

Yehor Voiko
S18917 gr 24c

Randki dla zwierząt ***„Piesikot”***

Dokumentacja do projektu UML z przedmiotu MAS

Spis treści

DZIEDZINA PROBLEMOWA	3
CEL.....	4
ZAKRES ODPOWIEDZIALNOŚCI SYSTEMU	5
UŻYTKOWNICY SYSTEMU	6
WYMAGANIA UŻYTKOWNIKA	7
KLIENT	7
ZWIERZ.....	7
WETERYNARZ	7
SPRZEDAWCA ZWIERZĄT.....	7
MODERATOR.....	8
WERYFIKATOR.....	8
DIAGRAM KLAS – ANALITYCZNY	10
DIAGRAM KLAS – PROJEKTOWY	11
OPIS METOD PRZEDSTAWIONYCH NA DIAGRAMACH KLAS	12
WYMAGANIA NIEFUNKCJONALNE.....	13
OPIS PRZYSZŁEJ EWOLUCJI SYSTEMU	14
SŁOWNIK	15
SZCZEGÓŁOWY OPIS PROJEKTU.....	16
SYSTEM MA UMOŻLIWIAĆ REALIZACJĘ PONIŻSZEJ FUNKCJONALNOŚCI:...	17
SCENARIUSZ DLA PRZYPADKU UŻYCIA DOSTANIE STATUSU WETERYNARZA:.....	18
DIAGRAM AKTYWNOŚCI DLA PRZYPADKU UŻYCIA DOSTANIE STATUSU WETERYNARZA:.....	19
DIAGRAM STANU DLA KLASY KLIENT:	20
DIAGRAM INTERAKCJI DLA PRZYPADKU UŻYCIA DOSTANIE STATUSU WETERYNARZA:.....	21
PROJEKT GUI (MODERATORA):.....	22
OMÓWIENIE DECYZJI PROJEKTOWYCH:.....	24
WYBÓR SPOSOBU ZAPEWNIENIA TRWAŁOŚCI.....	24
OMÓWIENIE DECYZJI PODJĘTYCH PODCZAS ANALIZY DYNAMICZNEJ	24

DZIEDZINA PROBLEMOWA

Ze względu na to, że w nasz czas ludzie są całkowicie zajęci pracą oraz sprawami gospodarczymi, „Piesikot” może pomóc nie tylko oszczędzić czas, ale także pomóc im być bardziej produktywnymi, i troskować o życiu prywatnym swoich zwierząt.

CEL

„Piesikot” - to uniwersalny serwis dla szukania radości, zdrowia i miłości dla twojego zwierzaka.

Ma on ułatwić komunikację pomiędzy ludźmi ze wspólnymi zainteresowaniami i problemami, ma również pomóc w rozwoju klinik weterynaryjnych.

ZAKRES ODPOWIEDZIALNOŚCI SYSTEMU

Klient może kupić lub sprzedać zwierza, a także może znaleźć najbliższego weterynarza dla swojego zwierzątka. Być może, nawet sam zarejestrować się jako weterynarz.

System powinien sprawnie przeprowadzać proces sprzedaży i kupowania zwierząt, rejestracji klientów i dostawiania przez nich pewnych statusów. W jego skład ma wchodzić baza danych i powiązana z nią aplikacja. Baza danych powinna przechowywać w sobie wszystkie informacje o klientach wraz z ich zwierzami i wszystkich pracownikach. Aplikacja natomiast ma oferować możliwość przeglądania i zapisywania odpowiednich danych dla poszczególnych aktorów.

UŻYTKOWNICY SYSTEMU

- ❑ Klient
- ❑ Weterynarz
- ❑ Sprzedawca Zwierząt
- ❑ Właściciel
- ❑ Moderator
- ❑ Weryfikator

WYMAGANIA UŻYTKOWNIKA

System do obsługi prywatnej kliniki stomatologicznej powinien przechowywać informacje dotyczące:

- Klient
- Weterynarz
- Sprzedawca
- Moderator
- Weryfikator
- Zwierz

KLIENT

W danych dotyczących klientów powinny znajdować się informacje:

- ID Klienta {unikatowe}
- Dane osobowe
- Dane geolokacji
- Zdjęcie [0..*]
- Informacja o sobie {do 200 znaków}
- Zwierze klienta [0..*]

ZWIERZ

W danych dotyczących zwierząt powinny znajdować się informacje:

- ID Zwierza {unikatowe w ramach konta właściciela}
- Imię
- Płeć
- Rasa
- Typ
- Rozmiar
- Status [„sprzedaje się”, „szuka miłości”, brak statusu, „usunięty”]
- Wypełniony formularz {jeśli zwierz ma status „sprzedaje się”}

WETERYNARZ

W danych dotyczących weterynarzy powinny znajdować się informacje:

