|  |
| --- |
| WFiIS AGH |
| **Gabinet Weterynaryjny** |
| Dokumentacja analityczna systemu |
|  |
| Rafał Kantor, Artur Kiszka, Sylwester Tomaszewski |
| 2015-01-17 |

|  |
| --- |
| Dokument ma na celu sprecyzowanie założeń projektu, jak również ustalenie zasad jego tworzenia, jego zakresu, ograniczeń, a także wyjaśnienie szczególnych korzyści wynikających z przyjętych technik jego realizacji |

[1 Wprowadzenie 3](#_Toc259470600)

[1.1 Opis systemu 3](#_Toc259470601)

[1.2 Cele do osiągnięcia 3](#_Toc259470602)

[1.3 Słownik 3](#_Toc259470603)

[2 Wymagania funkcjonalne - przypadki użycia 3](#_Toc259470604)

[2.1 Identyfikator – nazwa przypadku użycia 3](#_Toc259470605)

[3 Wymagania niefunkcjonalne 3](#_Toc259470606)

[3.1 Wymagania względem projektu 3](#_Toc259470607)

[3.2 Reguły biznesowe 3](#_Toc259470608)

[3.3 Ograniczenia implementacji 3](#_Toc259470609)

[3.4 Dostępność, wydajność, niezawodność 3](#_Toc259470610)

[3.5 Użyteczność 3](#_Toc259470611)

[3.6 Dokumentacja 3](#_Toc259470612)

[3.7 Utrzymanie systemu 3](#_Toc259470613)

[4 Model dziedziny 3](#_Toc259470614)

[5 Modele stanów obiektów dziedziny 3](#_Toc259470615)

[6 Problemy i ryzyka 3](#_Toc259470616)

# Wprowadzenie

## Opis systemu

## Cele do osiągnięcia

Celem projektu jest stworzenie aplikacji internetowej „Gabinet Weterynaryjny”, wspierającej obsługę lecznicy dla zwierząt, system łączący zalety klasycznych, papierowych kart zawierających historię leczenia wraz z nowoczesnym, postępowym podejściem do obsługi gabinetów weterynaryjnych.

Skierowany jest on do właścicieli niewielkich (kilku osobowych) gabinetów weterynaryjnych, w których ogół procesów związanych z zarządzaniem i funkcjonowaniem gabinetu spoczywa w rękach jednego lekarza, będącego często właścicielem gabinetu. Aplikacja stanowi kompleksowy produkt, umożliwiający m.in. zarządzanie procesem pojedynczej wizyty, obsługę magazynów leków, akcesoriów medycznych oraz gadżetów dla zwierząt, jak również możliwość rezerwacji czasu pracy osób zatrudnionych w gabinecie.

Celem powstania projektu jest chęć zautomatyzowania, usprawnienia procesu kierowania lecznicą zwierząt. Obecnie czas wolny stanowiący jedno z najbardziej pożądanych dóbr w świecie biznesu jest wartością poszukiwaną również przez lekarzy weterynaryjnych. W dzisiejszych czasach, właściciel niewielkiego gabinetu, nie jest jedynie lekarzem, lecz managerem, marketingowcem, odpowiedzialnym za leczenie zwierząt oraz zapewnienie niezbędnych artykułów potrzebnych do funkcjonowania gabinetu, czy reklamowanie własnego przedsiębiorstwa.

Celem „Gabinetu Weterynaryjnego” jest maksymalne odciążenie właściciela z obowiązku analizy ogromnej ilości danych przetwarzanych wewnątrz przedsiębiorstwa. Dostarczenie mu informacji na temat najbardziej opłacalnych kierunków rozwoju gabinetu (np. najczęściej wykonywane zabiegi, najbardziej popularne wśród klientów gadżety dla zwierząt, czy najczęściej wykorzystywane leki).

## Słownik

### Aplikacja internetowa

(ang.) *web application* – zwana również aplikacją webową, to program komputerowy, który pracuje na serwerze i komunikuje się poprzez sieć komputerową z hostem użytkownika komputera z wykorzystaniem przeglądarki internetowej użytkownika, będącego w takim przypadku interaktywnym klientem aplikacji internetowej.

