



1 MAJA 2024

MAS – ‘KLINIKA WETERYNARYJNA’
Dokumentacja projektu

KAMIL KULESZA S24651



Spis treści

| | |
|--|----|
| 1. Wymagania użytkownika. | 2 |
| Cel projektu..... | 2 |
| Użytkownicy systemu. | 2 |
| Wymagania użytkownika prezentują się następująco. | 2 |
| 2. Diagram przypadków użycia. | 4 |
| 3. Diagram klas – analityczny. | 5 |
| 4. Diagram klas – projektowy. | 6 |
| 5. Scenariusz przypadku użycia. | 7 |
| "Rejestracja nowego zwierzęcia, wraz z możliwością rejestracji wizyty":..... | 7 |
| Scenariusz główny: | 7 |
| Scenariusz alternatywny: | 7 |
| 6. Diagram aktywności dla przypadku użycia..... | 8 |
| 7. Diagram stanu dla klasy..... | 9 |
| 8. Projekt GUI..... | 10 |
| Okno 'Formularz nowej wizyty'. | 10 |
| Okno 'Formularz nowego zwierzęcia. | 10 |
| Okno 'Formularz nowego właściciela'. | 11 |
| Okno 'Potwierdzenia'. | 11 |
| 9. Omówienie decyzji projektowych. | 12 |
| 10. Słownik pojęć. | 13 |

1. Wymagania użytkownika.

Cel projektu.

Pewna klinika weterynaryjna postanowiła usprawnić obsługę swoich pacjentów poprzez wprowadzenie systemu informatycznego, który ułatwi przechowywanie i operowanie danymi.

Użytkownicy systemu.

System ma ułatwić pracę zarówno **Weterynarzom** i **Praktykantom** pracujących w klinice, jak i **Właścicielowi Kliniki**. Pomimo tego że pewne funkcjonalności systemu będą różne dla tych użytkowników, zakłada się, że wszyscy będą pracować na tej samej aplikacji.

Wymagania użytkownika prezentują się następująco.

System ma przechowywać dane:

- Dla **Osoby**: imię, drugie imię (opcjonalnie), nazwisko, adres zamieszkania oraz numer telefonu.
- Dla **Weterynarzy**: dodatkowo informacje o specjalizacjach jakie posiada.
- Dla **Praktykantów**: całkowitą liczbę godzin potrzebną do zaliczenia praktyk oraz liczbę godzin dotychczasowo przepracowanych w klinice (na rzecz praktyk).
- Dla **Właścicieli**: ich zwierzęta, czy jest hodowcą i opcjonalnie NIP, oraz okres bycia posiadaczem danego zwierzęcia.
- Dla **Zwierząt**: imię, gatunek, rasę, płeć (jako 'M' dla męskiej i 'F' dla żeńskiej), datę urodzenia, wiek, numer mikrochipa (jeżeli posiada), dane właściciel(a/i) oraz kartotekę wizyt zwierzęcia.
- Dla **Wizyt**: zwierzę, datę, kody Zabiegów (unikalne w ramach danej Wizyty), jednego lub dwóch weterynarzy prowadzących, opcjonalnie wspierającego praktykanta, stan (zaplanowana, w trakcie, zakończona, anulowana) oraz notatkę.
- Dla **Zabiegów**: kod i bazową opłatę,
 - działają się one na wykonywane:
 - **W Klinice**: adres kliniki (który będzie taki sam dla wszystkich zabiegów) oraz potencjalny rabat dla klienta,
 - i **Poza Kliniką**: adres zabiegu, dystans od kliniki, dodatkowy koszt (na podstawie dystansu od kliniki, aktualnie jest to 2zł/km) i czy zwierzę wymagało transportu do kliniki,
 - oraz dzielą się na metodologie w których klinika się specjalizuje, czyli:
 - **Szczepienia**: kontra jaki patogen, nazwa i producent szczepionki
 - **Immunoterapię**: nazwa choroby oraz rodzaj wybranej terapii
 - (ich szczególny przypadek) **Szczepienia Immunoterapeutyczne**: na jaką przypadłość, nazwę i producenta szczepionki, minimalna data do otrzymania kolejnej dawki
 - oraz **Inne**: opis zabiegu i opcjonalne dodatkowe koszty

