

# **BAZY DANYCH I**

Dokumentacja projektu wykonywanego w ramach zajęć

Weronika Ciurej

Wydział Fizyki i Informatyki Stosowanej

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

29 stycznia 2021

## **I. Projekt koncepcji, założenia**

### **1. Temat projektu**

Tematem projektu jest obsługa bazy danych dla kliniki weterynaryjnej. Baza przechowuje informacje o zwierzętach, ich właścicielach, weterynarzach, szczepieniach, dolegliwościach, wizytach, budynkach i gabinetach. Na stronie internetowej możliwe jest stworzenie własnego profilu i umawianie zwierząt na wizyty, sprawdzenie liczby użytkowników, weterynarzy i zwierząt zapisanych w klinice a także sprawdzenie średniej ceny wizyty.

### **2. Analiza wymagań użytkownika**

Niezałogowany użytkownik ma możliwość:

- sprawdzenia statystyk kliniki.

Zalogowany użytkownik posiada możliwość:

- sprawdzenia statystyk kliniki,
- wyświetlenia swoich danych,
- wyświetlenia swoich zwierząt,
- dodania zwierzęcia,
- umówienia wizyty,
- sprawdzenia listy umówionych wizyt,
- wyświetlenia listy weterynarzy wraz z informacją o budynku i gabinecie, w którym przyjmują pacjentów.

Zalogowany weterynarz ma możliwość:

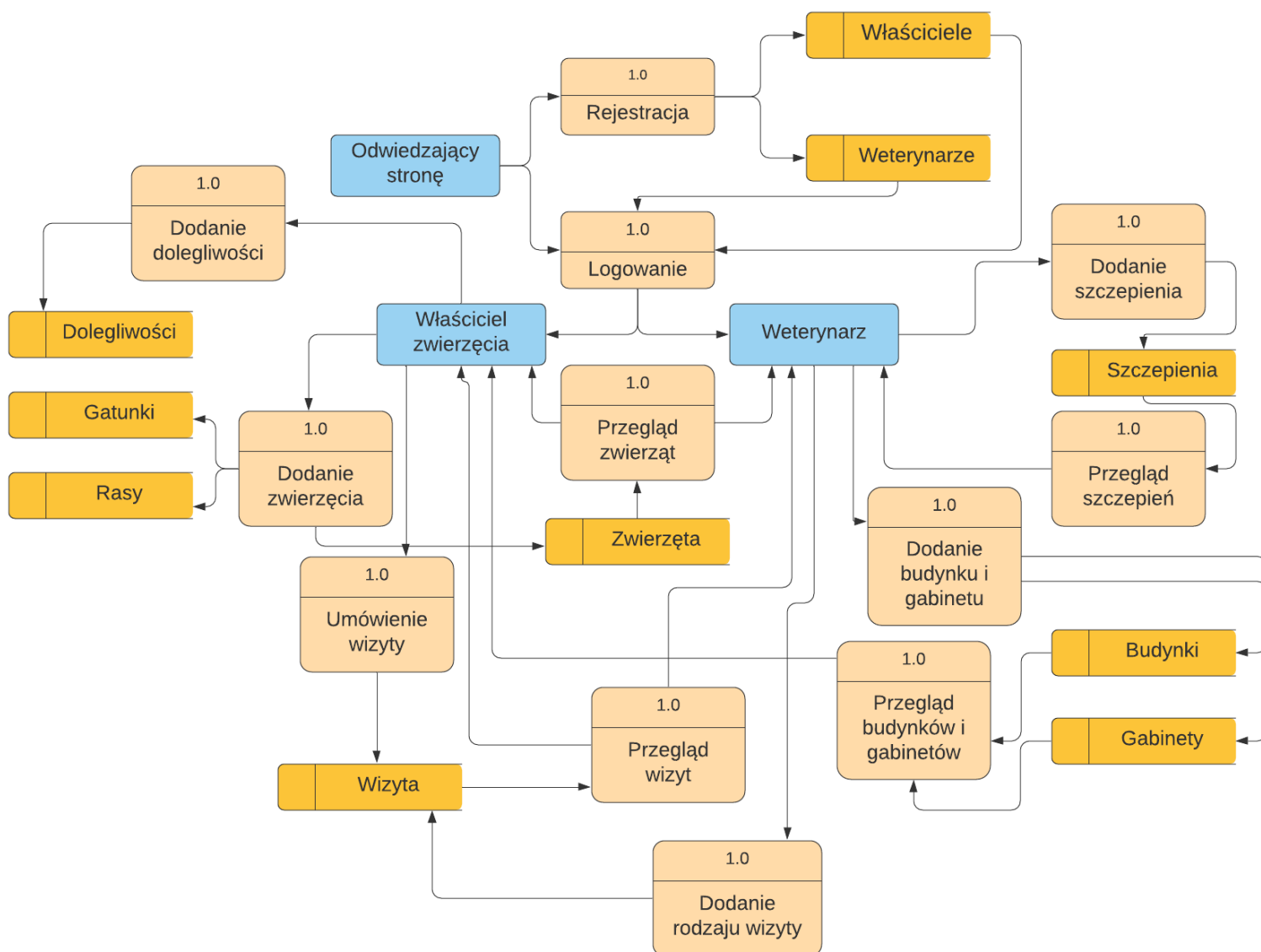
- sprawdzenia statystyk kliniki,
- wyświetlenia swoich danych,
- dodania budynku,
- dodania gabinetu,
- dodania rodzaju wizyty,
- zobaczenia umówionych do niego wizyt,
- dodania szczepienia dla zwierzęcia,
- wyświetlenia listy zaszczepionych zwierząt wraz z datą i nazwą szczepienia,
- przeglądania listy zwierząt w klinice.