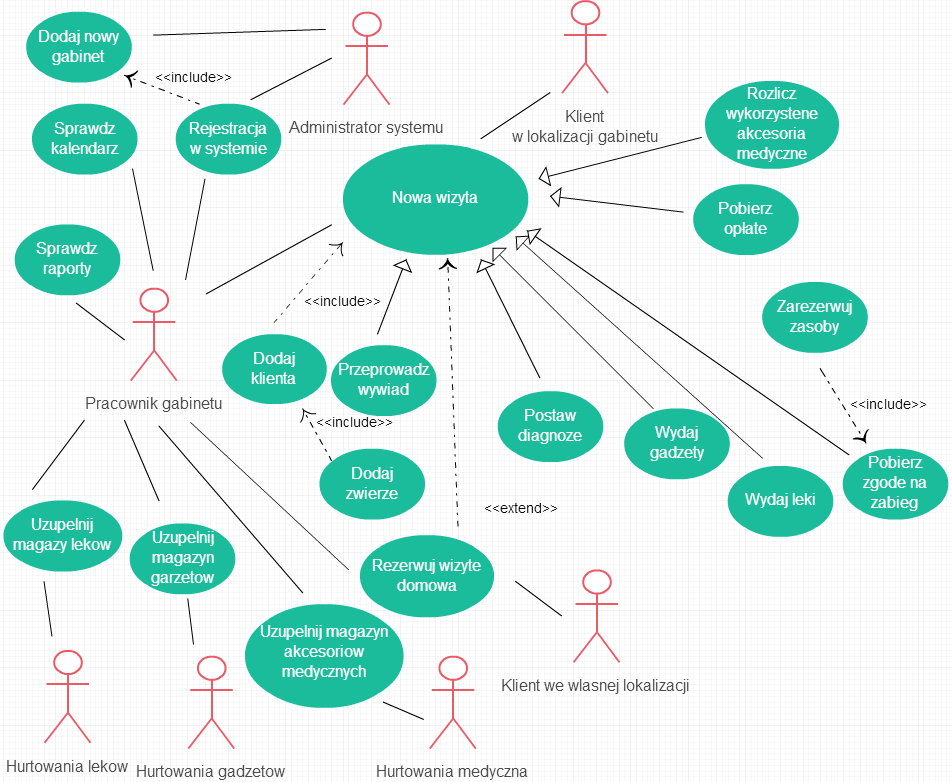
### Gadżet dla zwierząt

Przyrząd, urządzenie, czasem niepełniące funkcji użytkowej. Gadżetami mogą być smycze, piłki, ubrania, gwizdki, kosze, kuwety, szampony, karmy, obroże itp.

### Akcesoria medyczne

Ogół instrumentów medycznych wykorzystywanych w procesie diagnostyki i leczenia zwierząt. Mogą być nimi np. strzykawki, rękawice, fartuchy, prześcieradła, przyrządy USG, itp.

# Wymagania funkcjonalne - przypadki użycia



## Nowa wizyta

### Opis przypadku użycia

Nowa wizyta przedstawia jeden z podstawowych przypadków użycia aplikacji, obrazuję główną ideę funkcjonowania systemu.

### Aktorzy

Pracownik gabinetu

Klient w gabinecie weterynaryjnym

Kierownik lecznicy

### Poziom

Ogólny

### Cel

Automatyczne wypełnienie książki leczenia zwierząt przez lekarza weterynarii zarówno podczas wizyty klienta w lecznicy, jak również uzupełnienie sprawozdania z przeprowadzonej wizyty domowej.

### Cele użytkowników

1. Pracownik gabinetu: szybko i bez błędnie wypełnić książkę leczenia zwierząt określoną przez Dz.U.2011.224.1347
   1. Sporządzić w formie elektronicznej raport z przeprowadzonego wywiadu medycznego
   2. Udokumentować postawioną diagnozę medyczną
   3. Wykazać wykorzystane podczas leczenia aksonie medyczne
   4. Określić wykorzystane podczas leczenia lekarstwa
2. Klient lecznicy: otrzymać historię leczenia jego podopiecznego
3. Kierownik lecznicy: prowadzić aktualną dokumentację
4. Kierownik lecznicy: raportować stan magazynu leków i gadżetów
5. Pracownik gabinetu: szybko i skutecznie zaproponować odpowiednie lekarstwa
6. Pracownik gabinetu: zarejestrować nowego klienta w systemie bazodanowym
7. Pracownik gabinetu, Klient: zarejestrować nowe zwierzę w systemie
8. Pracownik gabinetu: uzyskać od klienta zgodę na przeprowadzenie zabiegu medycznego
9. Pracownik gabinetu: zarezerwować niezbędne do przeprowadzenia zabiegu medycznego zasobu (sale operacyjne, urządzenia medyczne)
10. Pracownik gabinetu: szybko i efektywnie zaksięgować fakt sprzedaży gadżetów
11. Pracownik gabinetu: zaraportować pobranie opłaty za leczenie

### Warunki początkowe

1. Rozpoczęcie wizyty weterynaryjnej wewnątrz gabinetu weterynaryjnego
2. Dokonanie wizyty domowej

### Częstotliwość wykonania

Kilkanaście razy dziennie

### Wyzwalacz

Lekarz wybiera akcję *Nowa wizyta* w dowolnym momencie pracy z aplikacją

### Warunki końcowe

Zakończona wizyta weterynaryjna.

### Scenariusz główny

1. System automatycznie uzupełnia nazwę i adres zakładu leczenia zwierząt.
2. System samoczynnie uzupełnia datę, godzinę a także numer przeprowadzanej wizyty.
3. System przeprowadza analizę stanu magazynu leków, gadżetów, akcesoriów medycznych.
4. Lekarz wybiera z rozwijanej listy imię, nazwisko oraz adres klienta zakładu weterynaryjnego.
5. Weterynarz uzupełnia formularz o dane zwierzęcia przypisanego do klienta, które zostanie poddane badaniu klikając z rozwijanej listy imię i rasę zwierzęcia.
6. Lekarz przeprowadza wywiad medyczny, uzupełnia spostrzeżenia w odpowiednim formularzu.
7. Pracownik gabinetu uzupełnia pole diagnoza w formularzu leczenia zwierząt.
8. System uzupełnia formularz leczenia o ilość i nazwę wykorzystanych leków, gadżetów.
9. Weterynarz uzupełnia podformularz wykorzystanych akcesoriów medycznych wykorzystanych podczas wizyty. Ich ilość i rodzaj.
10. System zmniejsza ilość dostępnych akcesoriów medycznych w magazynie gabinetu.
11. Lekarz na prośbę klienta dokonuje wydruku sprawozdania z przeprowadzonej wizyty.
12. Lekarz pobiera opłatę za usługę weterynaryjną.
13. System umieszcza w bazie informację o zainkasowaniu gotówki.

### Scenariusze alternatywne

4a. Nazwisko interesanta nie pojawia się na rozwijanej liście. Lekarz wybiera opcję dodaj klienta:

4a1. System wywołuje przypadek użycia: *Dodaj klienta.* Dalej wszystko przebiega jak w głównym scenariuszu.

4b. Nazwisko interesanta nie pojawia się na rozwijanej liście. Lekarz nie jest w stanie ustalić właściciela zwierzęcia. Wybiera opcję *Nagły* przypadek

4a1. System wywołuje przypadek użycia: *Nagły przypadek.* Dalej wszystko przebiega jak w scenariuszu *Nagły przypadek*.