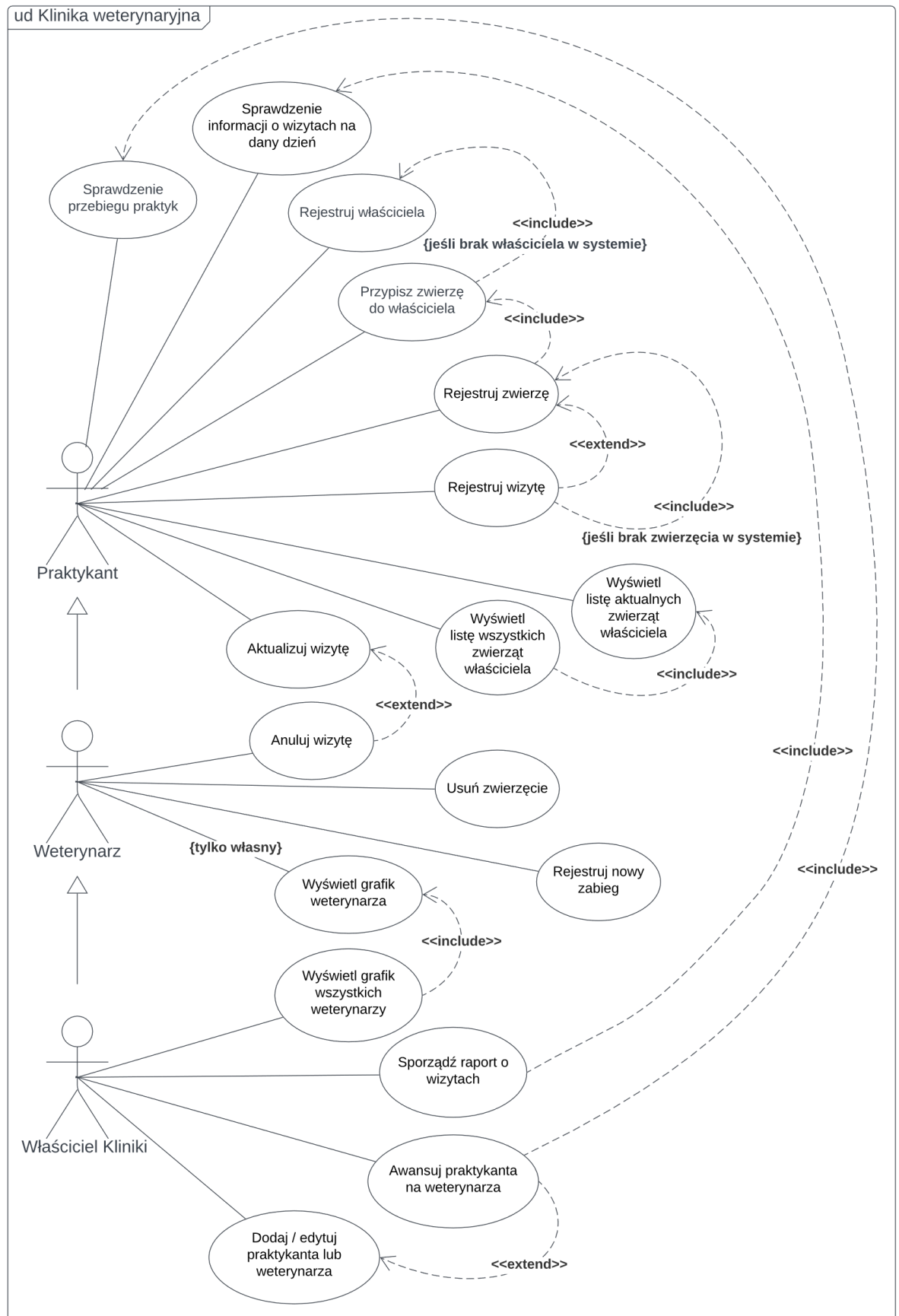
System ma również pilnować następujących ograniczeń:

- Dana **Osoba** może być zarówno Weterynarzem i Właścicielem, lub Praktykantem i Właścicielem, ale nigdy nie Praktykantem i Weterynarzem.
- Dany **Weterynarz** może być przypisany tylko do jednej Wizyty na dany czas oraz nie może być przypisany do wizyt dla swoich Zwierząt. Jeżeli jednak Weterynarz przeprowadził już Wizytę (Wizyta rejestrowana jako 'Zakończona') ze swoim Zwierzęciem, system przyjmie tą wiadomość ale oflaguje ją jako potencjalne złamanie zasad praktyk roboczych i będzie to widoczne na raporcie.
- Dany **Praktykant** po ukończeniu praktyk i zdobyciu specjalizacji może zostać Weterynarzem w klinice.
- Dany **Właściciel** może być posiadaczem wielu Zwierząt jak i dane Zwierzę może posiadać wielu Właścicieli. Z uwagi na możliwe adopcje i sprzedaże zwierząt należy również pamiętać wszystkich poprzednich Właścicieli danego Zwierzęcia oraz okres w którym byli tego Zwierzęcia Właścicielem.
- Dane **Zwierzę** można (na życzenie Właściciela) usunąć z systemu i skutkuje to usunięciem historii (i zaplanowanych) jego Wizyt.
- Do danej **Wizyty** ma być przypisany co najmniej jeden Weterynarz (max. 2), a sama ma być rejestrowana z odpowiednim stanem. Zakłada się że Wizyta może być zarejestrowana w systemie przed, (w trakcie jej trwania?) lub nawet po jej zakończeniu. Ten sam zabieg nie może się powtórzyć w ramach jednej danej Wizyty.
- Wszystkie **Zabiegi** mają unikalne względem siebie kody.

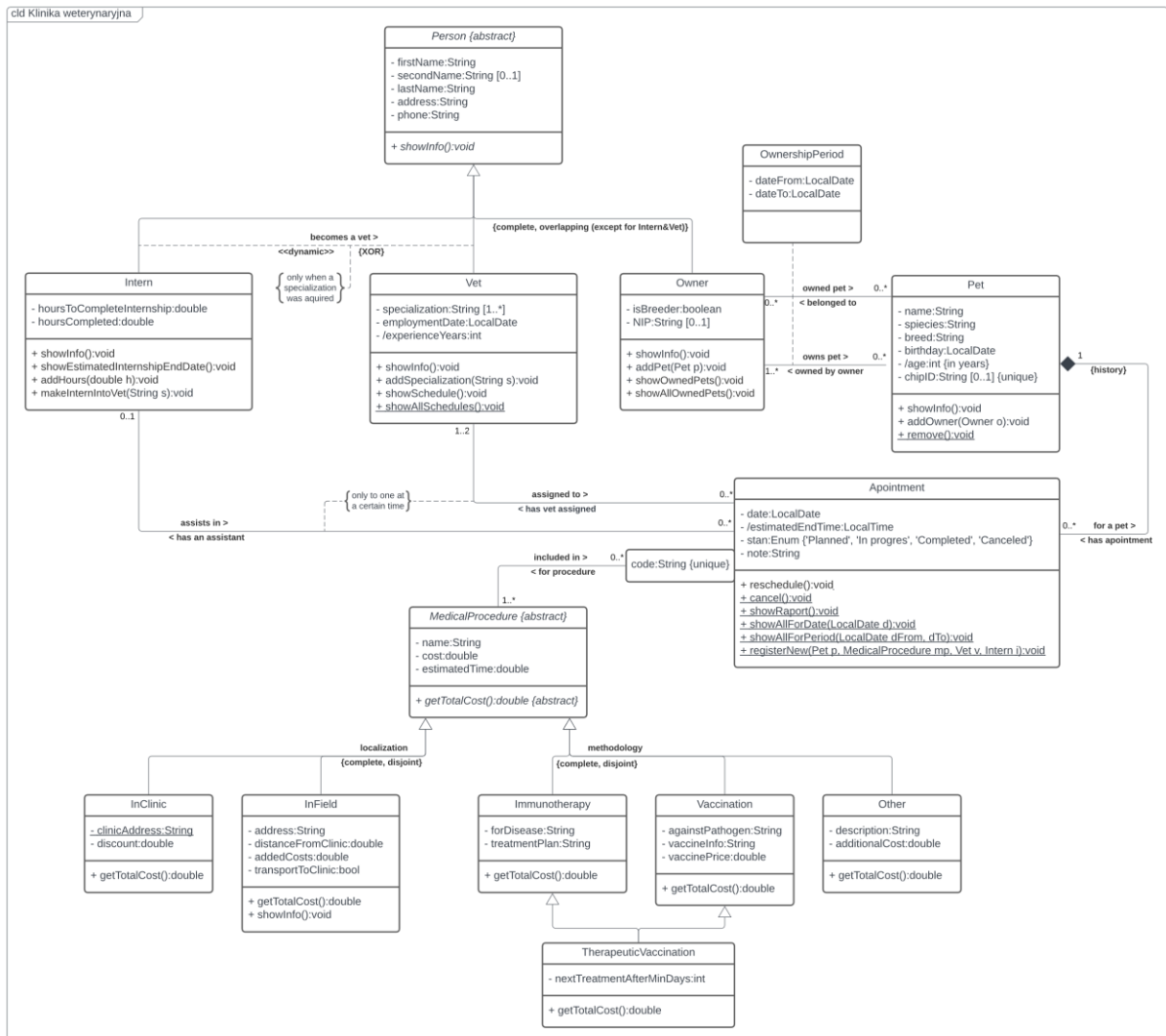
Oraz system ma zawierać następujące funkcjonalności:

1. Dla Praktykanta:
 - Sprawdzenie przebiegu praktyk.
 - Rejestracja wizyty.
 - Aktualizacja wizyty.
 - Sprawdzenie informacji o wizytach na dany dzień.
 - Rejestracja właściciela.
 - Przypisanie zwierzęcia do właściciela.
 - Wyświetlenie listy aktualnych lub wszystkich zwierząt właściciela.
 - Wyświetlenie informacji o danych osobach.
 - Rejestracja zwierzęcia.
2. Dodatkowo dla Weterynarza:
 - Zalogowanie godzin pracy dla danego praktykanta.
 - Anulacja wizyty.
 - Usunięcie informacji zwierzęcia z systemu (z opcją usunięcia jego właściciela?).
 - Rejestracja zabiegów.
 - Wyświetlenie własnego grafiku.
3. Oraz dodatkowo dla Właściciela Kliniki:
 - Wyświetlenie grafiku wszystkich weterynarzy.
 - Wyświetlenie raportu o wizytach w podanym okresie czasowym.
 - Dodanie nowego / edycję weterynarza.
 - Dodanie nowego / edycję praktykanta.
 - Awansowanie Praktykanta na Weterynarza.

2. Diagram przypadków użycia.

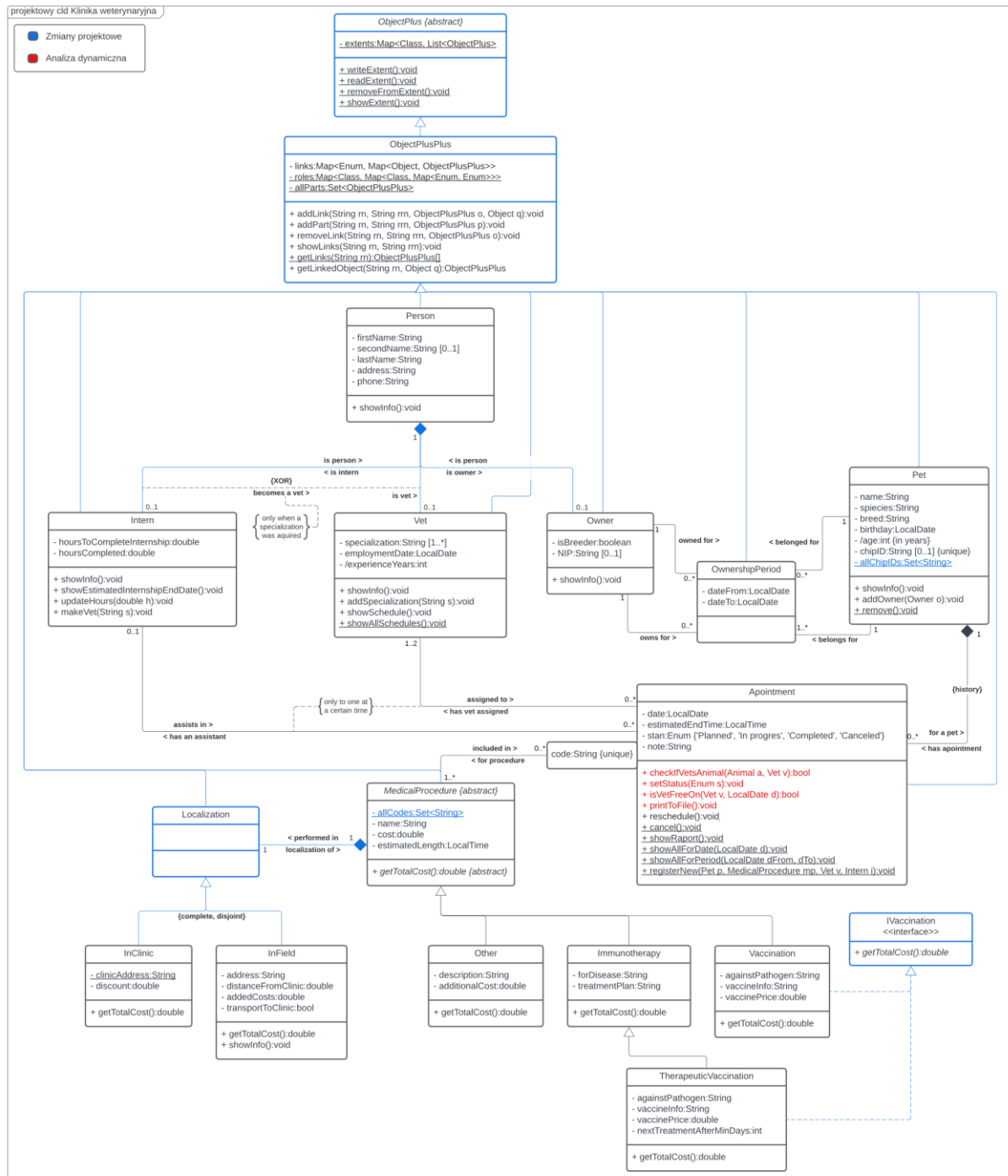


3. Diagram klas – analityczny.



*Metody takie jak settery, gettery i konstruktory oraz enumy zostały pominięte dla przejrzystości diagramów.

4. Diagram klas – projektowy.



*Metody takie jak settery, gettery i konstruktory oraz enumy zostały pominięte dla przejrzystości diagramów.

5. Scenariusz przypadku użycia.

"Rejestracja wizyty, wraz z możliwością rejestracji nowego zwierzęcia":

- **Aktorzy:** Weterynarz, System
- **Warunki wstępne:** Nowe Zwierzę; nowy lub istniejący Właściciel; chęć rejestracji Wizyty.
- **Warunki końcowe:** W systemie zostało zapisane nowe Zwierzę wraz z jego pierwszą Wizytą.
- **Rezultat:** Sprawne wprowadzenie nowego zwierzęcia i wizyty do systemu.