- ID Klienta {unikatowe}
- Dane osobowe
- Dane geolokacji
- Zdjęcie [0..*]
- Informacja o sobie {do 200 znaków}
- Zwierze klienta [0..*]
- Dokumenty potwierdzające umiejętności [1..*]

SPRZEDAWCA ZWIERZĄT

W danych dotyczących sprzedawców powinny znajdować się informacje:

- ID Klienta {unikatowe}
- Dane osobowe
- Dane geolokacji
- Zdjęcie [0..*]
- Informacja o sobie {do 200 znaków}
- Zwierze klienta [1..*]

MODERATOR

W danych dotyczących recepcjonistów powinny znajdować się informacje:

- ID Pracownika {unikatowe}
- Dane osobowe
- Minimalna stawka za godzinę pracy = 15 zł
- Stawka za godzinę pracy {>= Minimalnej stawce}
- Poziom wiedzy [0..10]

WERYFIKATOR

W danych dotyczących zabiegów powinny znajdować się informacje:

- ID Pracownika {unikatowe}
- Dane osobowe
- Minimalna stawka za godzinę pracy = 15 zł
- Stawka za godzinę pracy {>=Minimalnej stawce}
- Ilość weryfikacji

WYMAGANIA FUNKCYJALNE

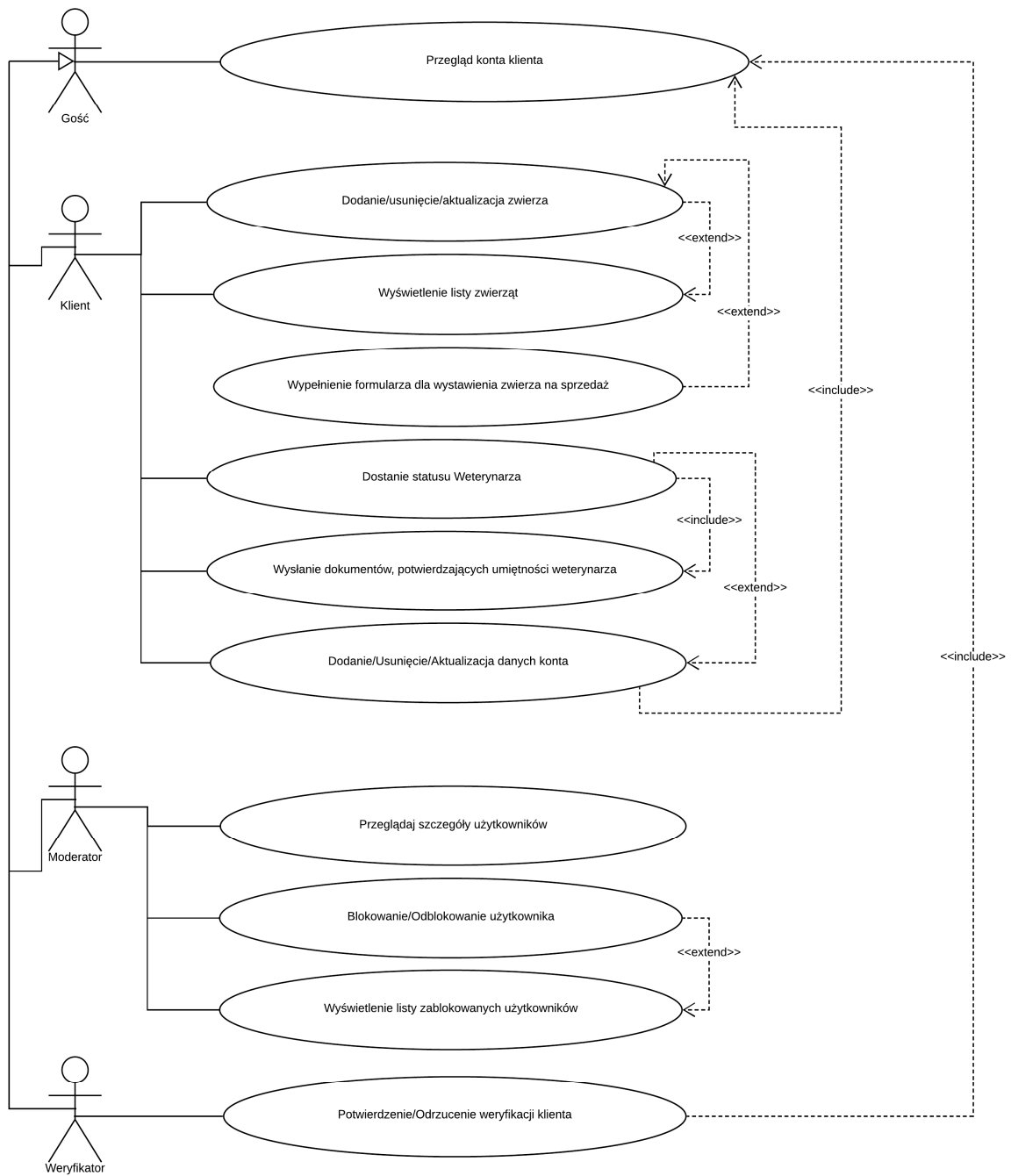


DIAGRAM KLAS – ANALITYCZNY

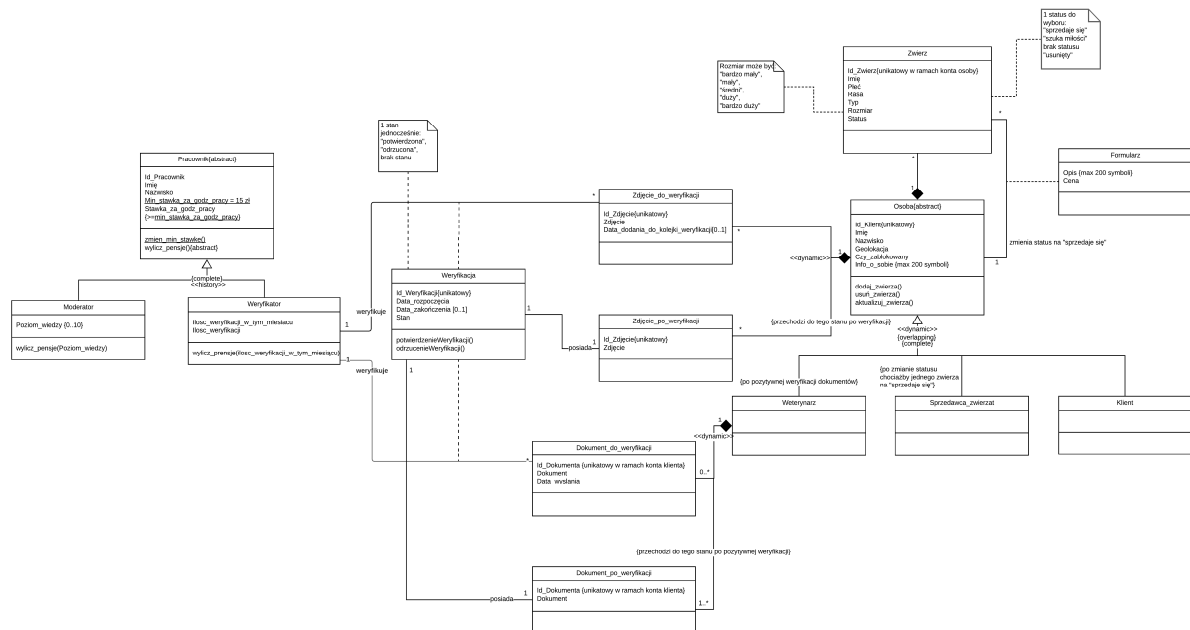
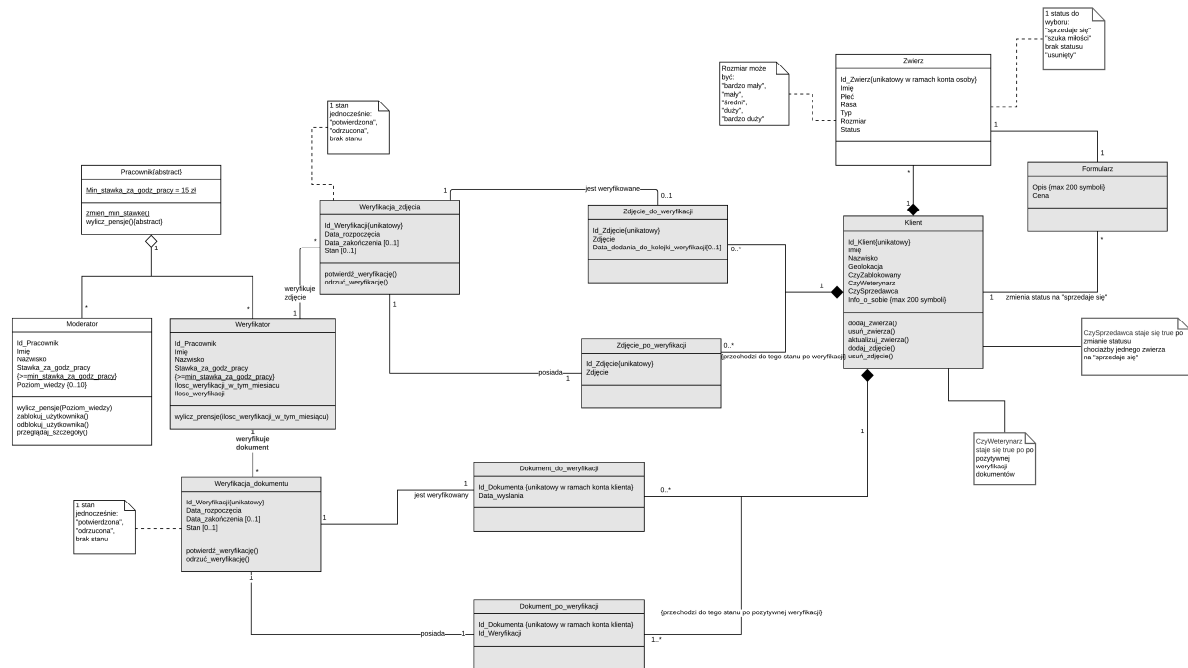


