Projekt aplikacji

Spis treści

[1. Wstęp](#_heading=h.gjdgxs) **2**

[2. Cel i zakres dokumentu](#_heading=h.30j0zll) **2**

[3. Definicja architektury aplikacji](#_heading=h.1fob9te) **2**

[4. Cele i ograniczenia architektury](#_heading=h.3znysh7) **2**

[5. Obraz logiczny aplikacji](#_heading=h.2et92p0) **2**

[5.1 Charakterystyka pakietów](#_heading=h.tyjcwt) 2

[5.2 Diagram klas aplikacji](#_heading=h.3dy6vkm) 2

[5.3 Specyfikacja funkcji i metod aplikacji](#_heading=h.1t3h5sf) 2

[6. Dynamiczny obraz modelowanej aplikacji](#_heading=h.4d34og8) **3**

[6.1 Diagram sekwencji UML dla obiektów](#_heading=h.2s8eyo1) 3

[6.2 Diagram aktywności UML dla obiektów](#_heading=h.17dp8vu) 3

[7. Statyczny obraz modelowanej aplikacji](#_heading=h.3rdcrjn) **9**

[7.1 Diagram komponentów UML](#_heading=h.26in1rg) 9

[7.2 Diagram instalacji UML](#_heading=h.lnxbz9) 10

[8. Projekt bazy danych](#_heading=h.35nkun2) **10**

# 1. Wstęp

Dokument przedstawia projekt aplikacji Hotel Dla Psów. Aplikacja ma pomóc w obsłudze hotelu dla psów, ma obsłużyć wszelką dokumentację, która była robiona podczas planowania wizyt/przyjmowania psów do hotelu/wyjazdu psów z hotelu.

# 2. Cel i zakres dokumentu

Celem dokumentu jest przedstawienie projektu aplikacji. Przedstawienie diagramu sekwencji i diagramów aktywności, które zachodzą w projekcie. Opis stanów w jakich mogą znajdować się obiekty. W dokumencie znajduje się także diagram instalacji aplikacji, diagram komponentów oraz pakietów.

# 3. Definicja architektury aplikacji

Architektura trójwarstwowa to dobrze ugruntowana architektura aplikacji, w której aplikacje są porządkowane na trzech logicznych i fizycznych warstwach obliczeniowych. Są to: warstwa prezentacji, czyli interfejs użytkownika; warstwa aplikacji, w której dane są przetwarzane; oraz warstwa danych, w której przechowywane i zarządzane są dane związane z aplikacją.

Ponieważ każda warstwa działa na własnej infrastrukturze, zespoły programistów mogą projektować poszczególne warstwy symultanicznie, a jedna warstwa może być modernizowana i skalowana bez naruszania innych warstw. Stanowi to podstawową zaletę architektury trójwarstwowej.

**4. Cele i ograniczenia architektury**

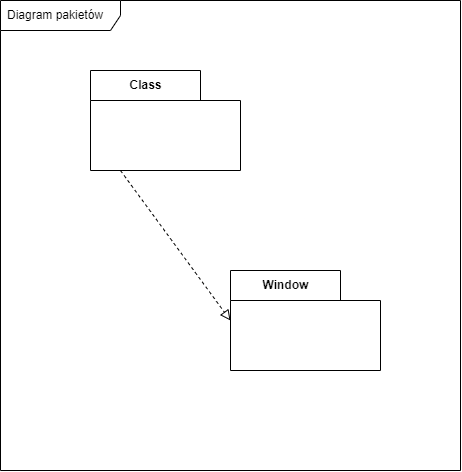
Celem architektury trójwarstwowej jest rozwiązanie następujących problemów:

* odciążenie stacji roboczej klienta (w szczególności w odniesieniu do klasycznych architektur danych [klient-serwer](https://pl.frwiki.wiki/wiki/Client-serveur) - typowych dla aplikacji w kontekście [Oracle](https://pl.frwiki.wiki/wiki/Oracle_(base_de_donn%C3%A9es)) / [Unix](https://pl.frwiki.wiki/wiki/Unix) );
* biorąc pod uwagę różnorodność platform (serwery, klienci, języki itp.);
* wprowadzenie tzw. „cienkich” klientów (bardziej powiązanych z technologiami [Intranet](https://pl.frwiki.wiki/wiki/Intranet) / [HTML](https://pl.frwiki.wiki/wiki/Hypertext_Markup_Language) niż z samą architekturą trójwarstwową);
* poprawę bezpieczeństwa danych poprzez usunięcie powiązania między klientem a danymi. Zadaniem serwera, oprócz czysto biznesowego przetwarzania, jest sprawdzenie integralności i aktualności danych przed wysłaniem ich do warstwy dostępu do danych;
* zerwanie wyłącznego powiązania własności między aplikacją a danymi. W tym modelu bazę danych można łatwiej ustandaryzować i zintegrować z [hurtownią danych](https://pl.frwiki.wiki/wiki/Entrep%C3%B4t_de_donn%C3%A9es) ;
* lepszy rozkład obciążenia między różnymi [serwerami aplikacji](https://pl.frwiki.wiki/wiki/Serveur_d%27applications) .

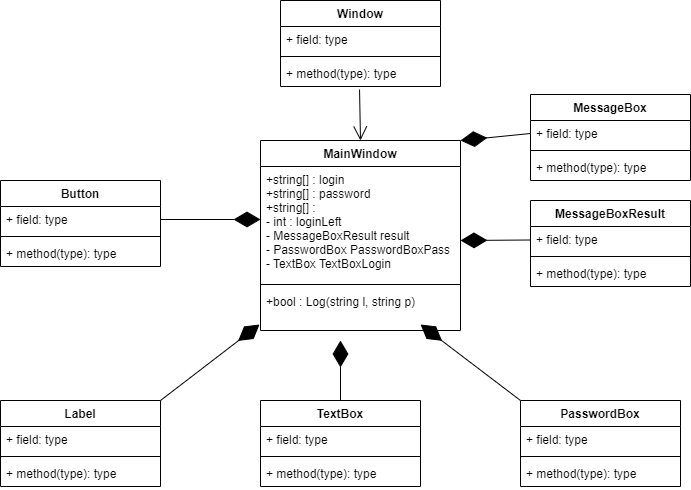
Ograniczeniem tej architektury jest fakt że kolejne warstwy wpływają na wydajność oprogramowania, dodatkowe operacje związane z odebraniem danych z poprzedniej warstwy i przygotowaniem danych do przekazania do kolejnej warstwy, dodatkowe transakcje i synchronizacja