### 3. Zaprojektowanie funkcji

W aplikacji zostały zaprojektowane funkcje pozwalające na wyświetlenie danych właścicieli i weterynarzy, dodanie zwierzęcia, dodania rodzaju wizyty, zapisania szczepienia dla danego zwierzęcia, umówienia wizyty przez właściciela, wyświetlenia listy zwierząt użytkownika oraz zwierząt zapisanych do zalogowanego weterynarza. Ponadto, w bazie dodana została funkcja aktualizująca statystyki.

## II. Projekt diagramów (konceptualny)

### 4. Budowa i analiza diagramu przepływu danych (DFD)



Rysunek 1. Diagram DFD

## 5. Zdefiniowanie encji (obiektów) oraz ich atrybutów

Tabele w bazie oraz ich atrybuty:

✱ Zwierzęta:

- **id\_zwierzę,**
- imię,
- id\_rasa,
- id\_właściciel.

✱ Szczepienia:

- **id\_szczepienie,**
- nazwa.

✱ Zwierzę\_Szczepienie:

- id\_zwierzę,
- id\_szczepienie,
- data.

✱ Dolegliwości:

- **id\_dolegliwość,**
- nazwa.

✱ Zwierzę\_Dolegliwość:

- id\_zwierzę,
- id\_dolegliwość.

✱ Właściciele:

- **id\_właściciel,**
- imię,
- nazwisko,
- email,
- hasło,
- numer\_telefonu.

✱ Gatunki:

- **id\_gatunek,**
- nazwa,
- id\_rasa,

✱ Rasy:

- **id\_rasa,**
- nazwa,
- wielkość.

✱ Budynki:

- **id\_budynek,**
- miasto,
- ulica,
- numer\_budynku.

✱ Gabinety:

- **id\_gabinet,**
- numer\_gabinetu,
- id\_weterynarz,
- id\_budynek.

✱ Weterynarze:

- **id\_weterynarz,**
- imię,
- nazwisko,
- email,
- hasło.

✱ Wizyta:

- **id\_wizyta,**
- nazwa.

✱ Rodzaj:

- id\_wizyta,
- cena.