DIAGRAM KLAS – PROJEKTOWY



OPIS METOD PRZEDSTAWIONYCH NA DIAGRAMACH KLAS

Metoda: <u>zmien_min_stawke()</u>
Zmienia minimalną stawkę za godzinę pracy u wszystkich pracowników
Metoda: <u>wylicz_pensję()</u>
Zwraca pensję wybranego pracownika. Dla moderatora uwzględniamy poziom wiedzy, a dla weryfikatora – ilość weryfikacji w tym miesiącu
Metoda: <u>potwierdź_weryfikację()</u> i <u>odrzuć_weryfikację()</u>
Potwierdza lub odrzuca weryfikację klienta. Po potwierdzeniu, dokument lub zdjęcie przechodzi do stanu „potwierdzone” lub „odrzucone”
Metoda: <u>zablokuj_użytkownika()</u> , <u>odblokuj_użytkownika()</u>
Pozwalają moderatorowi blokować oraz odblokować użytkowników aplikacji
Metoda: <u>przeoglądaj_szczegóły()</u>
Za pomocą tej metody, moderator może przeglądać różnego rodzaju ukryte szczegóły konta użytkownika
Metoda: <u>dodaj_zdjęcie()</u> , <u>usuń_zdjęcie()</u>
Metody Klienta, odpowiadające za dodanie oraz usunięcie zdjęć

WYMAGANIA NIEFUNKCJONALNE

System powinien spełniać następujące warunki:

- Ze względu na to, że platforma jest nowa, nie oczekujemy na początku dużej ilości klientów i dlatego, na początek, wystarczy nam, żeby system mógł obsługiwać jednocześnie 1000 użytkowników.
- W przypadku awarii, system powinien zostać naprawiony w przeciągu 24h od momentu przyjęcia zgłoszenia awarii, zatem wysłać klientowi maila z przeprosinami i małym prezentem.

OPIS PRZYSZŁEJ EWOLUCJI SYSTEMU

W pozytywnym rozwoju zdarzeń, planowane jest rozszerzenie funkcjonalności systemu, być może, stworzenie własnych klinik weterynaryjnych.

SŁOWNIK

- **WERYFIKACJA** – proces, podczas którego weryfikator prowadzi klientowi serię testów, po których weryfikator będzie pewien w tym, że klient nie jest botem i używa swoje prawdziwe dane.
- **SPRZEDAWCA ZWIERZĄT** – klient, który ma chociażby jednego zwierza ze statusem „sprzedaje się”.
- **WETERYNARZ** – fizyczna osoba lub instytucja, która świadczy usługi weterynaryjne.
- **KLIENT** – klient serwisu „Piesikot”. Korzysta z usług platformy.

SZCZEGÓŁOWY OPIS PROJEKTU

- 1) Na początku, klient powinien przejść przez rejestrację konta, wpisując swoje imię, nazwisko i geolokację. Także może dodać swoje zdjęcia i informację o sobie (max 200 symboli), jeśli chce. Po rejestracji klient dostaje unikatowe ID. Każde zdjęcie może być weryfikowane przez pracownika firmy. Do weryfikacji zdjęcie posiada tylko unikatowe ID i datę dodania do kolejki weryfikacji. Po ukończeniu procesu weryfikacji, zdjęciu przypisuje się ID weryfikacji i ono zostaje oznaczone jako zweryfikowane w serwisie.
- 2) Po rejestracji, klient od razu może zacząć szukać weterynarza, dla innych opcji potrzebne jest stworzenie konta dla swojego zwierza.
- 3) Jeśli klient posiada chociażby jednego zwierza, to może dla niego stworzyć własne konto. Konto zwierzątka powinno zawierać unikatowe, w ramach konta właściciela, ID, jego płeć, rasę i typ (n.p. pies, kot). Także przechowujemy bieżący status zwierza („sprzedaje się”, „szuka miłości”, „usunięty”, brak statusu) i rozmiar („bardzo mały”, „mały”, „średni”, „duży”, „bardzo duży”).
- 4) Jeśli klient chce zdobyć w serwisie status weterynarza, to powinien on wysłać do działu weryfikacji dokumenty, które potwierdzają jego umiejętności i wykształcenie. Do weryfikacji dokumenty są przechowywane razem z unikatowym ID w ramach konta klienta, który ich wysłał i datą ich wysłania.
Wszystkie dane z dokumentu są przechowywane razem z ID dokumenta i ID weryfikacji, i już po zakończeniu weryfikacji pojawiają się u klienta w profilu wraz ze statusem weterynarza.
- 5) Jeśli klient chce sprzedawać zwierząt, powinien on dostać status sprzedawcy zwierząt. Dla tego wystarczy zmienić status chociażby jednego zwierza na „sprzedaje się”.
Zatem klient powinien będzie wypełnić formularz z krótkim opisem zwierza (max 200 symboli) i jego ceną.
- 6) Dla pracowników przechowujemy ich dane osobowe i unikatowe ID. Pracownicy dzielą się na weryfikatorów i moderatorów, podział jest kompletny. Dla moderatorów dodatkowo przechowujemy ich poziom wiedzy w wykorzystywanych przez naszą firmę technologiach od 1 do 10, a dla weryfikatorów – ilość weryfikacji za bieżący miesiąc pracy w „Piesikocie”. Także powinniśmy przechowywać ich stawkę za godzinę pracy, która nie jest niższa od minimalnej, która wynosi 15 zł, ale jeszcze może ulec zmianie. Potrzebujemy także metody do wyliczenia pensji i zmiany minimalnej pensji. Do wyliczenia pensji moderatora, będziemy brali pod uwagę jego poziom wiedzy, a dla weryfikatorów – ilość weryfikacji
- 7) Każda weryfikacja ma w sobie datę rozpoczęcia i datę zakończenia, unikatowy ID i ID weryfikatora, który ją przeprowadził. Także ma jeden ze stanów: „potwierdzona”, „odrzucona”, brak stanu.

