### Wrocław, 06.06.2018

### PROGRAMOWANIE OBIEKTOWE

### INEW0003P

### Projekt

|  |  |
| --- | --- |
| Wydział Elektroniki | Kierunek: Informatyka |
| Grupa zajęciowa: Wt. 17:05 | Semestr: 2017/18 LATO |
| Nazwisko i Imię: Tobiasz Puślecki | Nr indeksu: 241354 |
| Nazwisko i Imię: Milena Korusiewicz | Nr indeksu: 241327 |
| Nr. grupy projektowej: |  |
| Prowadzący: mgr inż. Jakub Zgraja | |

TEMAT:

*Shelter – program do obsługi schroniska dla bezdomnych zwierząt*

|  |  |
| --- | --- |
| OCENA: | PUNKTY: |
| Data: |  |

## Założenia i opis funkcjonalny programu

Program wspomagający pracę schroniska dla bezdomnych zwierząt.

Pozwala ewidencjonować psy i koty.

Przechowywane dane o podopiecznym:

-Numer chipu (CID)

-Typ zwierzęcia

-Imię zwierzęcia

-Płeć

-Rasa

-Numer boxu (lokalizacja na terenie schroniska)

-Datę i miejsce odnalezienia

-Informacje nt. ewentualnych chorób

-Kolor sierści i kolor oczu

-Czy został adoptowany? (jeśli tak to można sprawdzić dane właściciela, tzn. jego imię, nazwisko i numer telefonu)

Inne funkcjonalności to:

-Moduł odpowiedzialny za adopcje

-Katalog numerów chipów odnalezionych zwierząt (po CID można odnaleźć właściciela w spisie)

Scenariusze użycia:

1. Odnalezione zwierzę trafia do schroniska, gdzie sprawdza się czy jest zachipowane. Jeśli tak, to w module „Znajdź właściciela” odszukuje się dane właściciela. W przeciwnym wypadku nadawany jest unikalny numer CID, a zwierzę dodawane do bazy w module „Znalezione”.

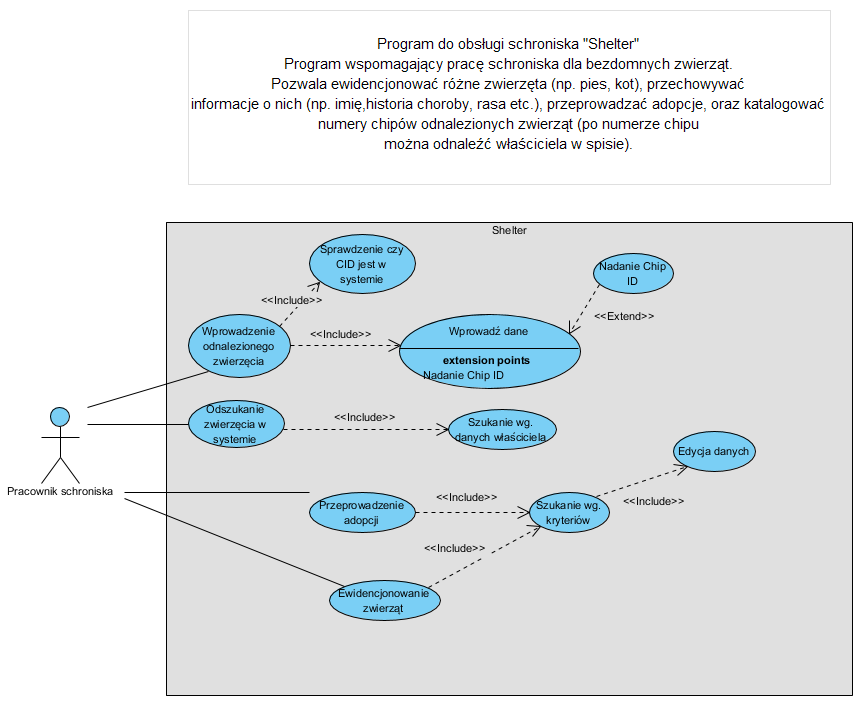
2. Właściciel przychodzi do schroniska po zgubione zwierzę i jeśli jest zarejestrowany w systemie, to na podstawie danych odszukuje się jego psa lub kota. Gdy właściciel nie był nigdy rejestrowany, należy szukać zwierzęcia przez moduły „Przeglądaj” lub „Tryb ręczny”, uwzględniając podane przez petenta cechy.