### Scenariusze alternatywne

5a. Imię zwierzęcia nie pojawia się na rozwijanej liście. Klient pragnie zarejestrować nowe zwierzę. Lekarz wybiera opcję dodaj nowego zwierzę.

5a1. Aplikacja wyświetla wszystkie zwierzęta przypisane do klienta.

5a2. Lekarz przeprowadza wywiad z klientem, weryfikuje słuszność przypisania zwierząt do interesanta.

5a2.a Klient informuje lekarza o śmierci innego zwierzęcia niż to, z którym zjawił się dzisiaj w lecznicy.

4a2.b Lekarz wybiera opcję Usuń zwierzę.

4a2.c System wywołuje przypadek: Usuń zwierzę.

4a2. System wywołuje przypadek użycia: *Dodaj zwierzę.* Dalej wszystko przebiega jak w głównym scenariuszu.

### Scenariusze alternatywne

7a. Lekarz stwierdza konieczność przeprowadzenia zabiegu medycznego. Lekarz wybiera opcję Terminarz

7a1. System wywołuje przypadek użycia: *Terminarz*

7a2. Lekarz informuje klienta o terminie przeprowadzenia zabiegu.

7b. Lekarz stwierdza konieczność przepisania lekarstw zwierzęciu. Lekarz wybiera opcję Dodaj lekarstwa.

7b1. System wywołuje przypadek użycia: *Dodaj lekarstwa* Dalej wszystko przebiega jak w głównym scenariuszu.

7c. Lekarz stwierdza chęć nabycia przez klienta gadżetów . Lekarz wybiera opcję Dodaj gadżet.

7b1. System wywołuje przypadek użycia: *Dodaj gadżety* Dalej wszystko przebiega jak w głównym scenariuszu.

### Powiązane reguły biznesowe

-

### Rozszerzane przypadki użycia

-

### Włączone przypadki użycia

1. *Dodaj klienta*
2. *Dodaj zwierzę*
3. Dodaj lekarstwa
4. *Dodaj gadżety*
5. *Nagły przypadek*
6. Terminarz

### Problemy

Może zajść potrzeba przeprowadzenia wizyty w odniesieniu do dzikiego zwierzęcia (np. sanitarne uśpienie, odłów zwierząt w tym egzotycznych), w stosunku do których pewne informacje nie mogą zostać zgromadzone

### Decyzje

1. Klient gabinetu musi zostać zarejestrowany. Wiąże się to z koniecznością rejestracji użytkownika NN, dla którego statystyki mogą być trudne w interpretacji.
2. Pracownik gabinetu nie ma możliwości edycji wprowadzonych danych po ich zapisaniu. Procesu tego może dokonać jedynie właściciel lecznicy.

## Dodaj klienta

### Opis przypadku użycia

Opisany poniżej przypadek użycia obrazuje zarządzanie przedsiębiorstwem w chwili pojawienia się w nim nowego, niezarejestrowanego wcześniej klienta.

### Aktorzy

Pracownik gabinetu

### Poziom

Podfunkcji

### Cel

Zarejestrowanie nowego klienta gabinetu weterynaryjnego. Uzupełnienie i przechowywanie jego danych personalnych, a co za tym idzie umożliwienie nawiązania kontaktu z interesantem w przyszłości.

### Cele użytkowników

1. Pracownik gabinetu: szybko i bez błędnie zarejestrować nowego klienta gabinetu
2. Pracowni gabinetu: weryfikować właścicieli zwierząt
3. Kierownik lecznicy: prowadzić aktualną dokumentację klientów gabinetu

### Warunki początkowe

1. Rozpoczęcie przez klienta pierwszej wizyty weterynaryjnej w określonym gabinecie

### Częstotliwość wykonania

Początkowo kilkanaście razy dziennie, w miarę rozwoju przedsiębiorstwa kilka razy tygodniowo.

### Wyzwalacz

Lekarz wybiera akcję *Dodaj klienta* podczas obsługi *Nowej wizyty*

### Warunki końcowe

Nowy klient został zarejestrowany.

### Scenariusz główny

1. Lekarz wypełnia formularz rejestracji nowego klienta, uzupełnia pola określające imię, nazwisko, adres zamieszkania, kod pocztowy oraz miejscowość nowego interesanta.
2. System wprowadza nowego klienta do bazy.

### Scenariusze alternatywne

1a. Dane interesanta nie mogą zostać zweryfikowane . Lekarz wybiera opcję *Nagły przypadek*:

1a1. System rejestruje nowego klienta*.* Dalej wszystko przebiega jak w głównym scenariuszu.

### Powiązane reguły biznesowe

Prowadzenie statystyk m.in. najaktywniejsi klienci, miesiące w których liczba wizyt byłą największa, pracownicy zapewniający największe zyski dla gabinetu.

### Rozszerzane przypadki użycia

Nowa wizyta

### Włączone przypadki użycia

-

### Problemy

-

### Decyzje

1. Klient gabinetu musi zostać zarejestrowany. Wiąże się to z koniecznością rejestracji użytkownika NN, dla którego statystyki mogą być trudne w interpretacji.

## Dodaj zwierzę

### Opis przypadku użycia

Opisany poniżej przypadek użycia obrazuje zachowanie systemu w momencie zgłoszenia przez klienta chęci leczenia nowego zwierzęcia.

### Aktorzy

Pracownik gabinetu

### Poziom

Podfunkcji

### Cel

Zarejestrowanie nowego zwierzęcia, przypisanego do klienta gabinetu weterynaryjnego. Uzupełnienie i przechowywanie danych dotyczących imienia, rasy, wieku itp. W konsekwencji stworzenie możliwości określenia tzw. targetu określonego gabinetu weterynaryjnego.