SYSTEM MA UMOŻLIWIAĆ REALIZACJĘ PONIŻSZEJ FUNKCJONALNOŚCI:

- 1) Dodanie/usunięcie/aktualizacja zwierza (Klient)
- 2) Wyświetlenie listy zwierząt na koncie z możliwością dodania/usunięcia/aktualizacji zwierza (Klient)
- 3) Dostanie statusu weterynarza, wysyłając dokument, potwierdzający te umiejętności
- 4) Blokowanie/odblokowanie użytkownika (Moderator)
- 5) Wyświetlenie listy zablokowanych użytkowników z możliwością odblokowania (Moderator)
- 6) Wyświetlenie konta Klienta (Gość)
- 7) Potwierdzenie/odrzućenie weryfikacji klienta z wyświetleniem konta Klienta (Weryfikator)

SCENARIUSZ DLA PRZYPADKU UŻYCIA DOSTANIE STATUSU WETERYNARZA:

Warunek początkowy:	W systemie istnieje co najmniej 1 Weryfikator i Klient, zalogowany Klient wyraża chęć do zdobycia statusu Weterynarza
Główny przepływ zdarzeń:	<ol style="list-style-type: none"> 1. Aktor Klient uruchamia przypadek użycia. 2. System prosi o przesłanie dokumentu(ów), potwierdzających umiejętności weterynaryjne. Aktor przesyła. 3. System prosi o potwierdzenie poprawności przesyłanych dokumentów. Aktor potwierdza. 4. System informuje o przyjęciu dokumentów i oczekuje na ukończenie procesu weryfikacji. 5. Po zmianie stanu weryfikacji na „potwierdzona”, system informuje o tym Klienta i nadaje status Weterynarza.
Alternatywne przepływy zdarzeń:	<ol style="list-style-type: none"> 2a. Aktor nie przesyła dokumenty, koniec przypadku użycia 3a. Aktor zaprzecza, system przechodzi do 2. 5a. Stan weryfikacji się zmienia na „odrzucona”, system informuje o tym Klienta, koniec przypadku użycia
Warunek końcowy:	Klient dostaje status Weterynarza