✱ Zwierzę\_Wizyta:

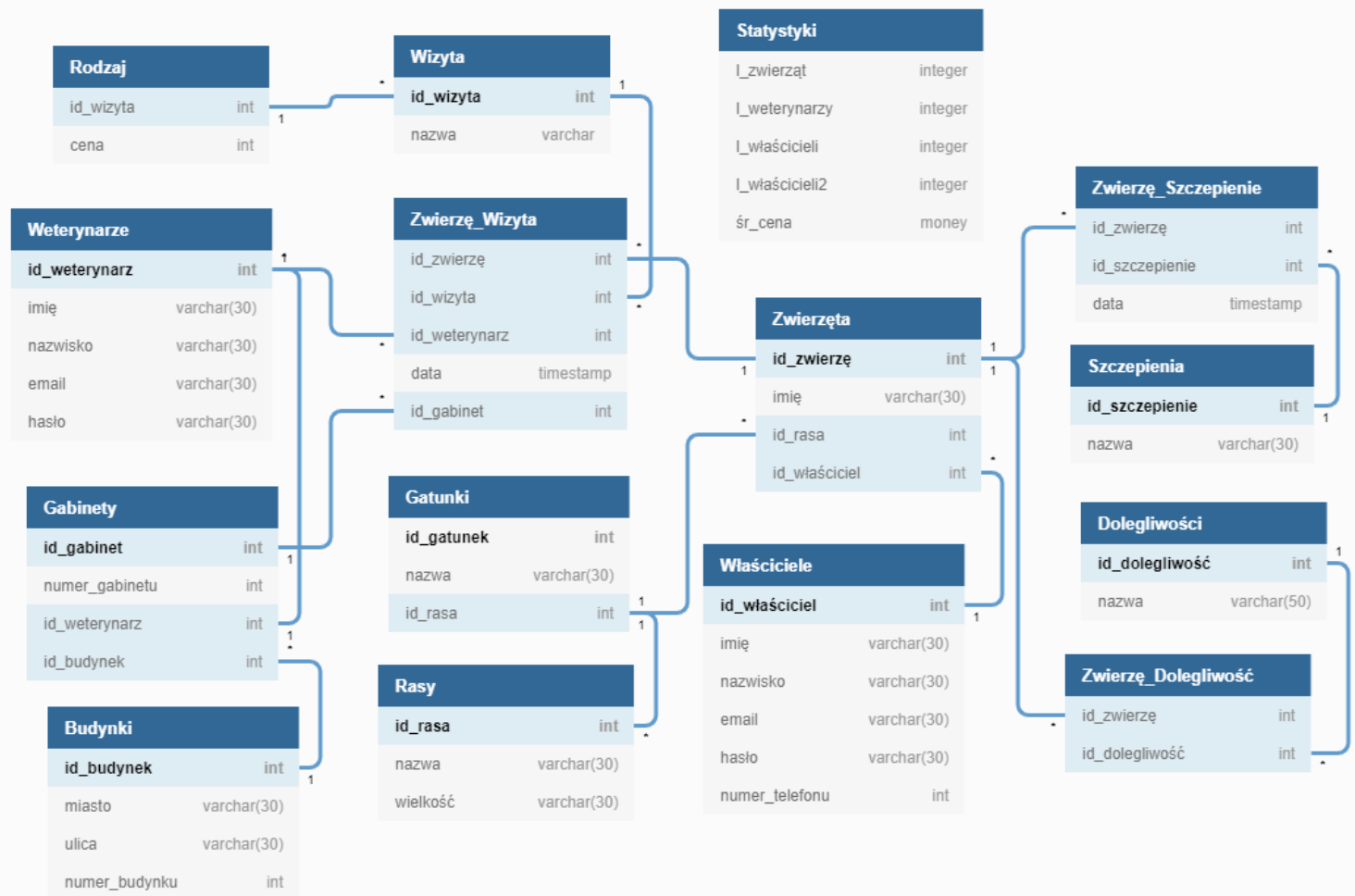
- id\_zwierzę,
- id\_wizyta,
- id\_weterynarz,
- data,
- id\_gabinet.

✱ Statystyki:

- l\_zwierząt,
- l\_weterynarzy,
- l\_właścicieli,
- l\_właścicieli2,
- śr\_cena.

## 6. Zaprojektowanie relacji pomiędzy encjami

Wszystkie relacje między encjami są typu jeden do wielu (1: N) co zostało przedstawione na poniższym diagramie ERD:



Rysunek 2. Diagram ERD

## III. Projekt logiczny

### 7. Projektowanie tabel, kluczy, indeksów

Tworzenie tabel i powiązań między nimi zapisane jest w pliku `tabele.sql` w folderze `SQL`. Większość tabel posiada klucz główny, który jest automatycznie inkrementowany. Atrybuty tabel nie mogą być pustymi wartościami. Klucze obce mają taką samą nazwę jak klucze główne w tabeli, do której się odnoszą.

