

Specyfikacja Wymagań

System zarządzania rehabilitacją zwierząt

Wersja 1.0

Autorzy:

Faustyna Kula

Mateusz Gruszczyk

Izabela Krzyżowska

Daria Hebda

Data : 17.12.2024 r.

SPIS TREŚCI

1.	Wprowadzenie	2
1.1.	Cel dokumentu	2
1.2.	Konwencja Dokumentu	2
1.3.	Zakres Dokumentu	2
2.	Opis	2
2.1.	Cel systemu	2
2.2.	Słownik	3
2.3.	Funkcje systemu	3
2.3.1.	Ogólne.....	3
2.3.2.	Zarządzanie danymi o zwierzętach	3
2.3.3.	Uprawnienia Użytkowników	4
3.	Architektura i wybór technologii.....	6
3.1.	Wybór technologii	6
3.2.	Wybór architektury	6
3.3.	Projekt bazy danych	7
3.4.	Przypadki użycia	8

1. WPROWADZENIE

1.1. CEL DOKUMENTU

Niniejszy dokument ma na celu opisanie specyfikacji wymagań oraz rezultatów fazy analizy związanej z projektem systemu zarządzania rehabilitacją zwierząt. Służy on do zaprezentowania głównych założeń projektu oraz proponowanych kierunków rozwoju systemu.

1.2. KONWENCJA DOKUMENTU

Niniejszy dokument napisany jest czcionką Arial w rozmiarze 9, dla nagłówków

pierwszego stopnia wykorzystano czcionkę Calibri w rozmiarze 11, dla nagłówków drugiego i trzeciego stopnia wykorzystano czcionkę Calibri w rozmiarze 10.

1.3. ZAKRES DOKUMENTU

Dokument ten opisuje plany, założenia i cele, które system powinien zrealizować zgodnie z wymaganiami klienta, przedstawionymi w sposób zwięzły i klarowny.

2. OPIS

2.1. CEL SYSTEMU

System zarządzania rehabilitacją zwierząt to aplikacja umożliwiająca kompleksowe wsparcie pracy specjalistów zajmujących się terapią weterynaryjną. System pozwala na rejestrowanie danych o zwierzętach, w tym imienia, gatunku i informacji o właścicielach, a także planowanie i zarządzanie sesjami terapeutycznymi. Kluczowe funkcje obejmują harmonogramowanie wizyt, monitorowanie postępów terapii poprzez raporty oraz przechowywanie historii leczenia. System wykorzystuje bazę danych SQL Server do bezpiecznego przechowywania danych, z zastosowaniem mechanizmów szyfrowania wrażliwych informacji. Dzięki prostemu interfejsowi użytkownika, bazującemu na aplikacji konsolowej, rozwiązanie jest intuicyjne w obsłudze, co umożliwia szybkie wdrożenie w praktyce. System został zaprojektowany z myślą o bezpieczeństwie danych i skalowalności, co pozwala na jego przyszłą rozbudowę o dodatkowe funkcje.

2.2. SŁOWNIK

T-SQL (Transact-SQL) -Rozszerzenie języka SQL (Structured Query Language) oferujące dodatkowe funkcjonalności, takie jak transakcje, procedury składowane i zarządzanie błędami. Jest używane w bazach danych Microsoft SQL Server.

Baza danych – program służący do gromadzenia i przechowywania danych w aplikacji

Back-end -Część aplikacji odpowiedzialna za logikę, dane i komunikację z bazą danych oraz innymi usługami.

Obowiązki - Zadania przypisane danemu pracownikowi, dla których dostaje przypomnienia w formie notyfikacji, które powinien wykonać, wprowadzając pozyskane z nich dane do systemu.

Użytkownik – osoba z dostępem do internetu, korzystająca z systemu i posiadająca konto w systemie.

•**Administrator** – osoba z wyższymi uprawnieniami do zarządzania systemem

2.3. FUNKCJE SYSTEMU

2.3.1. OGÓLNE

- Tworzenie konta użytkownika
- Logowanie użytkownika
- Wylogowanie użytkownika
- Modyfikacja uprawnień użytkownika
- Usunięcie konta użytkownika
- Obsługa wielu użytkowników
- Przypisanie obowiązków użytkownikowi
- Sprawdzenie stanu obowiązków użytkownika
- Sprawdzenie stanu wszystkich obowiązków