### Cele użytkowników

1. Pracownik gabinetu: szybko i bez błędnie zarejestrować nowe zwierzę klienta gabinetu
2. Pracowni gabinetu: weryfikować posiadane przez klienta zwierzęta
3. Kierownik lecznicy: prowadzić aktualną dokumentację gatunków zwierząt obsługiwanych przez gabinet.

### Warunki początkowe

1. Rozpoczęcie przez klienta pierwszej wizyty weterynaryjnej w określonym gabinecie
2. Brak wpisów dotyczących określonego zwierzęcia w bazie danych.

### Częstotliwość wykonania

Początkowo kilkanaście razy dziennie, w miarę rozwoju przedsiębiorstwa kilka razy tygodniowo.

### Wyzwalacz

Lekarz wybiera akcję *Dodaj zwierzę*  podczas obsługi *Nowej wizyty*

### Warunki końcowe

Nowe zwierzę zostało zarejestrowane.

### Scenariusz główny

1. Lekarz wypełnia formularz rejestracji nowego zwierzęcia, uzupełnia pola określające imię, rasę oraz datę urodzenia.
2. Lekarz przypisuje zwierzę do istniejącego klienta gabinetu.
3. System wprowadza nowego klienta do bazy.
4. System wyświetla zwierzęta przypisane klientowi.

### Scenariusze alternatywne

4a1. Lekarz przeprowadza wywiad z klientem, weryfikuje słuszność przypisania zwierząt do interesanta.

4a2.a Klient informuje lekarza o śmierci innego zwierzęcia niż to, z którym zjawił się dzisiaj w lecznicy.

4a2.b Lekarz wybiera opcję Usuń zwierzę.

4a2.c System wywołuje przypadek: Usuń zwierzę.

### Powiązane reguły biznesowe

Prowadzenie statystyk określających najczęściej badanych zwierząt w określonym gabinecie weterynaryjnym.

### Rozszerzane przypadki użycia

Nowa wizyta

### Włączone przypadki użycia

-

### Problemy

-

### Decyzje

1. Zwierzę musi być jednoznacznie identyfikowane w systemie za pomocą 3 zmiennych: imię, rasę, datę urodzenia a także właściciela.

## Dodaj leki

### Opis przypadku użycia

Opisany poniżej przypadek użycia obrazuje zachowanie systemu w momencie zgłoszenia przez lekarza weterynarii konieczności leczenia z wykorzystaniem lekarstw.

### Aktorzy

Pracownik gabinetu

### Poziom

Podfunkcji

### Cel

Określenie dostępności lekarstw w magazynie gabinetu weterynaryjnego lub konieczności zamówienia ich w hurtowni.

### Cele użytkowników

1. Pracownik gabinetu: szybko i bez błędnie zarejestrować lekarstwa wykorzystane podczas leczenia zwierzęcia
2. Kierownik lecznicy: weryfikować wykorzystywane przez pracowników lekarstwa
3. Kierownik lecznicy: prowadzić aktualną dokumentację stanu magazynu leków

### Warunki początkowe

1. Stwierdzenie konieczności leczenia pacjenta z wykorzystaniem lekarstw.

### Częstotliwość wykonania

Kilkanaście razy dziennie.

### Wyzwalacz

Lekarz wybiera akcję *Dodaj lekarstwa*  podczas obsługi *Nowej wizyty*

### Warunki końcowe

Nowe leki zostały wydane.

### Scenariusz główny

1. W wyszukiwarce lekarz wpisuje nazwę leku, lub choroby, którą zdiagnozował.

2. System odpowiada, wyświetlając listę leków oraz ich etykiety.

3. System stwierdza dostępność wskazanych leków w magazynie gabinetu.

### Scenariusze alternatywne

3a. System stwierdza brak wybranych lekarstw w magazynie leków.

3a1 Lekarz wypisuje receptę.

3a1a. System informuje lekarza o konieczności kontaktu z hurtownią leków.

### Powiązane reguły biznesowe

**-**

### Rozszerzane przypadki użycia

Nowa wizyta

### Włączone przypadki użycia

-

### Problemy

1. Konieczność monitorowania najnowszych trendów medycznych i wzbogacanie bazy danych o te informacje.
2. Bardzo duży narzut pracy podczas katalogowania etykiet dostępnych farmaceutyków.

### Decyzje

-

## Dodaj gadżety

### Opis przypadku użycia

Opisany poniżej przypadek użycia obrazuje zachowanie systemu w momencie zgłoszenia przez lekarza weterynarii chęci sprzedaży gadżetów dla zwierząt.

### Aktorzy

Pracownik gabinetu

### Poziom

Podfunkcji

### Cel

Określenie dostępności gadżetów w magazynie gabinetu weterynaryjnego lub konieczności zamówienia ich w hurtowni.

### Cele użytkowników

1. Pracownik gabinetu: szybko i bez błędnie zarejestrować gadżety udostępnione klientowi gabinetu weterynaryjnego.
2. Kierownik lecznicy: weryfikować sprzedawane gadżety
3. Kierownik lecznicy: prowadzić aktualną dokumentację stanu magazynu gadżetów

### Warunki początkowe

1. Stwierdzenie chęci nabycia gadżetu przez klienta leczenia.

### Częstotliwość wykonania

Kilkanaście razy dziennie.

### Wyzwalacz

Lekarz wybiera akcję *Dodaj gadżety*  podczas obsługi *Nowej wizyty*

### Warunki końcowe

Nowe leki zostały wydane.

### Scenariusz główny

1. W wyszukiwarce lekarz wpisuje nazwę gadżetu który klient chce nabyć.

2. System odpowiada, wyświetlając listę dostępnych gadżetów.

3. System stwierdza dostępność wskazanych gadżetów w magazynie gabinetu.

### Scenariusze alternatywne

3a. System stwierdza brak wybranych gadżetów w magazynie.