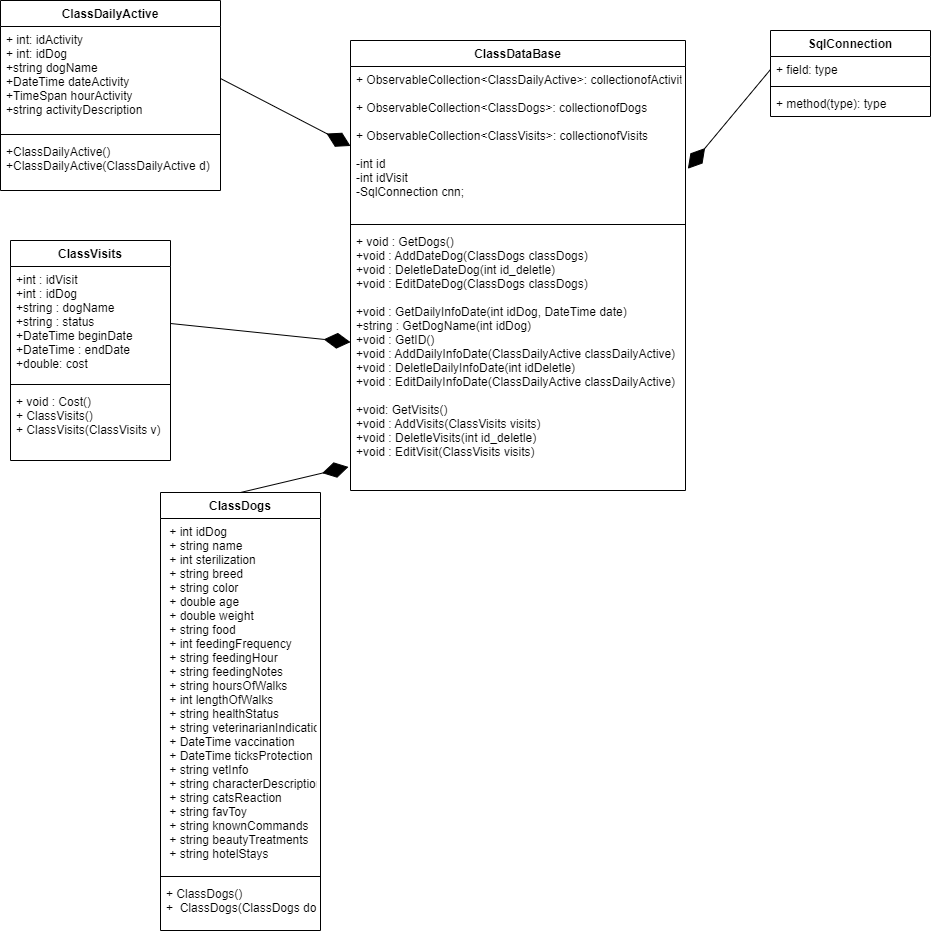
# 5. Obraz logiczny aplikacji