DIAGRAM AKTYWNOŚCI DLA PRZYPADKU UŻYCIA *DOSTANIE STATUSU WETERYNARZA:*

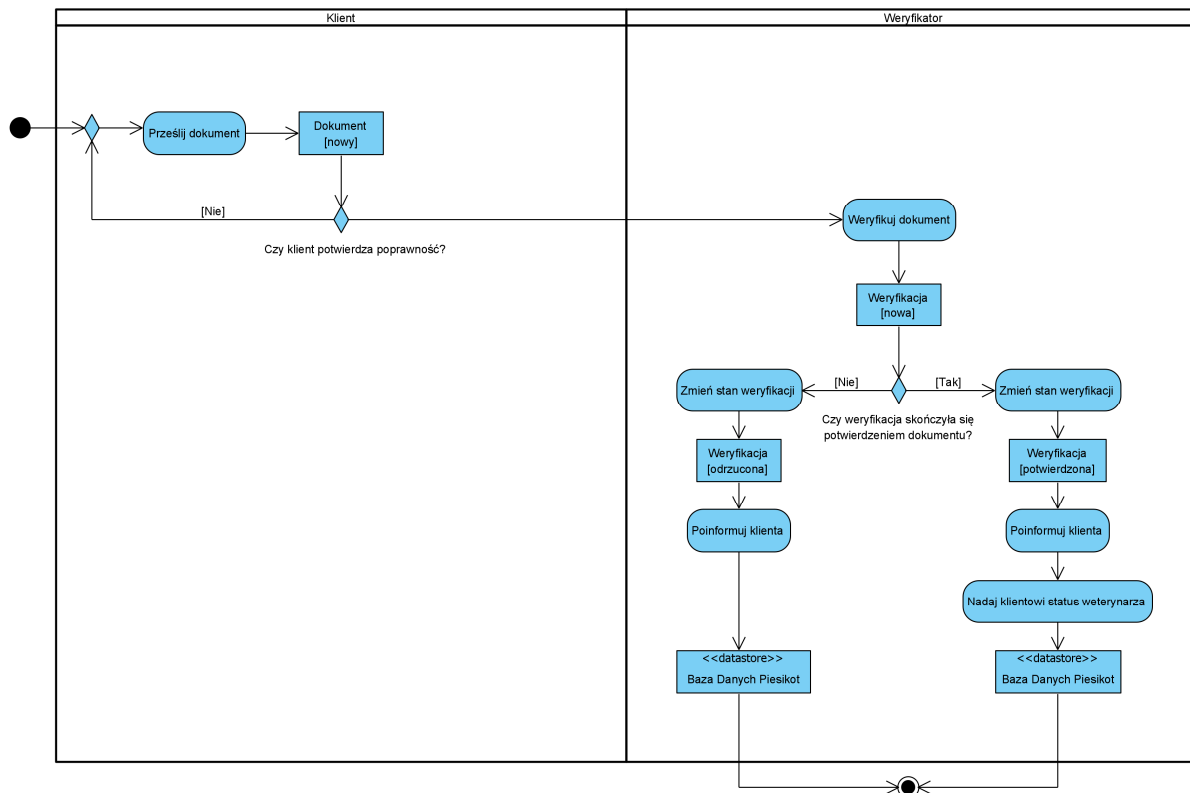


DIAGRAM STANU DLA KLASY *KLIENT*:

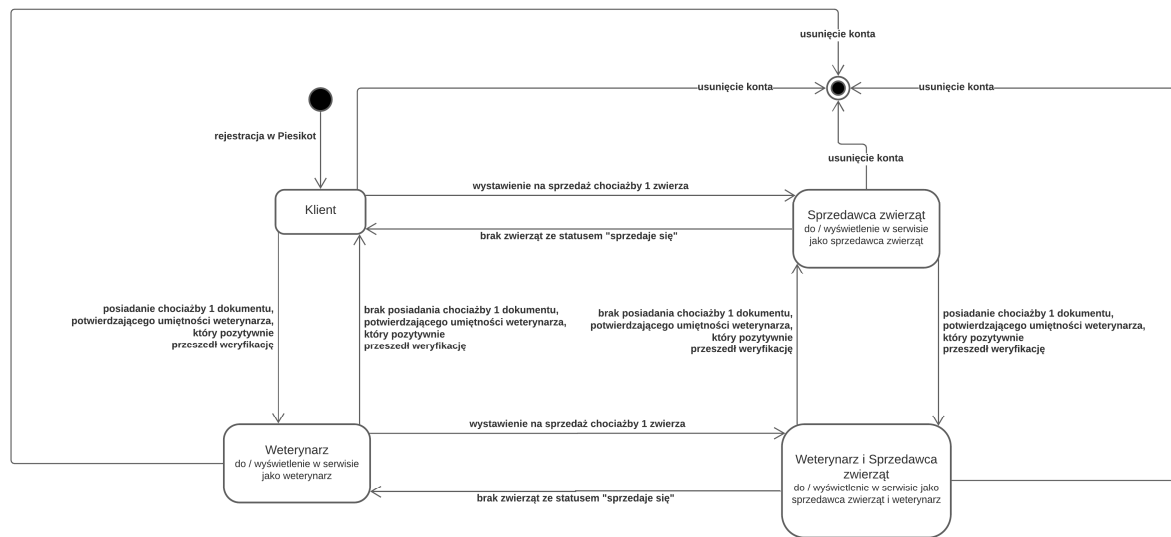
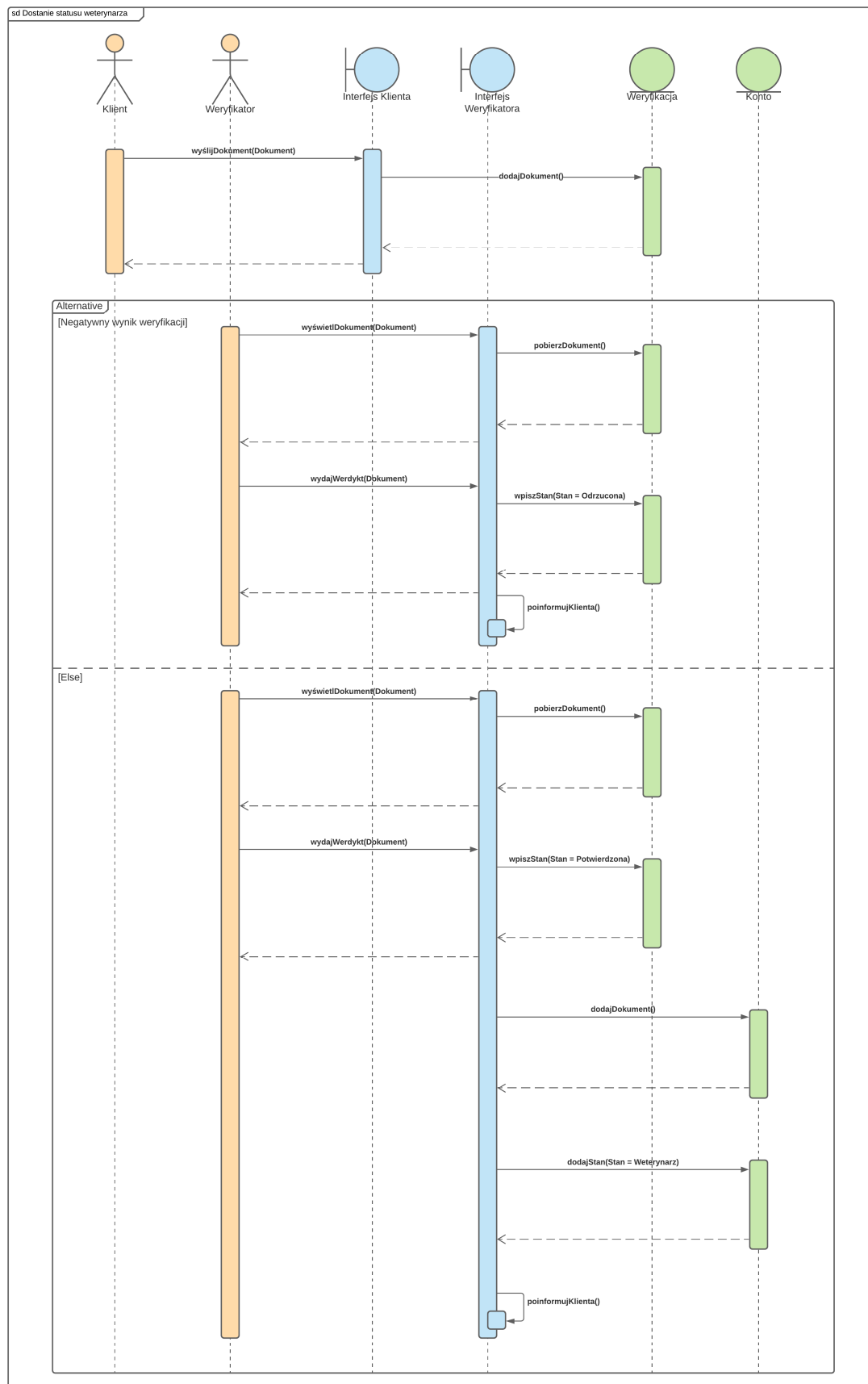
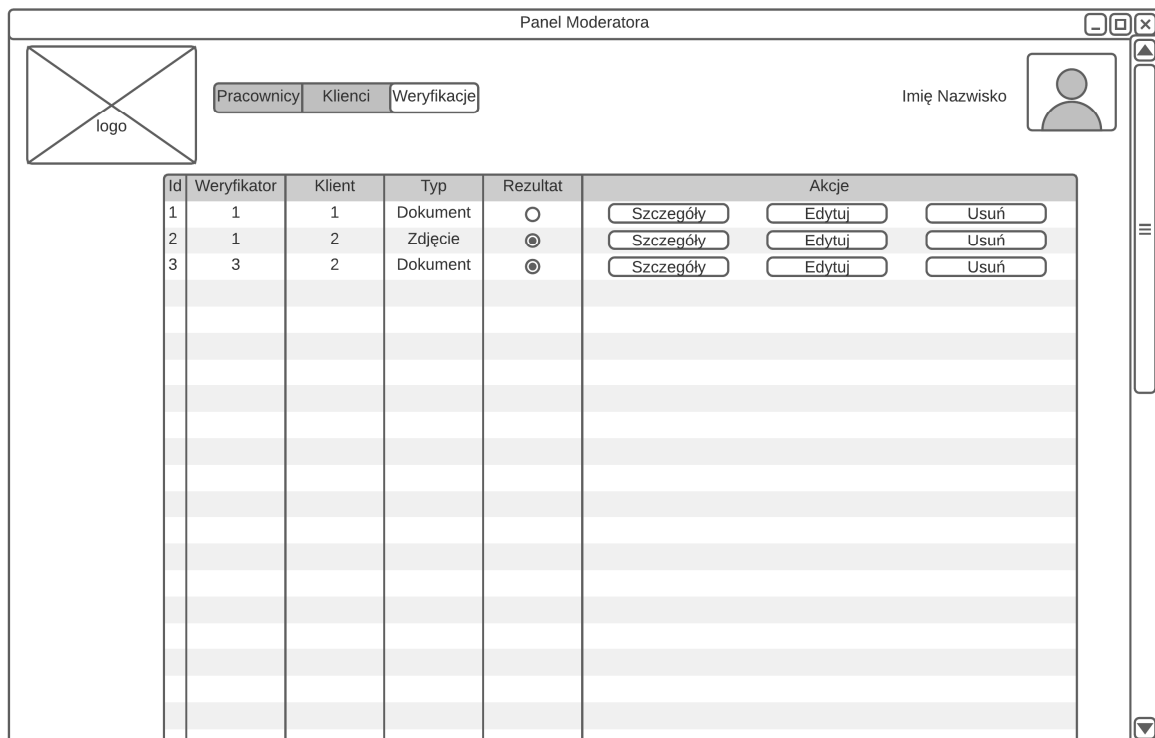