## 8. Słowniki danych

**Tabela 1.** Tabela bazodanowa Właściciele

id_właściciel	SERIAL	PRIMARY KEY NOT NULL	Klucz główny generowany automatycznie
imię	varchar (30)	NOT NULL	
nazwisko	varchar (30)	NOT NULL	
email	varchar (30)	NOT NULL	
hasło	varchar (30)	NOT NULL	
numer_telefonu	int	NOT NULL	

**Tabela 2.** Tabela bazodanowa Rasy

id_rasa	SERIAL	PRIMARY KEY NOT NULL	Klucz główny generowany automatycznie
nazwa	varchar (30)	NOT NULL	
wielkość	varchar (30)	NOT NULL	

**Tabela 3.** Tabela bazodanowa Gatunki

id_gatunek	SERIAL	PRIMARY KEY NOT NULL	Klucz główny generowany automatycznie
nazwa	varchar (30)	NOT NULL	
id_rasa	int	FOREIGN KEY NOT NULL	Klucz obcy do tabeli Rasy

**Tabela 4.** Tabela bazodanowa Zwierzęta

id_zwierzę	SERIAL	PRIMARY KEY NOT NULL	Klucz główny generowany automatycznie
imię	varchar (30)	NOT NULL	
id_rasa	int	FOREIGN KEY NOT NULL	Klucz obcy do tabeli Gatunki i Rasy
Id_właściciel	int	FOREIGN KEY NOT NULL	Klucz obcy do tabeli Właściciele

**Tabela 5.** Tabela bazodanowa Dolegliwości

id_dolegliwość	SERIAL	PRIMARY KEY NOT NULL	Klucz główny generowany automatycznie
nazwa	varchar (50)		

**Tabela 6.** Tabela bazodanowa Zwierzę\_Dolegliwość

id_zwierzę	int	FOREIGN KEY NOT NULL	Klucz obcy do tabeli Zwierzęta
id_dolegliwość	int	FOREIGN KEY NOT NULL	Klucz obcy do tabeli Dolegliwości

**Tabela 7.** Tabela bazodanowa Szczepienia

id_szczepienie	SERIAL	PRIMARY KEY NOT NULL	Klucz główny generowany automatycznie
nazwa	varchar (30)	NOT NULL	

**Tabela 8.** Tabela bazodanowa Zwierzę\_Szczepienie

id_zwierzę	int	FOREIGN KEY NOT NULL	Klucz obcy do tabeli Zwierzęta
id_szczepienie	int	FOREIGN KEY NOT NULL	Klucz obcy do tabeli Szczepienia
data	timestamp	NOT NULL	

**Tabela 9.** Tabela bazodanowa Wizyta

id_wizyta	SERIAL	PRIMARY KEY NOT NULL	Klucz główny generowany automatycznie
nazwa	varchar (30)	NOT NULL	

**Tabela 10.** Tabela bazodanowa Rodzaj

id_wizyta	SERIAL	FOREIGN KEY NOT NULL	Klucz obcy do tabeli Wizyta
cena	int	NOT NULL	

**Tabela 11.** Tabela bazodanowa Zwierzę\_Wizyta

id_zwierzę	int	FOREIGN KEY NOT NULL	Klucz obcy do tabeli Zwierzęta
id_wizyta	int	FOREIGN KEY NOT NULL	Klucz obcy do tabeli Wizyta
id_weterynarz	int	FOREIGN KEY NOT NULL	Klucz obcy do tabeli Weterynarze
data	timestamp	NOT NULL	
id_gabinet	int	FOREIGN KEY NOT NULL	Klucz obcy do tabeli Gabinety

**Tabela 12.** Tabela bazodanowa Weterynarze

id_weterynarz	SERIAL	PRIMARY KEY NOT NULL	Klucz główny generowany automatycznie
imię	varchar (30)	NOT NULL	
nazwisko	varchar (30)	NOT NULL	
email	varchar (30)	NOT NULL	
hasło	varchar (30)	NOT NULL	