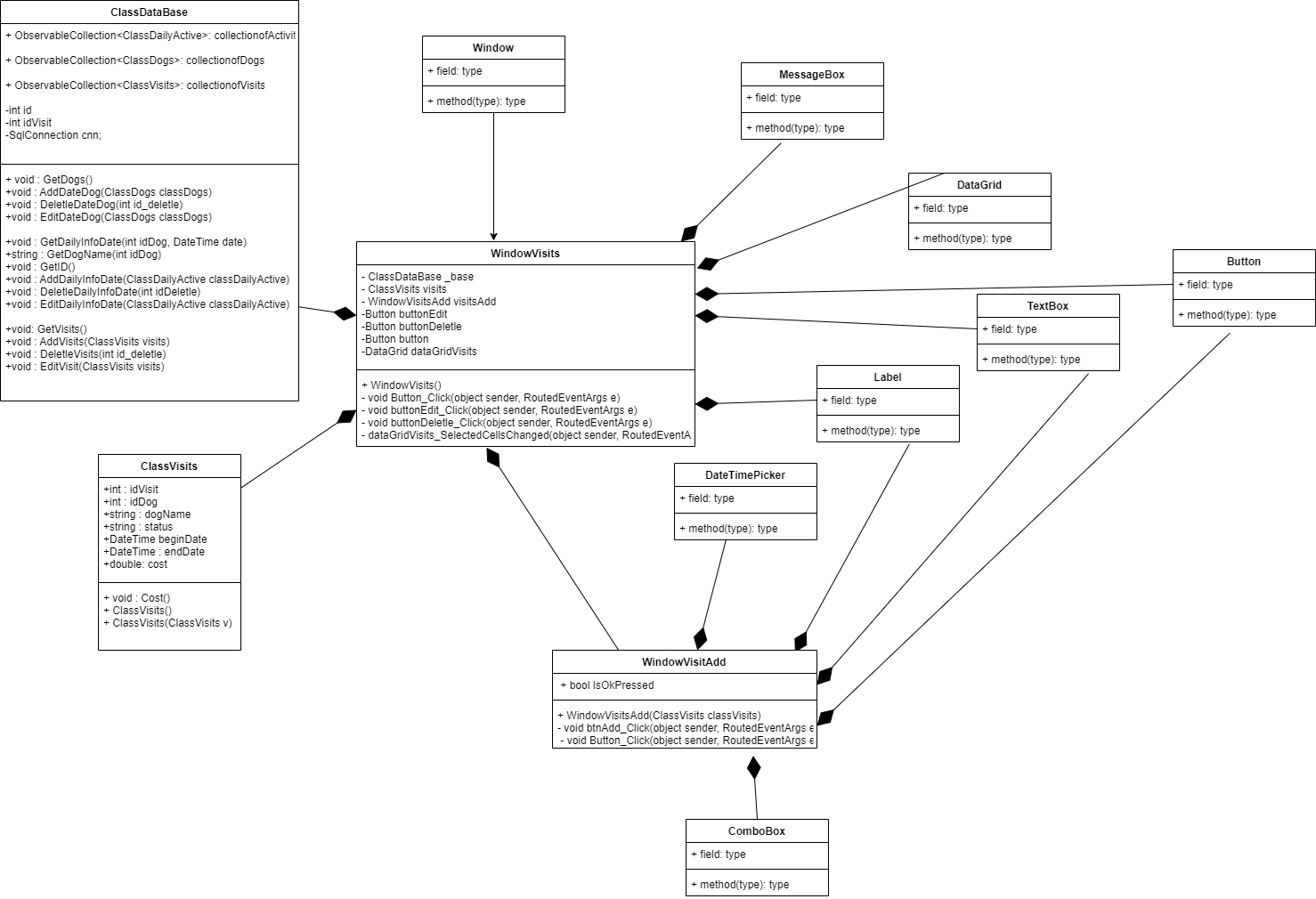
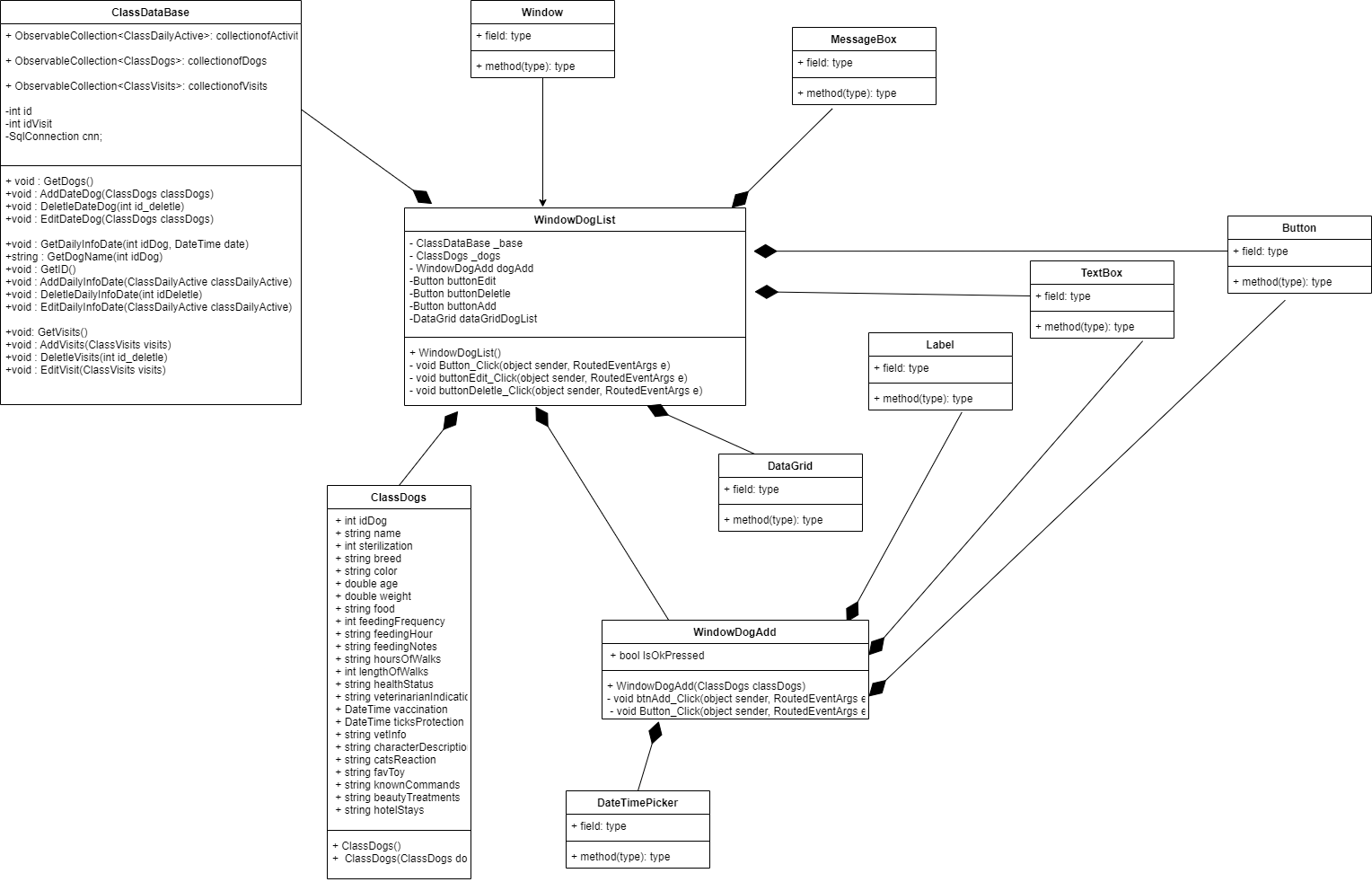
## 5.1 Diagram pakietów