DIAGRAM INTERAKCJI DLA PRZYPADKU UŻYCIA ***DOSTANIE STATUSU WETERYNARZA:***



PROJEKT GUI (MODERATORA):

[illegible][illegible]



OMÓWIENIE DECYZJI PROJEKTOWYCH:

WYBÓR SPOSOBU ZAPEWNIENIA TRWAŁOŚCI

Jako sposób zapewnienia trwałości jednoznacznie zostaje wybrana baza danych, dla efektywnej i wygodnej pracy z danymi z różnych urządzeń w sieci.

Jak już było omówiono wcześniej, firma jest młoda i rozwijająca się, dlatego wykorzystanie płatnego oprogramowania nie jest żądane.

Także, dla sprawnego działania aplikacji, jest wymagana realizacja zasad ACID przez Bazę Danych.

Z najpopularniejszych baz danych, PostgreSQL i MySQL pasują do tego opisu, ale po przejściu MySQL pod opiekę Oracle i niejasność jego rozwoju,

PostgreSQL – jedyny kandydat na bycie SZBD w Piesikot.

Dla wygodnego i wydajnego mapowania klas będzie wykorzystany Hibernate.

OMÓWIENIE DECYZJI PODJĘTYCH PODCZAS ANALIZY DYNAMICZNEJ

Jak widać z „nowego” diagramu klas, został on prawie całkowicie zmieniony, podczas tych zmian było wprowadzono dużo niestandardowych decyzji biznesowych, które będą uzasadnione niżej.

1. Asocjacja z atrybutem Osoba-Zwierz przy zmianie stanu zwierza na „sprzedaje się” została zamieniona na asocjacje Osoba-Formularz i Zwierz-Formularz, dlatego, że Formularz wystąpi jako tabela pośrednia.
2. Asocjacje z atrybutem Weryfikator-Zdjęcie_do_weryfikacji i Weryfikator-Dokument_do_weryfikacji, jak i ich atrybut Weryfikacja zostały zmienione w nieco bardziej kreatywny sposób. Weryfikacja została podzielona na Weryfikacja_zdjęcia i Weryfikacja_dokumentu dla prezentowania dziedziczenia Table Per Class (TPC). Ten rodzaj dziedziczenia został obrany głównie przez łatwe i szybkie łączenie zapytań do wyświetlenia danych za pomocą UNION. Te klasy również występują jako tabele pośrednie.
3. Dynamic połączenia ze Zdjęciami i Dokumentami zostały zamienione na kompozycje. Na początku myślałem o zamianie <<dynamic>> poprzez XOR, jak to było omówiono na wykładzie, ale w tym przypadku to by znaczyło, że Osoba może jednocześnie mieć wszystkie zweryfikowane zdjęcia albo wszystkie nie zweryfikowane, co jest niemożliwe do dopuszczenia z punktu widzenia logiki biznesowej.
4. Dziedziczenie Pracowników zostało zamienione zgodnie z Table Per Class (TPC), ale Pracownik jako klasa pozostał się dla przechowywania zmiennej statycznej. Ten typ dziedziczenia również został wybrany nie tylko do łatwego łączenia wyników przez UNION, ale również dla wyświetlenia całej informacji o pracowniku bez wykorzystania JOIN, co pozytywnie wpłynie na wydajność aplikacji (na przykład, w GUI moderatora).

5. Klasa Osoba została zamieniona klasą Klient i dziedziczeniem Table Per Hierarchy (TPH). Dziedziczenia Osoby nie robiły nic oprócz zmiany statusu i, ewentualnie, dodania dokumentów do wyświetlenia na koncie Klienta, dlatego TPH jest, moim zdaniem, najlepszym wyborem.