3a. System informuje lekarza o konieczności kontaktu z hurtownią gadżetów.

### Powiązane reguły biznesowe

Prowadzenie statystyk

### Rozszerzane przypadki użycia

Nowa wizyta

### Włączone przypadki użycia

-

### Problemy

-

### Decyzje

-

## Nagły przypadek

### Opis przypadku użycia

Opisany poniżej przypadek użycia obrazuje zachowanie systemu w momencie zgłoszenia przez lekarza weterynarii chęci przeprowadzania nowej wizyty w stosunku do klienta którego personalia nie mogą zostać ustalone.

### Aktorzy

Pracownik gabinetu

### Poziom

Podfunkcji

### Cel

Przeprowadzenie badania zwierzęcia, którego właściciel nie może zostać ustalony.

### Cele użytkowników

1. Pracownik gabinetu: szybko i bez błędnie zarejestrować nową wizytę weterynaryjną.
2. Kierownik lecznicy: prowadzić aktualną dokumentację.

### Warunki początkowe

1. Rozpoczęta nowa wizyta

### Częstotliwość wykonania

Kilkanaście razy miesięcznie.

### Wyzwalacz

Lekarz wybiera opcję *Nagły przypadek* w polu „Imię i nazwisko” klienta, podczas obsługi *Nowej wizyty*

### Warunki końcowe

Nowa wizyta zastałą zakończona.

### Scenariusz główny

1. Lekarz stwierdziwszy brak możliwości rejestracji nowego klienta, uzupełnia jego dane wartością XX lub NN

2. System odpowiada, wyświetlając komunikat: „Nagły przypadek”. Dalej wszystko przebiega jak w scenariuszu *Nowa wizyta*.

### Scenariusze alternatywne

-

### Powiązane reguły biznesowe

-

### Rozszerzane przypadki użycia

Nowa wizyta

### Włączone przypadki użycia

-

### Problemy

-

### Decyzje

1. W przypadku konieczności leczenia pacjenta, którego właściciela pracownik lecznicy nie jest w stanie ustalić powinien skorzystać z faktora Nagły przypadek
2. Przypadek z definicji powinien zapewnić możliwość księgowania przypadków niestandardowych np. leczenie zwierząt dziko żyjących, będących ofiarami wypadków samochodowych.

## Terminarz

### Opis przypadku użycia

Opisany poniżej przypadek użycia obrazuje zachowanie systemu w momencie zgłoszenia przez lekarza weterynarii chęci zarezerwowania lub sprawdzenia dostępności szeroko rozumianych zasobów medycznych (sal operacyjnych, urządzeń medycznych).

### Aktorzy

Pracownik gabinetu

Klient gabinetu

### Poziom

Podfunkcji

### Cel

Określenie dostępności zasobów medycznych w określonym przez pracownika gabinetu terminie. Rezerwacja ich czasu pracy. Zaplanowanie wizyt domowych i w gabinecie.

### Cele użytkowników

1. Pracownik gabinetu: szybko i bez błędnie sprawdzić stan dostępnych zasobów medycznych.
2. Pracownik gabinetu: rezerwować zasoby
3. Pracownik gabinetu: rezerwować wizyty domowe
4. Pracownik gabinetu: sprawdzić zaplanowane zabiegi
5. Klient gabinetu: zapisać się na ustaloną godzinę
6. Kierownik lecznicy: weryfikować najczęściej użytkowane urządzenia i terminy zabiegów

### Warunki początkowe

1. Konieczność przeprowadzenia zabiegu medycznego.

### Częstotliwość wykonania

Kilka razy tygodniowo.

### Wyzwalacz

Lekarz wybiera akcję *Terminarz*  podczas obsługi *Nowej wizyty* lub w dowolnym innym miejscu w aplikacji

### Warunki końcowe

Dodanie wpisu w kalendarzu

### Scenariusz główny

1. System informuje lekarza o możliwości rezerwacji zasobu medycznego w określonym terminie wyświetlając „tabelę pracy lecznicy”

2.Lekarz rezerwuje czas pracy zasobu medycznego w określonym terminie.

### Scenariusze alternatywne

3a. System stwierdza brak możliwości rezerwacji zasobów.

3a1. Lekarz wykonuje akcję *Nagły przypadek*.

3a1a. System potwierdza dokonanie zasobów.

3a2 Lekarz kieruje pacjenta do innego gabinetu weterynaryjnego.

### Powiązane reguły biznesowe

-

### Rozszerzane przypadki użycia

Nowa wizyta

### Włączone przypadki użycia

-

### Problemy

-

### Decyzje

## Rejestracja w systemie

### Opis przypadku użycia

Opisany poniżej przypadek użycia obrazuje zachowanie systemu w momencie zgłoszenia przez nowego pracownika chęci rejestracji w systemie „Gabinet Weterynaryjny”

### Aktorzy

Pracownik gabinetu

Kierownik gabinetu

### Poziom

Ogólny

### Cel

Określenie przypisania nowego pracownika do gabinetu weterynaryjnego.

### Cele użytkowników

1. Pracownik gabinetu: uzyskać dostęp do możliwości systemu
2. Kierownik lecznicy: weryfikować weryfikacji jakości pracy pracownika

### Warunki początkowe

1. Zatrudnienie nowego pracownika

### Częstotliwość wykonania

Kilka razy.

### Wyzwalacz

Kierownik lecznicy wybiera akcję *Zarejestruj w systemie* w dowolnym momencie pracy z aplikacją.

### Warunki końcowe

Pracownik został zarejestrowany.

### Scenariusz główny

1. Nowy użytkownik wypełnia formularz rejestracji.
2. System wysyła formularz do Kierownika lecznicy.
3. Kierownik lecznicy potwierdza dane rejestrowanego pracownika.
4. System odbiera wiadomość potwierdzającą, generuje link rejestracyjny.
5. Aplikacja przesyła link aktywacyjny za pośrednictwem wiadomości email do Pracownika Gabinetu.
6. Pracownik otrzymuje email potwierdzający rejestrację.
7. Pracownik gabinetu potwierdza rejestracje klikając w link rejestracyjny.