## 5.2 Diagram klas aplikacji







## DiagramKlasWindowActivity.drawio

## 5.3 Specyfikacja funkcji i metod aplikacji

Openconnection

Closeconnection

Getdogs

AddDateDog

DeleteDateDog

EditDateDog

GetDailyInfoDate

GetDogName

GetID

AddDailyInfoDate

DeletleDailyInfoDate

EditDailyInfoDate

GetVisits

AddVisits

DeleteVisit

EditVisit

ButtonLogin\_Click

HandleHandleInput

ButtonSearch\_Click

ButtonAdd\_Click

ButtonEdit\_Click

ButtonDelete\_Click

btnAdd\_Click

Button\_Click

btnAdd\_Click

Button\_Click

ButtonEdit\_Click

ButtonAdd\_Click

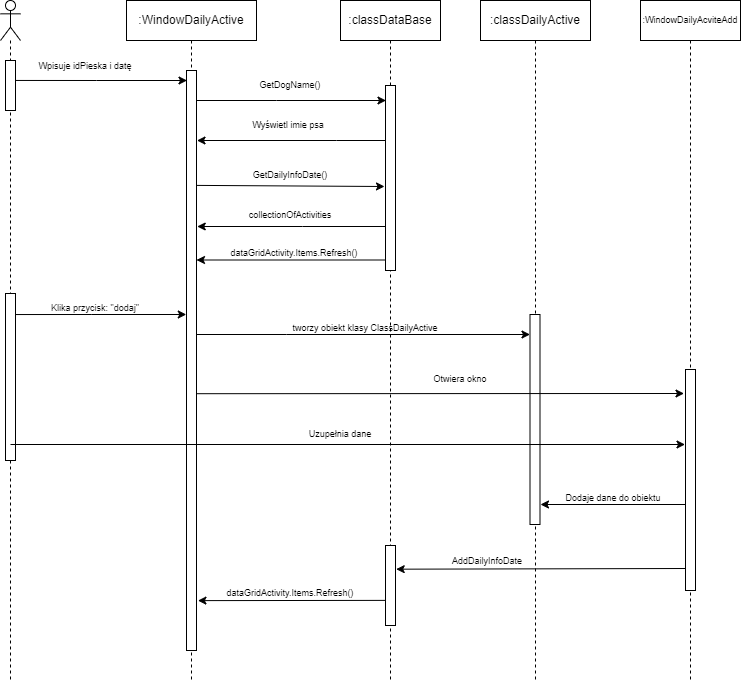
ButtonVisit\_Click

ButtonDailyActive\_Click

ButtonDogList\_Click

# 6. Dynamiczny obraz modelowanej aplikacji

## 6.1 Diagram sekwencji UML dla obiektów



6.2 Diagram aktywności UML dla obiektów

Diagram aktywności - “Planowanie wizyty”