2.3.2. ZARZĄDZANIE DANYMI O ZWIERZĘTACH

- Rejestracja danych biometrycznych zwierzęcia
- Rejestracja stanu zdrowia zwierzęcia
- Rejestracja nakarmienia zwierzęcia
- Rejestracja danych inspekcji zagrody
- Sprawdzanie historii karmienia zwierzęcia
- Sprawdzanie historii inspekcji zagrody
- Sprawdzanie historii stanu zdrowia zwierzęcia
- Sprawdzanie historii danych biometrycznych zwierzęcia

- Modyfikacja historii karmienia zwierzęcia
- Modyfikacja historii inspekcji zagrody
- Modyfikacja historii stanu zdrowia zwierzęcia
- Modyfikacja historii danych biometrycznych zwierzęcia

2.3.3. UPRAWNIENIA UŻYTKOWNIKÓW

O – Opiekun

W – Weterynarz

K - Kierownik

A – Administrator

FUNKCJA	O	W	K	A
KONTA UŻYTKOWNIKÓW				
STWORZENIE				X
USUNIĘCIE				X
LOGOWANIE	X	X	X	X
WYLOGOWANIE	X	X	X	X
MODYFIKACJA UPRAWNIENÍ				X
BAZA DANYCH				
SPRAWDZENIE				
HISTORII KARMIENTA ZWIERZĘCIA	X	X	X	X
HISTORII INSPEKCJI ZAGRODY	X	X	X	X
HISTORII STANU ZDROWIA ZWIERZĘCIA	X	X	X	X
HISTORII DANYCH BIOMETRYCZNYCH ZWIERZĘCIA	X	X	X	X
REJESTRACJA				
KARMIENTA ZWIERZĘCIA	X			X
INSPEKCJI ZAGRODY			X	X
STANU ZDROWIA ZWIERZĘCIA		X		X
DANYCH BIOMETRYCZNYCH ZWIERZĘCIA		X		X
MODYFIKACJA				
HISTORII KARMIENTA ZWIERZĘCIA				X
HISTORII INSPEKCJI ZAGRODY				X
HISTORII STANU ZDROWIA ZWIERZĘCIA				X
HISTORII DANYCH BIOMETRYCZNYCH ZWIERZĘCIA				X
OBOWIĄZKI				
SPRAWDZENIE STANU	X	X	X	X
SPRAWDZENIE STANU WSZYSTKICH			X	X
PRZYPISANIE			X	X

```
using System.Data;
```

```
namespace SpaceRocket
```

```
{
```

```
    internal class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            string rocket = "    / \\  \n" +
```

```
                "    | |  \n" +
```

```
                "    | |  \n" +
```

```
                "    | |   \n" +
```

```
                "    /  \\   \n" +
```

```
                "    |   | \n" +
```

```
                "    |   | ";
```

```
            string[] tab = new string[] { " // // // \n", "  \\ \\ \\ \\ \\ \\ \\ \\ \n" };
```

```
            for (int i = 0; i < 7; i++)
```

```
            {
```

```
                Console.Clear();
```

```
                Console.WriteLine(rocket);
```

```
                Console.WriteLine(tab[i%2]);
```

```
                rocket = rocket.Insert(0, "    \n");
```

```
        Thread.Sleep(700);

    }

    Console.WriteLine("The rocket Landed!!!!");

}

}
```

3. ARCHITEKTURA I WYBÓR TECHNOLOGII

3.1. WYBÓR TECHNOLOGII

Komponent	Technologia
Język programowania	C#
Bazy danych	SQL Server (T-SQL)
Dostęp do bazy danych	EF core
Szyfrowanie danych	AES(System.Security.Cryptography)
Testowanie	NUnit lub MSTest
Kontrola wersji	Git + GitHub/GitLab

3.2. WYBÓR ARCHITEKTURY

ARCHITEKTURA TYPU KLIENT-SERWER (CLIENT-SERVER)

System będzie oparty na architekturze klient-serwer, gdzie użytkownicy (pracownicy zoo) będą korzystać z aplikacji, aby wprowadzać dane, przeglądać je, a także zarządzać informacjami o zwierzętach, inspekcjach,

karmieniu itp. Aplikacja będzie miała front-end (interfejs użytkownika), który będzie komunikować się z serwerem, który obsługuje logikę biznesową i dostęp do danych.

W tym przypadku aplikacja będzie miała podział na dwa główne elementy:

Backend (serwer): Przechowuje dane, logikę biznesową i zapewnia dostęp do danych z bazy danych.

Klient (aplikacja konsolowa): Interfejs umożliwiający użytkownikowi wykonywanie operacji, takich jak dodawanie zwierząt, planowanie sesji terapeutycznych itp.