**Tabela 13.** Tabela bazodanowa Gabinety

id_gabinet	SERIAL	PRIMARY KEY NOT NULL	Klucz główny generowany automatycznie
numer_gabinetu	int	NOT NULL	
id_weterynarz	int	FOREIGN KEY NOT NULL	Klucz obcy do tabeli Weterynarze
id_budynek	int	FOREIGN KEY NOT NULL	Klucz obcy do tabeli Budynki

**Tabela 14.** Tabela bazodanowa Budynki

id_budynek	SERIAL	PRIMARY KEY NOT NULL	Klucz główny generowany automatycznie
miasto	varchar (30)	NOT NULL	
ulica	varchar (30)	NOT NULL	
numer_budynku	int	NOT NULL	

**Tabela 15.** Tabela bazodanowa statystyki

l_zwierząt	int		
l_weterynarzy	int		
l_właścicieli	int		
l_właścicieli2	int		
śr_cena	money		

## 9. Analiza zależności funkcyjnych i normalizacja tabel

- Pierwsza postać normalna:  
Każda tabela opisuje jeden obiekt i posiada klucz główny, który ją identyfikuje. Wartości atrybutów są atomowe.
- Druga postać normalna:  
Relacje są w 1NF. Każdy niekluczowy atrybut jest nieredukowalnie zależny od klucza głównego.
- Trzecia postać normalna:  
Relacje są w 2NF. Atrybuty nie wchodzące w skład klucza są w pełni funkcjonalnie zależne od całego klucza głównego.

## 10. Zaprojektowanie operacji na danych

Funkcje SQL zapisane są w pliku funkcje.sql w folderze SQL. Realizują one zadania przedstawione w punkcie 3:

- `dodaj_zwierze` – dodaje dane do tabel Zwierzęta, Gatunki i Rasy,
- `dodaj_wizyte_wet` – umożliwia dodanie rodzaju wizyty przez weterynarza,
- `dodaj_szczepienie` – wstawia informacje o szczepieniu zwierzęcia,
- `dodaj_wizyte_wlasc` – służy do umawiania wizyt przez właściciela,
- `moje_dane_wet` – zwraca dane weterynarza,
- `moje_dane` – zwraca dane właściciela,
- `moje_zwierzeta` – wyświetla informacje o zwierzętach należących do właściciela,
- `moi_pacjenci` - wyświetla informacje o zwierzętach zapisanych do danego weterynarza,
- `znajdz_zwierze` – zwraca id właściciela o podanym imieniu i nazwisku,
- `updatestats` – aktualizuje dane w tabeli Statystyki,
- `normalize_animals` – zmienia pierwszą literę imienia zwierzęcia z małej na dużą i zwraca trigger,
- `normalize_upper` – zmienia pierwsze litery imienia i nazwiska weterynarza/właściciela z małych na duże, sprawdza czy email zawiera znak '@' i zwraca trigger,
- `normalize_build` – zmienia pierwsze litery miasta i ulicy w tabeli Budynki z małych na duże i zwraca trigger.

W bazie wykorzystane są również widoki:

- `moje_wizyty` – zwraca informacje o umówionych wizytach właściciela,
- `moje_wizyty_wet` – zwraca informacje o umówionych wizytach właściciela
- `szczepienia_widok` – zwraca informacje o zaszczepionych zwierzętach oraz nazwy i daty szczepień,
- `wets` – zwraca dane z tabel Weterynarze, Budynki i Gabinety,
- `wszystkie_zwierzeta` – zwraca informacje o zwierzętach zapisanych w bazie,

oraz triggery:

- `upper_normal_for_users` – wykonuje procedurę `normalize_upper` przed wykonaniem zapytania INSERT do tabeli Właściele,
- `upper_normal_for_vets` – wykonuje procedurę `normalize_upper` przed wykonaniem zapytania INSERT do tabeli Weterynarze,
- `upper_normal_for_buildings` – wykonuje procedurę `normalize_build` przed wykonaniem zapytania INSERT do tabeli Budynki,
- `upper_normal_for_animals` – wykonuje procedurę `normalize_animals` przed wykonaniem zapytania INSERT do tabeli Zwierzęta.