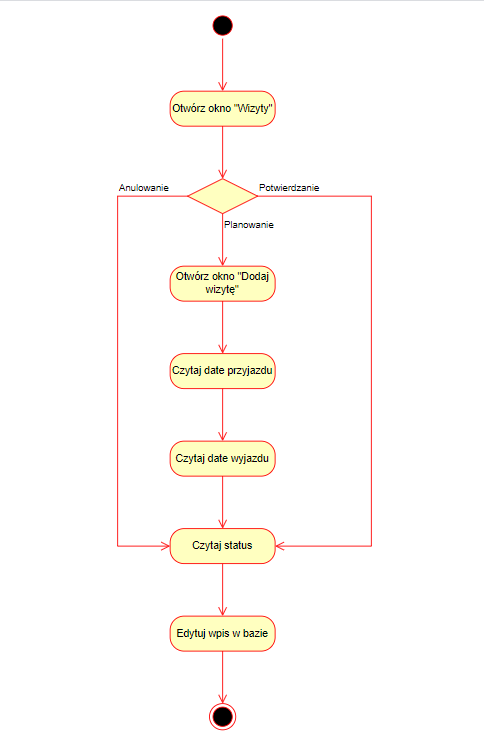
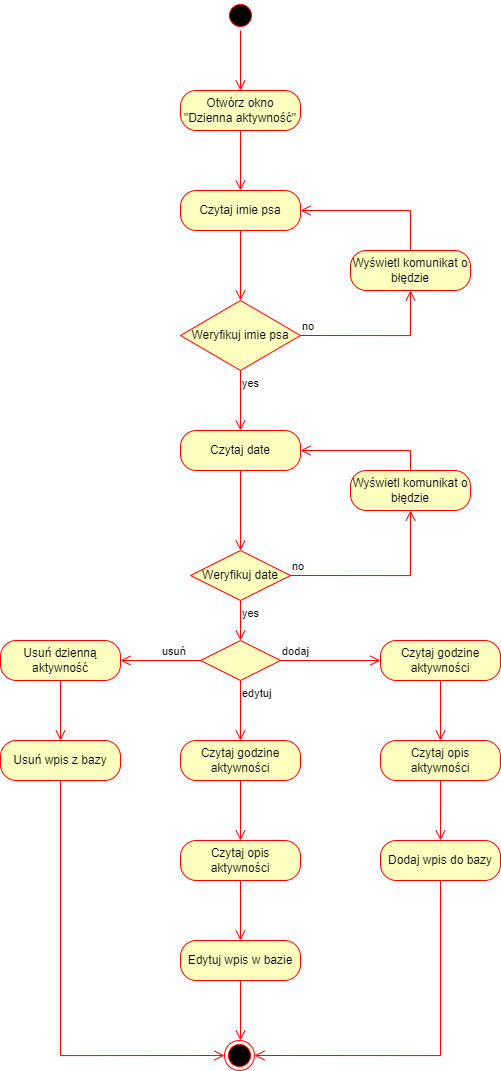
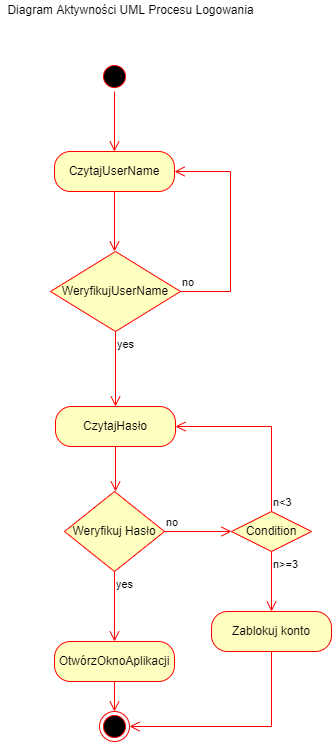
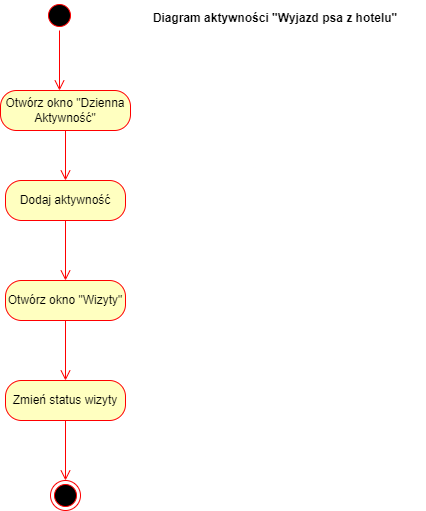
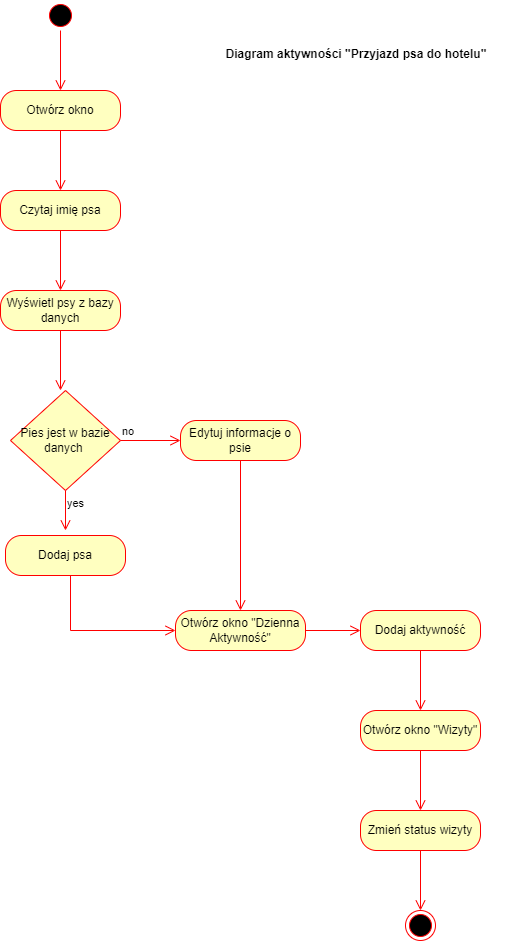