3. Przeprowadzanie adopcji, które polega na powiązaniu CID zwierzęcia z nowowprowadzonymi danymi nowego właściciela.

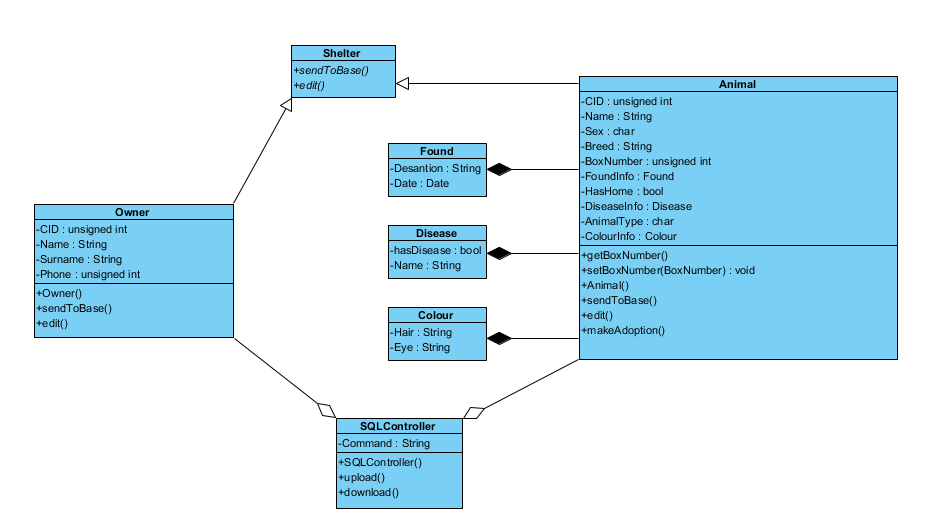
4. Możliwość szukania według zadanych kryteriów zwierząt, które je spełniają. Przeglądanie profili pozwala odnaleźć podopiecznych według np. daty przyjęcia, choroby.

## Diagramy UML

1. **Przypadków użycia**

****

1. **Klas**



## Kod klas C++ / C#

// tylko zawartość plików nagłówkowych \*.h/ \*.hpp – zarówno deklarowane zmienne jak i prototypy funkcji powinny być skomentowane

## Schematy blokowe oraz kod własnych funkcji C++ / C#

// Opisując własne funkcje podać: co funkcja robi, dane wejściowe, dane wyjściowe i wyniki, ewentualnie realizowane wzory lub algorytmy (np. qsort, minimalizacja czasu przejścia, minimalizacja liczby przeskoków itp.)

## Opis użytkowy programu C++ / C#

//opis jak poruszać się po programie wraz z zrzutami ekranu (zwłaszcza opis menu), wymagania techniczne i uwagi instalacyjne

\* ostatni etap to dokumentacja + prezentacja projektu; proszę w ramach dokumentacji zamieścić prezentację projektu -- tj. screenshoty aplikacji wraz z opisami (np. w tym widoku można zrobić ..., z tego poziomu można wybrać opcje ... itd., wraz z opisem wszystkich widoków),

## Listing kodu C++ / C# – wraz z komentarzami.

*<Wstawiono na ePortal>*

## Wnioski

Zaimplementowano podstawowe funkcjonalności i zrealizowano część początkowych założeń. Program nadaje się do użytku, jednak brakuje lepszego kreatora zapytań ręcznych, do którego obsługi nie byłaby wymagana znajomość SQL-a. Nie udało się dobrze zoptymalizować kodu oraz w niektórych miejscach sprawić, aby był on zgodny z wymaganiami. Zabrakło również sprawdzania poprawności danych przed wysłaniem ich do bazy oraz komunikacji zwrotnej z bazą (nie ma informacji czy zapytanie powiodło się).

**OPISÓWKA BEZ KODU**

**PDF**

**3**

**4**

**5**