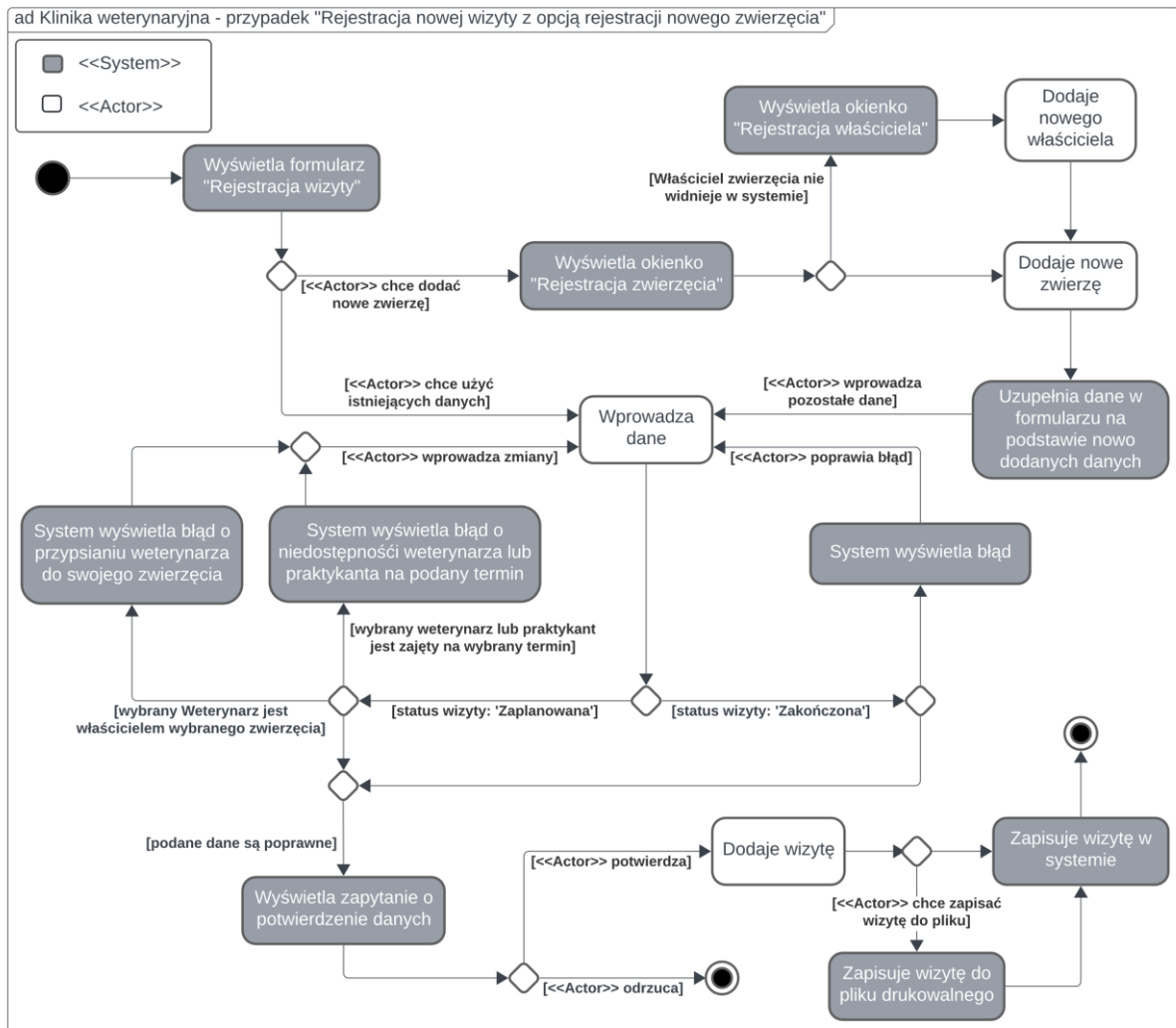
Scenariusz główny:

1. *System:* wyświetla formularz "Rejestracja wizyty".
2. *Aktor:* wybiera opcję "Dodaj nowe zwierzę".
3. *System:* wyświetla formularz "Rejestracja zwierzęcia" i listę zapisanych w systemie Właścicieli.
4. *Aktor:* wprowadza potrzebne dane do formularza.
5. *System:* uzupełnia formularz wizyty o informacje nowo dodanego zwierzęcia.
6. *Aktor:* wybiera Weterynarza odpowiedzialnego za daną wizytę i wybiera termin.
7. *System:* czeka na potwierdzenie danych formularza.
8. *Aktor:* potwierdza formularz "Rejestracja wizyty"
9. *System:* zapisuje dane, zamyka formularz "Rejestracja wizyty" i kończy przypadek użycia