## IV. Projekt funkcjonalny

### 11. Interfejsy do prezentacji, edycji i obsługi danych

Baza danych prezentowana jest poprzez stronę internetową. Wchodząc na stronę użytkownik widzi stronę główną kliniki. W menu bocznym może wybrać zakładkę „Informacje” prezentującą statystyki kliniki lub zakładki „Panel użytkownika”, „Panel weterynarza” służące do zalogowania lub zarejestrowania się.

## Klinika weterynaryjna



Rysunek 3. Strona główna kliniki

Wybierając dany panel odwiedzający stronę ma do wyboru opcję zalogowania lub zarejestrowania się.

## Panel użytkownika



Rysunek 4. Opcje logowania i rejestracji

Klikając daną opcję użytkownik widzi przed sobą odpowiedni formularz, po przesłaniu formularza otrzymuje informację zwrotną czy udało mu się zalogować/zarejestrować.



# Panel użytkownika

Imię:

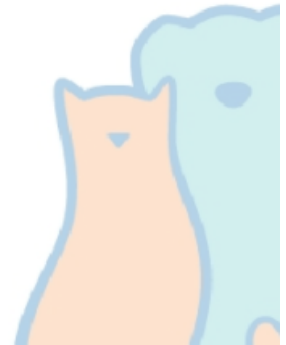
Nazwisko:

Email:

Hasło:

Numer telefonu:

Zapisz



Rysunek 5. Formularz rejestracji użytkownika

W razie niepowodzenia należy spróbować ponownie. Po skutecznym przesłaniu formularza użytkownik powinien powrócić na stronę główną klikając na guzik do tego przeznaczony, aby mieć dostęp do wszystkich funkcjonalności.

Formularze są ściśle powiązane z tabelami, ich wypełnianie jest intuicyjne, pola tekstowe są opisane.

## Umów swojego zwierzaka na wizytę

Informacje o weterynarzach i budynkach znajdziesz w zakładce "Dostępni weterynarze" w menu bocznym

Imię zwierzęcia:

Nazwisko weterynarza:

Id budynku:

Numer gabinetu:

Rodzaj wizyty:

Data: (format: RRRR-MM-DD HH:MM:SS)

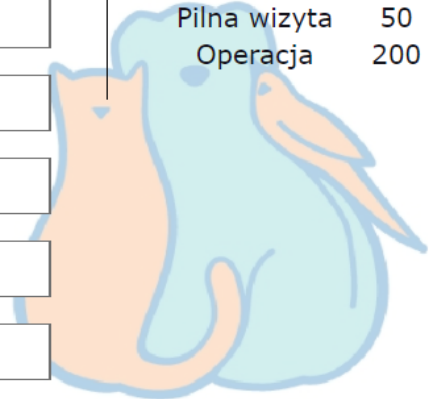
2021-02-15 14:00:00

Dolegliwości (opcjonalnie):

Umów się

## Lista wizyt:

Nazwa	Cena
Kontrola	50
Szczepienie	60
Odrobaczenie	70
Pilna wizyta	50
Operacja	200



:=User&action=showprofile

Rysunek 6. Przykładowy formularz, służy do umawiania wizyty

## 12. Wizualizacja danych

Strona napisana została głównie w języku PHP. Aby wyświetlić dane w skryptach PHP wywoływane są odpowiednie funkcje a ich rezultat jest przesyłany do użytkownika. Dane prezentowane są w formie tabel.

Zwierzę	Właściciel	Data szczepienia	Nazwa szczepienia
Milka	Monika Ciurej	2021-01-27 10:30:00	Przeciw wścieklźnie
Bruno	Weronika Ciurej	2021-02-28 14:00:00	Przeciw nosówce

Rysunek 7. Przykładowa tabela, widok zaszczepionych zwierząt

## 13. Zdefiniowanie panelu sterowania aplikacji

Panel sterowania na stronie głównej został opisany w punkcie 11. Po zalogowaniu na odpowiednim panelu menu boczne zmienia się.