Diagram aktywności - “Dziennik Aktywności”

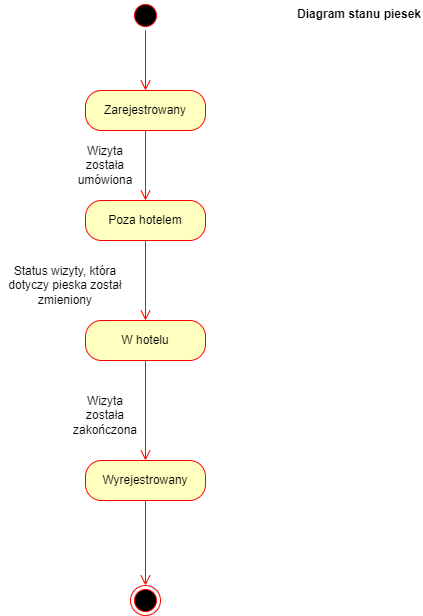
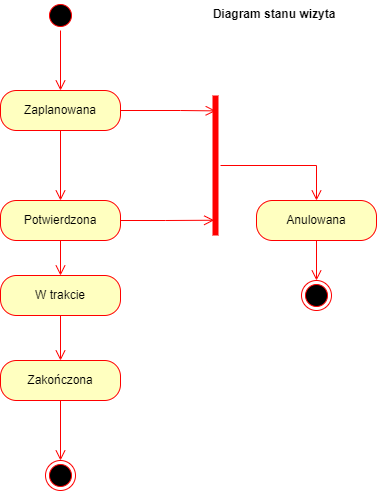