### Scenariusze alternatywne

4a. Po upływie 24 godzin od przesłania formularza rejestracji, aplikacja nie otrzymuje potwierdzenia, system wysyła wiadomość sms z prośbę o weryfikację pracownika.

4a1. Lekarz potwierdza rejestrację pracownika

4a1. Lekarz odrzuca zaproszenie, uznając wiadomość jako niezaadresowaną do niego

### Powiązane reguły biznesowe

**-**

### Rozszerzane przypadki użycia

-

### Włączone przypadki użycia

-

### Problemy

-

### Decyzje

Każdy użytkownik systemu za wyjątkiem administratora i Kierownika Gabinetu musi przejść proces weryfikacji.

## Uzupełnij magazyn leków

### Opis przypadku użycia

Opisany poniżej przypadek użycia obrazuje zachowanie systemu w momencie zgłoszenia przez pracownika chęci sprawdzenia oraz uzupełnienia magazynu leków.

### Aktorzy

Pracownik gabinetu

### Poziom

Ogólny

### Cel

Weryfikacja stanu magazynu leków

### Cele użytkowników

1. Pracownik gabinetu: zweryfikować stan magazynu leków
2. Pracownik gabinetu: uzupełnić stan magazynu leków
3. Kierownik lecznicy: udokumentować zasoby magazynu leków.

### Warunki początkowe

1. Subiektywne odczucie pracownika

### Częstotliwość wykonania

Niemożliwa do określenia

### Wyzwalacz

Pracownik lecznicy wybiera akcję *Uzupełnij magazyn leków.*

### Warunki końcowe

1. Pracownik został poinformowany o stanie magazynu.
2. Pracownik zamówił lekarstwa w hurtowni

### Scenariusz główny

1. System automatycznie wyświetla informację na temat lekarstw zarejestrowanych w systemie, których ilość opakowań jest mniejsza niż 20szt.
2. System wysyła wiadomość email do pracowników lecznicy z informacją na temat lekarstw których ilość jest mniejsza niż 10szt.
3. Pracownik lecznicy wpisuję nazwę leku, jaką chciałby zamówić w hurtowni.

# Wymagania niefunkcjonalne

Tworzona w ramach projektu aplikacja ma oferować komfortowy styl pracy, który będzie spełniał oczekiwania tzw. użytkownika casualowego.

Interfejs musi być intuicyjny i prosty w obsłudze, a sama obsługa aplikacji nie powinna wymagać od klienta żadnej wiedzy na temat pracy aplikacji internetowych.

Wymagania sprzętowe muszą być bardzo niskie, a program powinien oferować maksymalną przenośność – możliwość uruchomienia na komputerach różnego typu i pod dowolnymi systemami operacyjnymi.

## Wymagania względem projektu

Planowana funkcjonalność tworzonej w ramach projektu aplikacji kształtuje się następująco:

– prosta oprawa graficzna,

– praca w czasie rzeczywistym

Tworzony system narzuca na programistów następujące wymagania:

1. Praca w środowisku programistycznym NetBeans
2. Wykorzystanie Java 1.8 oraz Java EE SDK 1.7
3. Praca z wykorzystaniem MySQL server (dodatek workbench 6.1)
4. Umiejętność posługiwania się standardami JPA, JSF,
5. Konieczność posiadanie wiedzy z zakresu obsługi systemu kontroli wersji git

## Ograniczenia implementacji

Internet pierwotnie był wykorzystywany w celach naukowych. W miarę upływu czasu okazał się on jednak, świetnym narzędziem komunikacji, pracy a także rozrywki. Wykształciło się wiele narzędzi ułatwiających pracę z globalną siecią, ciągle wzrastała, więc liczba jej użytkowników. Internet zaczął odgrywać bardzo ważną rolę w życiu codziennym każdego z nas. Nowoczesne aplikacje internetowe nie są już prostymi stronami internetowymi, mającymi na celu zastąpić ogłoszenia prasowe, a skomplikowanymi strukturami, charakteryzującymi się rozbudowaną interakcją z użytkownikiem, przyjaznym interfejsem oraz współpracą z systemami bazodanowymi umożliwiającymi trwałe przechowywanie bardzo dużych ilości informacji (np. wartości liczbowych, łańcuchów znaków, plików filmowych czy graficznych) oraz bardzo szybkie operowanie na tych danych (wyszukiwanie, sortowanie, dodawanie i usuwanie, a także edycję).

Zadaniem programisty, jako osoby tworzącej aplikacje jest jak najbardziej efektywne połączenie ww. warstw. Tworzona przez niego aplikacja podczas próby przetwarzania informacji każdorazowo wykonuje szereg instrukcji, który możemy podzielić na kilka podstawowych etapów:

* nawiązanie połączenia z bazą danych,
* wysłanie zapytania,
* odebranie wyniku,
* zamknięcie połączenia z bazą danych.