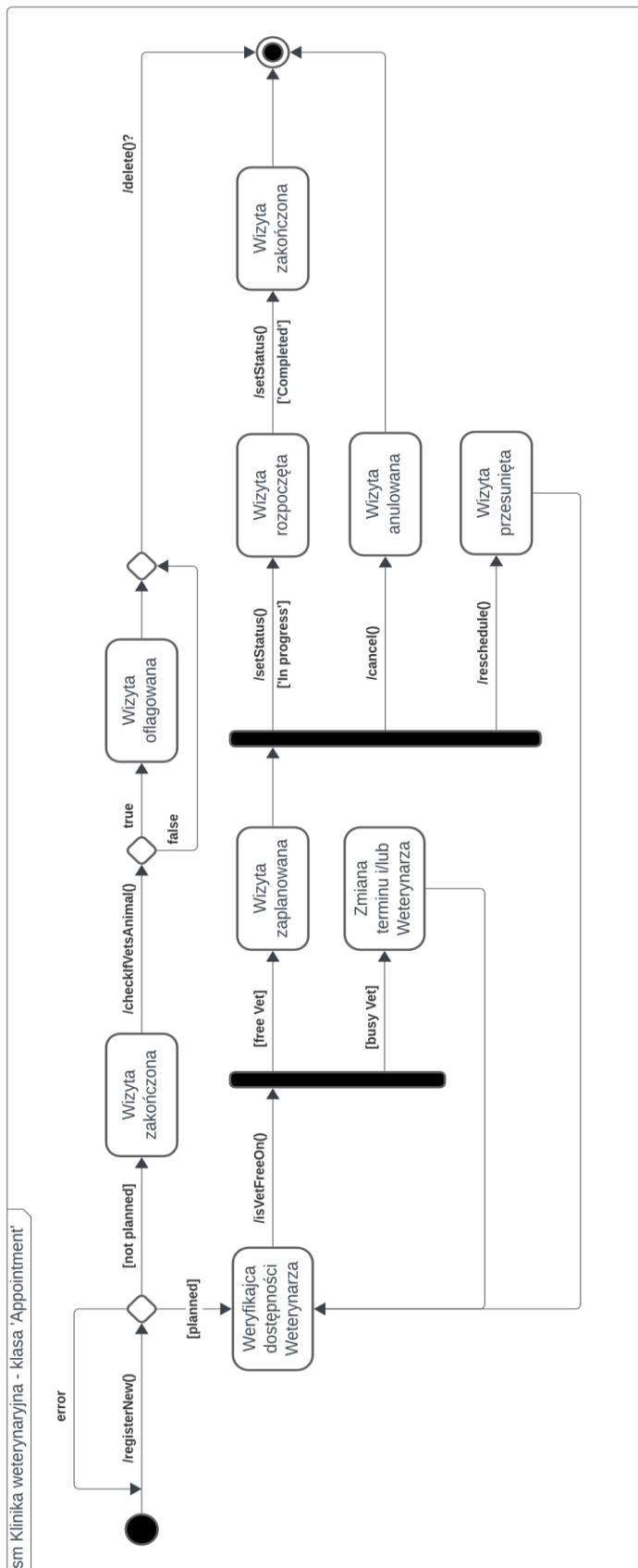
Scenariusz alternatywny:

- 3.1. *Aktor:* wybiera opcję "Rejestracja właściciela" i wprowadza dane, jeśli żaden z właścicieli zwierzęcia nie widnieje w systemie.
- 4.1. *System:* informuje o błędzie, jeżeli taki się pojawi i wraca do punktu 4.
- 5.1. *System:* uzupełnia formularz o informacje nowo dodanego właściciela.
- 6.1. *System:* informuje o błędzie, jeżeli taki się pojawi i wraca do punktu 6.
- 6.1. *System:* jeśli status wizyty został ustawiony na 'Zaplanowana' a weterynarz jest właścicielem danego zwierzęcia, system informuje o błędzie i wraca do punktu 6.
- 6.2. *System:* jeśli status wizyty został ustawiony na 'Zaplanowana' i weterynarz zajmuje się w podanym terminie inną wizytą, system informuje o błędzie i wraca do punktu 6.
- 6.3. *System:* jeśli status wizyty został ustawiony na 'Zaplanowana' i podana data jest starsza niż aktualna lub vice-versa, system informuje o błędzie i wraca do punktu 6.
- 10.1. *Aktor:* anuluje zatwierdzenie formularza i kończy przypadek użycia.
- 11.1. *System:* przygotowuje plik z informacją o utworzonej wizycie do druku.

6. Diagram aktywności dla przypadku użycia.



7. Diagram stanu dla klasy.



8. Projekt GUI.

Okno 'Formularz nowej wizyty'.

MEOW App - New Appointment

MEOW Veterinary Clinic

NEW APPOINTMENT

NAME

DATE

TIME

Kot
Pies
Żółw

PERSONEL

Weterynarz1
Weterynarz2
Praktykant

PROCEDURE

Szczepienie
Immunoterapia
Kontrola

DECLINE

CONFIRM

Okno 'Formularz nowego zwierzęcia'.

Pojawia się po interakcji z przyciskiem '+' przy sekcji 'PET' poprzedniego ekranu.