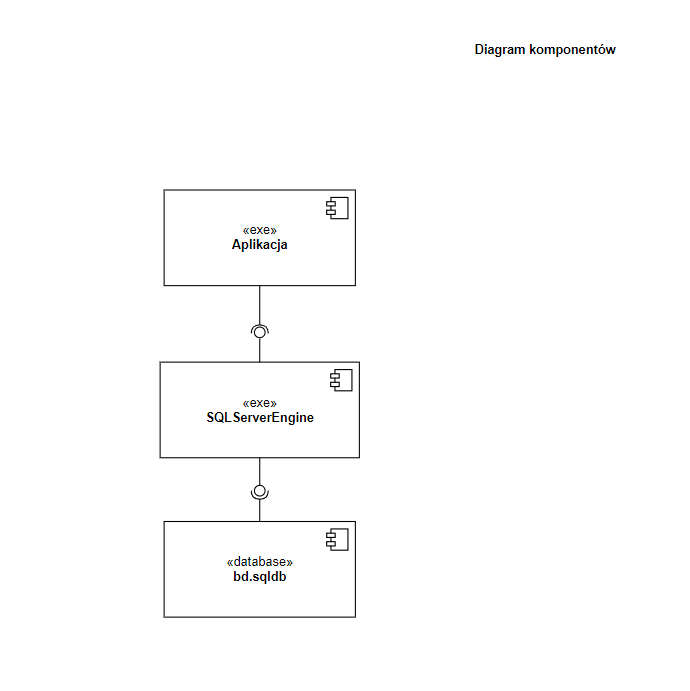


6.3 Diagramy stanu UML dla obiektów

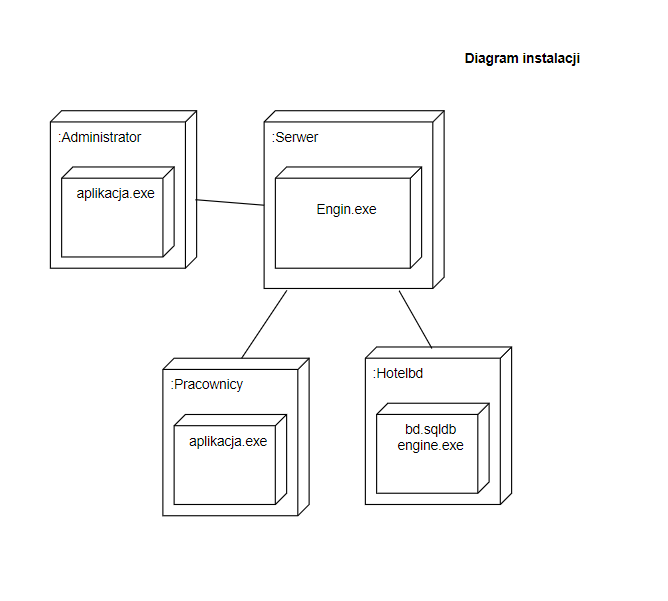


# 7. Statyczny obraz modelowanej aplikacji

## 7.1 Diagram komponentów UML



## 7.2 Diagram instalacji UML



# 8. Projekt bazy danych

Spis procedur składowanych:

dbo.addDailyInfoDate - dodaje informacje na temat daty oraz godziny aktywności oraz opis aktywności danego psa

dbo.addDateDog - dodawany jest nowy pies oraz informacje o nim

dbo.addVisit - dodaje nową wizytę psa w hotelu

dbo.deleteDailyInfoDate - usuwa informacje na temat opisu, daty oraz godziny aktywności

dbo.deleteDog - usuwa informacje na temat psa z bazy danych hotelu

dbo.deleteVisit - usuwa dane na temat wizyty psa w hotelu

dbo.editDailyInfoDate - edytujemy informacje na temat dziennej aktywności psa

dbo.editDateDog - edytujemy informacje na temat psa

dbo.editVisit - za pomocą tej procedury edytowane są wartości poszczególnego wiersza w tabeli visits

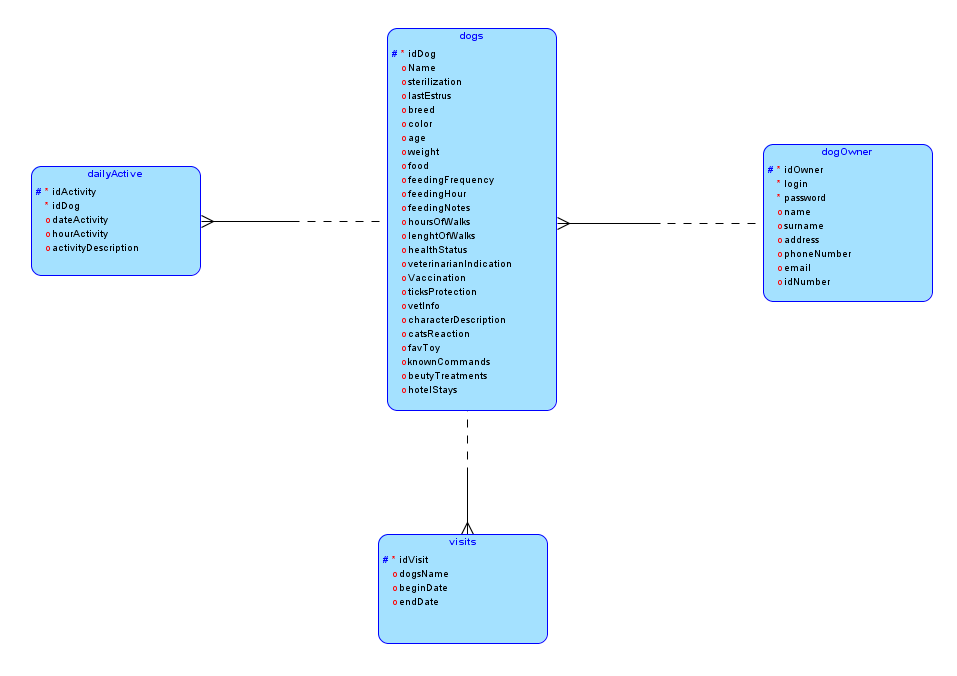
dbo.getDailyInfo - zwraca informacje na temat daty oraz godziny aktywności oraz opis aktywności danego psa

dbo.getDogInfo - zwraca wszystkie informacje na temat danego psa zawarte w bazie danych

dbo.getDogName - po podaniu id psa zwraca informacje na temat jego imienia

dbo.getVisits - zwraca id wizyty, datę rozpoczęcia oraz datę zakończenia danej wizyty

Logiczny diagram bazy danych:



Relacyjny diagram bazy danych

