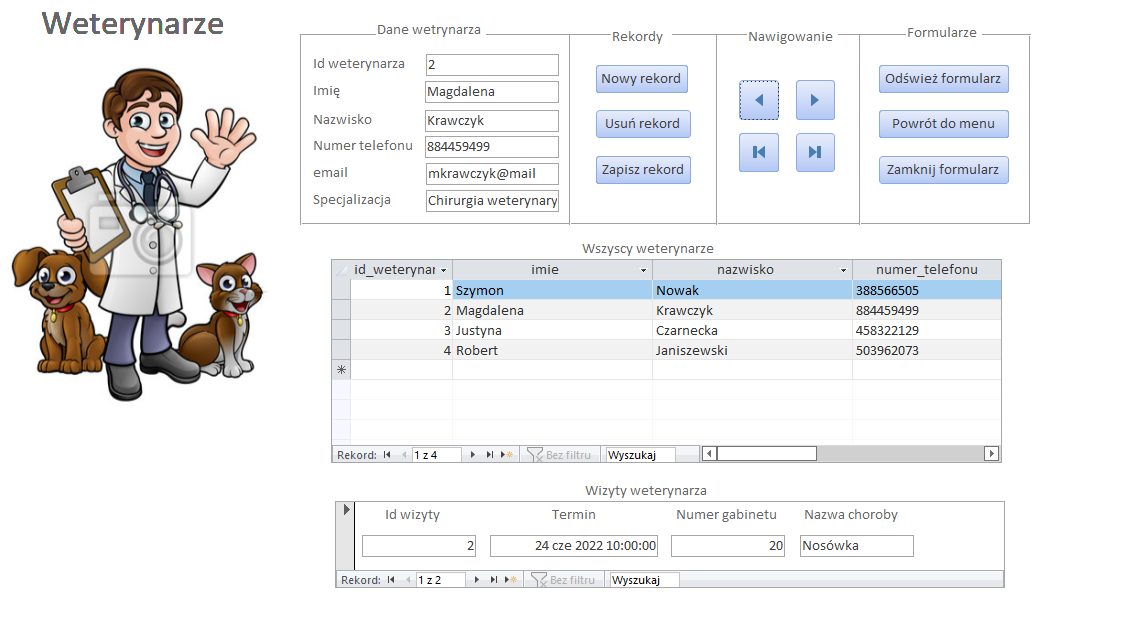
Formularz zawiera pola:

* Id zwierzaka: Unikalny numer zwierzaka
* Imię: Imię zwierzaka
* Gatunek: Gatunek zwierzaka
* Rasa: Rasa zwierzaka
* Płeć: Płeć zwierzaka
* Data urodzenia: Data urodzoenia zwierzaka
* PESEL: 11-znakowy unikalny PESEL klienta

Formularz zawiera dwa podformularze:

* Informacje na temat klienta który jest właścicielem danego zwierzaka
* Informacje o uprzednio dodanych zwierzakach

## Formularz Weterynarze



Po naciśnięciu przycisku „Weterynarze” w formularzu Menu, zostaje otwarty formularz Weterynarze. Pozwala on na dodawanie, edycje, usuwanie weterynarzy do/z bazy danych oraz przeglądanie wcześniej dodanych weterynarzy.

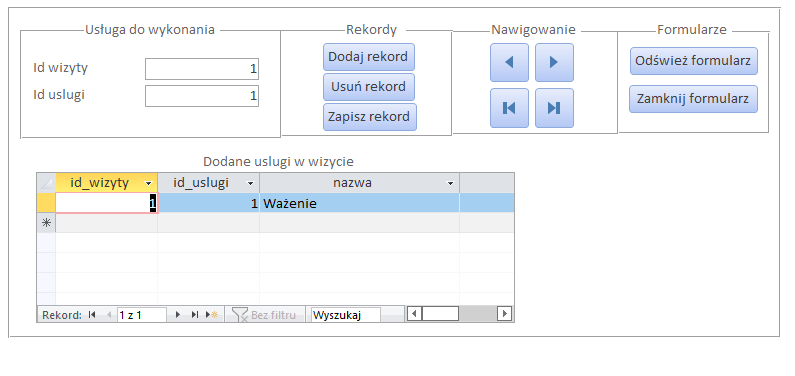
Formularz zawiera pola:

* Id weterynarz: Id weterynarza
* Imię: Imię weterynarza
* Nazwisko: Nazwisko weterynarza
* Numer telefonu: Kontaktowy numer telefonu weterynarza
* Email: Kontaktowy email weterynarza
* Specjalizacja: Specjalizacja weterynarza

Formularz zawiera dwa podformularze:

* Informacje o uprzednio dodanych weterynarzach
* Informacje o wizytach wybranego weterynarza

## Formularz Usługi\_w\_wizycie



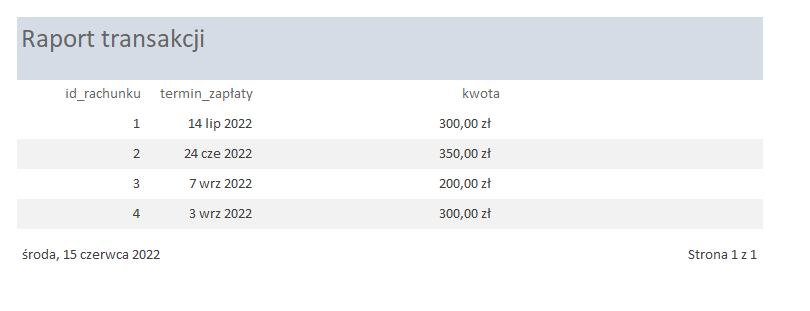
Po naciśnięciu przycisku „Dodaj usługi” w formularzu Wizyty, zostaje otwarty formularz Usługi\_w\_wizycie pozwalający na dodawanie, edycje, usuwanie usług które mają być wykonane w ramach danej wizyty do/z bazy danych oraz na przeglądanie usług które mają być wykonane w wybranym zleceniu.

Formularz zawiera pola:

* Id wizyty: Id wizyty
* Id usługi: Id usługi w ramach wizyty

Formularz zawiera podformularz wyświetlający informacje o uprzednio dodanych usługach do wykonania w danej wizycie.

## Raport Rachunki



Po naciśnięciu przycisku „Raport transakcji” w formularzu Rachunki zostaje otwarty raport Rachunki w którym znajdują się dokonane transakcje, gotowe do druku.   
Raport zawiera: informacje o Id rachunków, terminie zapłaty oraz kwocie.