MEOW App - New Pet

NEW PET

Name

Species

Breed

ChipID

Birthday

OWNER

DECLINE

CONFIRM

Okno 'Formularz nowego właściciela'.

Pojawia się po interakcji z przyciskiem '+' przy sekcji 'OWNER' poprzedniego ekranu.

MEOW App - New Owner

NEW OWNER

Name

Second Name

Surname

Address

Phone number

NIP

Is a breeder ☐

DECLINE

CONFIRM

Okno 'Potwierdzenia'.

Pojawia się przy każdej weryfikacji opuszczania okna, zatwierdzania zmian, informacji o błędach, itp. Zamieniana jest odpowiednio wiadomość wyświetlana na środku ekranu.

Generic message. Do you wish to continue?

DECLINE

CONFIRM

9. Omówienie decyzji projektowych.

Decyzje projektowe:

- **Język programowania:** Java.
- **Do obsługi atrybutów 'nullable'** - implementacja metody create(atr...) która waliduje dane atrybutów i rzuca adekwatnym wyjątkiem lub tworzy obiekt za pomocą prywatnego konstruktora.
- **Do obsługi ograniczeń {unique}** – użycie statycznych setów wszystkich ograniczonych atrybutów w ich klasach.
- **Do obsługi ekstensji** - klasa 'ObjectPlus', która obsługuje zapis i odczyt pliku z ekstensją (plik .json), zapisywanie i wczytywanie obiektów z ekstensji, etc.
- **Do obsługi asocjacji** - klasa 'ObjectPlusPlus', która głównie obsługuje łączenie i rozłączanie asocjacji i relacji między obiektami oraz zapisy owych połączeń.
- **Do kontroli nazw asocjacji** – klasy posiadają Enumy dozwolonych pomiędzy sobą połączeń, a klasa 'ObjectPlusPlus' przechowuje mapy nazw tych połączeń
- **Do obsługi klas „części” kompozycji** – zagnieżdżenie klasy w pakiecie (package-private) z klasą „całości” w celu ograniczenia tworzenia części bez całości.
- **Zamiana dziedziczenia 'overlapping' i 'dynamic'** po klasie Osoba na kompozycję z uwagi na ograniczenia językowe. Efektem tego jest również porzucenie koncepcji abstrakcyjności dla klasy 'Person'.
- **Zamiana wielodziedziczenia** po klasach Szczepienie i Immunoterapia, na dziedziczenie tylko po klasie Immunoterapia i implementacje interfejsu Szczepienie z uwagi na ograniczenia językowe.
- **Zamiana dziedziczenia wieloaspektowego** na kompozycję z jedną klasą nadrzędną (aspektową) z uwagi na ograniczenia językowe.
- **Zamiana dziedziczenia z atrybutem** na dziedziczenie z dodatkową klasą zawierającą dany atrybut.
- **GUI** – stworzone przy pomocy biblioteki JavaFX.

Analiza dynamiczna:

- Metoda walidująca czy dany Weterynarz/Praktykant jest dostępny na wybrany termin.
- Metoda zmieniająca status Wizyty.
- Metoda przygotowująca plik do druku dla danego obiektu wizyty.

10. Słownik pojęć.

Klinika weterynaryjna – miejsce, w którym świadczone są usługi medyczne dla zwierząt.

Weterynarz – lekarz specjalizujący się w leczeniu zwierząt.

Praktykant – osoba odbywająca praktyki zawodowe (w tym przypadku - w klinice).

Szczepienie – procedura medyczna, polegająca na podaniu szczepionki, która zawiera osłabione lub martwe patogeny lub ich fragmenty, w celu wywołania odpowiedzi immunologicznej organizmu.

Immunoterapia – metoda leczenia, która polega na stymulowaniu, modulowaniu lub hamowaniu odpowiedzi immunologicznej organizmu w celu zwalczania chorób, w tym nowotworów, alergii, chorób autoimmunologicznych i infekcji.

Szczepienia terapeutyczne (immunoterapeutyczne) – specjalny rodzaj szczepień, które w przeciwieństwie do tradycyjnych szczepień profilaktycznych mają na celu stymulowanie układu odpornościowego do zwalczania istniejącej już choroby, takiej jak nowotwory czy różne przewlekłe infekcje.