Kilka lat temu zadanie to realizowane było to w języku Java wyłącznie za pomocą JDBC. Obecnie wykorzystywana jest w tym celu technologia ORM (odpowiedzialna za odwzorowania obiektowej architektury systemu informatycznego na bazę danych o relacyjnym charakterze), która zyskiwała na popularności wraz z rozwojem programowania obiektowego. W związku ze wzrostem złożoności obiektów, wzrósł również stopień skomplikowania, a także czas wykonywania zapytań kierowanych do systemu zarządzania bazą danych. Relacyjne bazy danych umożliwiają operowanie jedynie na płaskich, jednowymiarowych modelach danych. Podejście takie wydaje się być rozwiązaniem nienaturalnym, sprawiającym trudności w projektowaniu i utrzymywaniu systemów. Dlatego też zaistniała potrzeba stworzenia mechanizmu umożliwiającego zapis obiektów w relacyjnym modelu bazy danych. Rozwiązaniem okazało się zastosowanie narzędzi ORM (rozwiązanie to pozostaje odporne na niektóre błędy generowane przez programistów oraz umożliwia osiągnąć lepszą wydajność m.in. poprzez wyeliminowanie wielokrotnego pisania złożonych zapytań). Twórcy systemu „Gabinet Weterynaryjny” zdecydowali się na wykorzystanie standardu JPA w języka Java (implementacji standardu dostarcza biblioteka Hibernate). Powodem takiej decyzji była chęć ograniczenia czasu potrzebnego do stworzenia systemu. Interfejs użytkownika wymagał wykorzystania serwera aplikacyjnego, w tym celu wykorzystany został serwer GlassFish, który umożliwił wykorzystanie JSF. Do przechowywania danych użyta została baza MySQL.

## Dostępność, wydajność, niezawodność

## Użyteczność

System skierowany jest do właścicieli kilku osobowych gabinetów weterynaryjnych. W których szereg procesów związanych z zarządzaniem i funkcjonowaniem gabinetu spoczywa w rękach jednego lekarza. Aplikacja stanowi kompleksowy produkt, umożliwiający m.in. zarządzanie procesem pojedynczej wizyty, obsługę magazynów leków, akcesoriów medycznych oraz gadżetów dla zwierząt, jak również możliwość rezerwacji czasu pracy osób zatrudnionych w gabinecie.

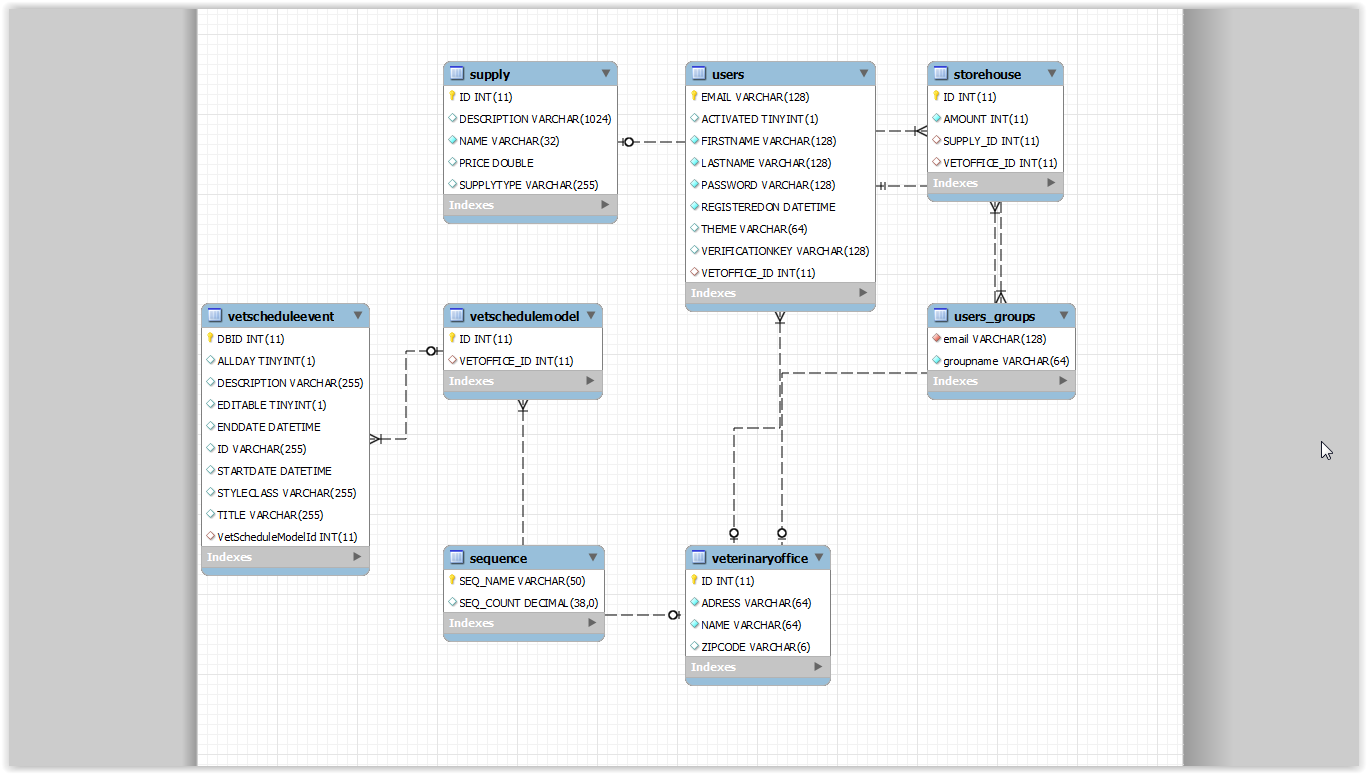
## Dokumentacja

1. Twórcy systemu zalecają wykorzystanie narzędzia JavaDoc w procesie tworzenia dokumentacji kodu aplikacji
2. Przyjęte założenia:
   1. Wszelkie nazwy wykorzystywane w aplikacji powinny być dobrane tak aby odkrycie ich przeznaczenia było intuicyjne
   2. Narzuconym językiem roboczym jest język angielski
   3. zmienne nazywamy małą literą, przy wielowyrazowych nazwach zmiennych używamy wielkiej litery do rozdzielania wyrazów, np. clientName,
   4. jeżeli intuicyjność nazwy nie jest osiągnięta w 100% to komentujemy jej pierwsze pojawienie się w kodzie
   5. klasy nazywamy wielką literą
   6. klasy opisujemy krótkim komentarzem (cel powstania klasy)
   7. w ciele klasy kolejno: pola, następnie konstruktor(y), na końcu implementujemy funkcje
   8. metody nazywamy małą literą, używamy zapisu w konwencji camelCase
   9. funkcje opisujemy  w komentarzu (cel jej przeznaczenia, argumenty funkcji)
   10. do każdej funkcji należy napisać równolegle do jej tworzenia (lub zaraz po jej zaimplementowaniu) kilka testów jednostkowych, które będą wyczerpywać możliwości wystąpienia błędu w działaniu funkcji
   11. nazwa testu musi jednoznacznie wskazywać na funkcję jaką testuj