Zalogowany weterynarz ma dostępne opcje:

Strona główna	• Strona główna – powrót do początkowej strony,
Informacje	• Informacje – wyświetlenie statystyk,
Moje dane	• Moje dane – wyświetlenie danych weterynarza,
Dodaj klinikę	• Dodaj klinikę – formularz do dodania budynku,
Dodaj gabinet	• Dodaj gabinet – formularz do dodania gabinetu,
Dodaj wizytę	• Dodaj wizytę – formularz do dodania rodzaju wizyty i jej ceny,
Moje wizyty	• Moje wizyty – lista umówionych wizyt,
Dodaj szczepienie	• Dodaj szczepienie – formularz do wpisania szczepienia zwierzęcia,
Przegląd szczepień	• Przegląd szczepień – lista zaszczepionych zwierząt,
Przegląd zwierząt	• Przegląd zwierząt – lista wszystkich zwierząt i zwierząt należących do danego weterynarza,
Wyloguj się	• Wyloguj się – wylogowanie z aplikacji.

Zalogowany użytkownik ma dostępne opcje:

Strona główna	• Strona główna – powrót do początkowej strony,
Informacje	• Informacje – wyświetlenie statystyk,
Moje dane	• Moje dane – wyświetlenie danych użytkownika,
Moje zwierzęta	• Moje zwierzęta – przegląd zwierząt i możliwość dodania zwierzęcia,
Moje wizyty	• Moje wizyty – lista umówionych wizyt,
Dodaj wizytę	• Dodaj wizytę – formularz do umówienia wizyty,
Dostępni weterynarze	• Dostępni weterynarze – lista weterynarzy, budynków i gabinetów
Wyloguj się	• Wyloguj się – wylogowanie z aplikacji.

## V. Dokumentacja

### 14. Wprowadzanie danych

Dane wprowadzane są do tabel ręcznie, poprzez odpowiednie formularze w interfejsie użytkownika lub weterynarza.

### 15. Dokumentacja użytkownika

Strona dostępna jest pod adresem [http://pascal.fis.agh.edu.pl/~8ciurej/projekt\\_BD](http://pascal.fis.agh.edu.pl/~8ciurej/projekt_BD).

Odwiedzający stronę może sprawdzić statystyki a także zalogować/zarejestrować się jako właściciel lub weterynarz. Jako właściciel może wpisać swoje zwierzęta i umówić je na wizytę, ma możliwość przeglądania swoich danych, zwierząt, umówionych wizyt i dostępnych weterynarzy wraz z informacjami o budynkach i gabinetach. Weterynarz posiada możliwość dodania budynku, gabinetu, rodzaju wizyty oraz informacji o zaszczepieniu zwierzęcia. Może także przeglądać swoje dane, umówione do niego wizyty, listę zaszczepionych zwierząt oraz listę wszystkich zapisanych zwierząt do kliniki i listę zapisanych do niego zwierząt.

### 16. Opracowanie dokumentacji technicznej

Strona napisana jest w języku PHP. Do tworzenia strony użyty został wzorzec MVC, który składa się z trzech komponentów programowych - modelu, widoku i kontrolera. Komponent kontrolera odpowiedzialny jest za sterowanie aplikacją. Komponent szablonów odpowiedzialny jest za przygotowanie stron przesyłanych do użytkownika. Zawiera on skrypty wprowadzające wartości zmiennych do odpowiednich szablonów przygotowanych w języku HTML. Na koniec, komponent modelu zawiera interfejs połączenia z bazą danych, zawiera on polecenia przetwarzające dane.

Klasy Controller i View znajdują się w plikach .php o tej samej nazwie, w folderze *appl*. Stworzone klasy Info, User, Wet i Zwierze umieszczone są w osobnych folderach i mają oddzielne klasy Model. Szablony wykorzystywane do wyświetlania stron znajdują się w folderze *template*. W folderze *js* znajduje się skrypt JavaScript obsługujący przesyłanie danych z formularzy do plików PHP.

Kod aplikacji udokumentowany jest w kodzie źródłowym, tj. w skryptach PHP i JavaScript.

## 17. Wykaz literatury

- Dokumentacja PostgreSQL - <https://www.postgresql.org/docs/13/index.html>
- Dokumentacja PHP - <https://www.php.net/manual/en>
- PHP Data Objects - <https://www.php.net/manual/en/book.pdo.php>
- Wykłady dr. inż. Antoniego Dydejczyka
- Narzędzie do wygenerowania diagramu ERD - <https://dbdiagram.io/home>