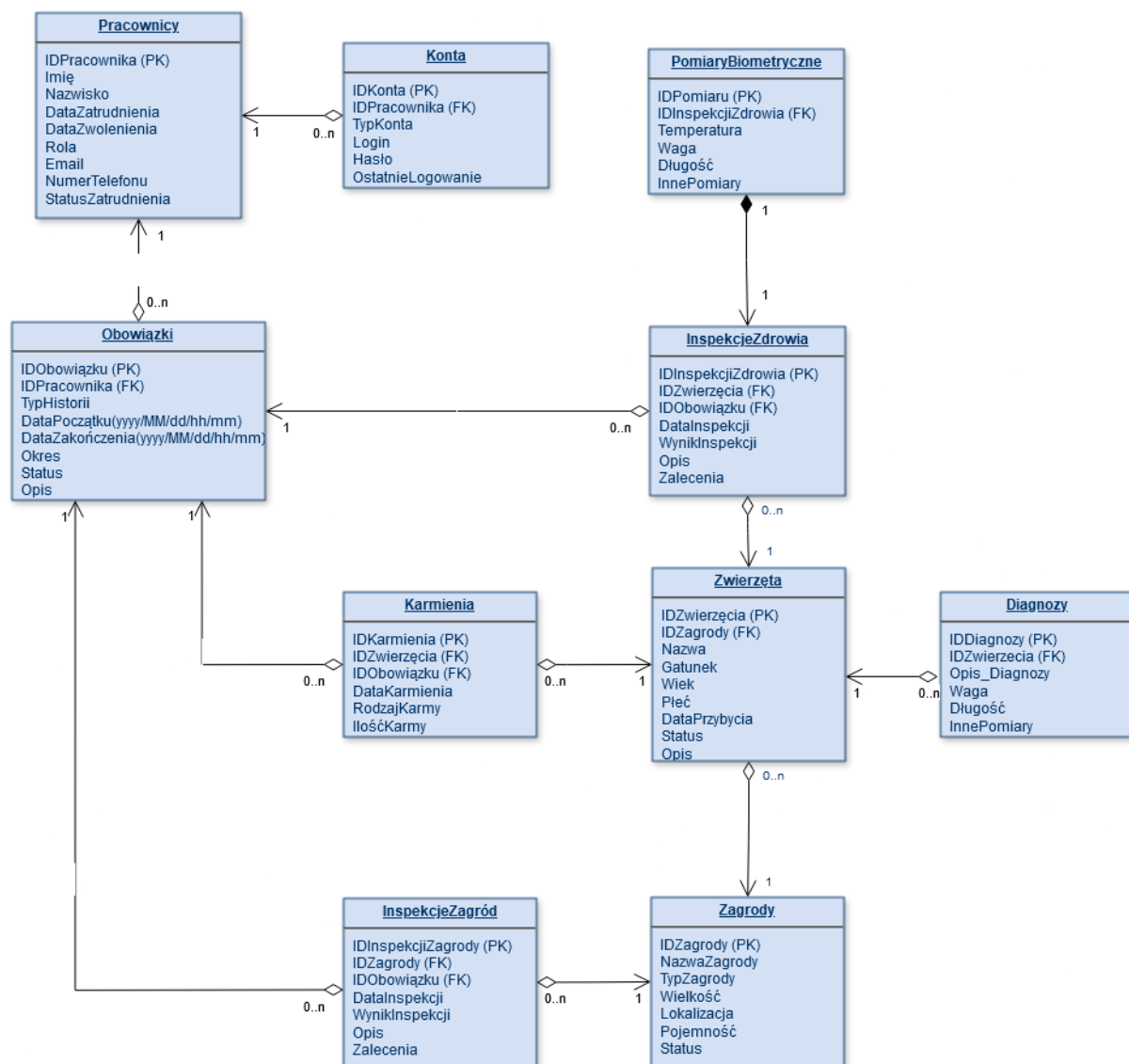
TRÓJWARSTWOWA ARCHITEKTURA (LAYERED ARCHITECTURE)

-Warstwa prezentacji: Aplikacja konsolowa. W tej warstwie użytkownik interakcjonuje z systemem, wysyłając żądania do backendu.

-Warstwa logiki biznesowej: Klasy odpowiedzialne za implementację funkcji zarządzania rehabilitacją, planowanie sesji, monitorowanie postępów.

-Warstwa dostępu do danych: Repozytoria i klasy odpowiedzialne za komunikację z bazą danych (SQL Server).

3.3. PROJEKT BAZY DANYCH



3.4. PRZYPADKI UŻYCIA