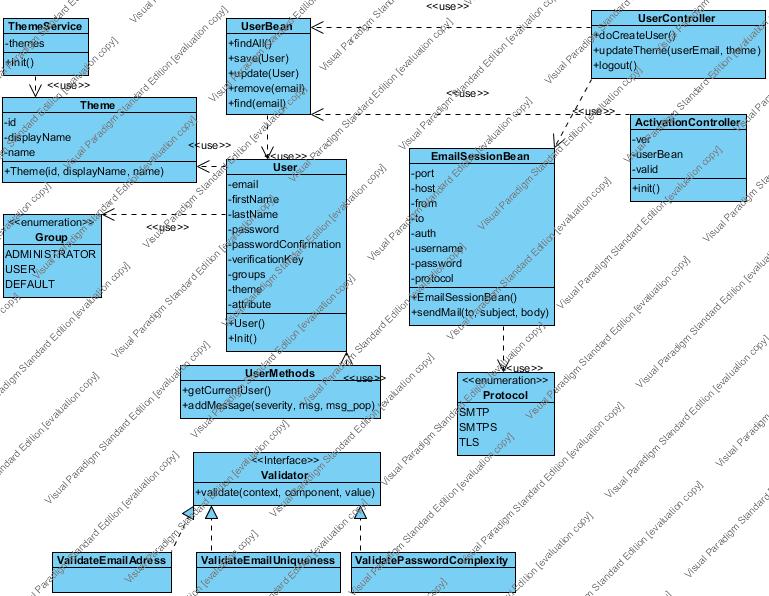
## Utrzymanie systemu

System „Gabinet Weterynaryjny” zakłada istnienie jednego, centralnego serwera aplikacyjnego. Twórcy nie zakładają możliwości wdrażanie systemu lokalnie na komputerach klientów. Cały proces umożliwienia klientowi korzystania z aplikacji opiera się na utworzeniu mu odpowiedniego konta w systemie i przydzielenie adresu strony www, na której będzie znajdował się interfejs użytkownika. System z założenie musi stwarzać możliwość zarządzania i utrzymania przez administratorów z dowolnej lokalizacji.

# Model dziedziny



Struktura bazy danych



Opis klas:

1. User – klasa encji pracownika gabinetu weterynaryjnego. Przechowuje informacje na temat jego personaliów a także, hasła, poziom uprawnień.
2. SupplyConverter – klasa konwertera umożliwiająca rzutowanie obiektów klas DRUG, GADGET oraz MEDICALSUPPLY na typ String
3. VetOfficeConverter – klasa konwertera umożliwiająca rzutowanie obiektów klasy VeterinaryOffice na typ String
4. Group – typ wyliczeniowy określający uprawnienia użytkowników
5. SupplyType - typ wyliczeniowy określający uprawnienia rodzaj asortymentu
6. Theme – klasa odpowiedzialna za przechowywanie informacji na temat wyglądu interfajsu użytkownika
7. EmailSessionBean – klasa ziarna odpowiadająca za funkcjonalność wysyłania wiadomości email
8. StoreHouseBean– klasa ziarna reprezentująca magazyn leków, gadżetów i asortymentów medycznych
9. SupplyBean – ziarno umożliwiające przeprowadzanie operację bazodanowe na obiektach znajdujących się w magazynie
10. UsearBean – odpowiedzialna za operacje na obiektach User
11. VetScheduleEventBean – ziarno odpowiedzialne za przechowywanie stanu wydarzenia jakie pragniemy zarezerwować w kalendarzu
12. VetScheduleModelBean – klasa odpowiedzialna za powiązanie klas VetScheduleEvent oraz VetSchedule
13. VeterinaryOfficeBean – klasa odpowiedzialna za operacje dodawania, usuwania i edycji gabinetów weterynaryjnych
14. StoreHouse – encja magazynu
15. Supply – encja zasobów magazynu
16. VetScheduleEvent – klasa encji wydarzeń znajdujących się w terminarzu
17. VetScheduleModel – klasa encji terminarza
18. ThemeService – klasa odpowiedzialna za przetwarzanie informacji na temat wyglądu interfejsu użytkownika
19. EmailAdressValidator, EmailUniquenessValidator, PasswordComplexityValidator, SupplyUniquenessValidator – klasy walidatorów odpowiedzialnych za kontrolę poprawności wprowadzanych do bazy obiektów.

# Modele stanów obiektów dziedziny

-

# Problemy i ryzyka

1. Produkt może nie przyjąć się na rynku.
2. Istnieje konkurencja : możliwość rozwoju oferty przez konkurencje.
3. Produkt celuje w małe gabinety weterynaryjne : możliwość że specyfika pracy w każdym z nich jest na tyle różnorodna że ciężko będzie wdrożyć produkt bez istotnych modyfikacji
4. Ograniczony czas na development może spowodować to że aplikacja będzie trudna w utrzymaniu.
5. Wiąże się z poprzednim punktem : niektóre sytuacje mogą być trudne do przeprocesowania za pomocą standardowych mechanizmów zaimplementowanych w sytuacji; potrzeba przewidzenia obsługi przypadków nietypowych.