Analiza specyfikacji wymagań

Hotel dla zwierząt

Spis treści

[1. Wstęp 3](#_Toc17439)

[2. Cele analizy specyfikacji 3](#_Toc15688)

[3. Zakres specyfikacji 3](#_Toc29687)

[4. Analiza wymagań funkcjonalnych - model use case'ów 3](#_Toc27866)

[4.1 Specyfikacja aktorów 3](#_Toc24843)

[4.2 Lista use case'ów 3](#_Toc1753)

[4.3 Diagram use case'ów UML 4](#_Toc29056)

[4.4 Specyfikacja use case’ów 4](#_Toc13252)

[4.5 Diagramy aktywności UML dla use case’ów 4](#_Toc25345)

[5. Analiza wymagań niefunkcjonalnych 4](#_Toc15644)

[5.1 Interfejsy użytkownika 4](#_Toc22916)

[5.2 Interfejsy programowe 4](#_Toc7574)

[6. Analiza wymagań dotyczących jakości modelowanego systemu 4](#_Toc31027)

[7. Analiza warunków serwisowania 5](#_Toc21455)

[8. Analiza ograniczeń architektury systemu 5](#_Toc3283)

[9. Model bazy danych 5](#_Toc5039)

[9.1 Specyfikacja atrybutów bazy danych 5](#_Toc1046)

[9.2 Specyfikacja operacji na danych 7](#_Toc5088)

[9.3 Specyfikacja reguł poprawności i zgodności typów danych 7](#_Toc12726)

# 1. Wstęp

W podanym dokumencie przedstawiono następujące elementy: analizę wymagań funkcjonalnych, analizę wymagań niefunkcjonalnych, analizę wymagań dotyczących jakości modelowanego systemu, analizę warunków serwisowania oraz analizę ograniczeń architektury systemu. Dodatkowo w ostatnim rozdziale przedstawiono model bazy danych, który został podzielony na specyfikację atrybutów bazy danych, specyfikację operacji na bazie danych oraz s[pecyfikację reguł poprawności i zgodności typów danych](#_heading=h.1y810tw). W rozdziale dotyczącym analizie wymagań funkcjonalnych wyszczególniono, specyfikację aktorów, listę use case’ów, diagram use case UML oraz diagram aktywności UML. Następnie rozdział dotyczący analizy metod niefunkcjonalnych został podzielony na poszczególne interfejsy: użytkownika, sprzętowy, komunikacyjny oraz programowy.

# 2. Cele analizy specyfikacji

Celami analizy specyfikacji wymagań jest stworzenie use case’ów, określenie wymagań niefunkcjonalnych: interfejsów użytkownika, sprzętowych, komunikacyjnych i programowych. Przeanalizowanie wymagań jakości, warunków serwisowania i ograniczeń architektury.

# 3. Zakres specyfikacji

Oprogramowanie obsługuje system bazo-danowy w celu ułatwienia prowadzenia dokumentacji oraz obsługi hotelu. Oprogramowanie działa poprawnie na systemach Windows 10/11

Następujące funkcje oprogramowania:

* Logowanie do oprogramowania,
* Dodawanie, edycja oraz usuwanie zwierząt,
* Dodawanie, edycja oraz usuwanie właścicieli zwierząt,
* Dodawania, edycja oraz usuwanie wizyt w hotelu.

Użytkownikiem systemu będzie osoba zatrudniona przez hotel dla zwierząt, również właściciel danego przedsiębiorstwa będzie użytkownikiem tego oprogramowania. Użytkownicy mają prawo do edycji, usuwania oraz dodawania krotek do bazy danych.

Tabele takie jak wizyty, właściciele oraz aktywności są połączone w sposób naturalny z tabelą zawierającą informacje o zwierzętach.

Wymagania niefunkcjonalne oprogramowania:

* Zapewnienie bezpieczeństwa danych w celu zachowania poufności oraz przepisów RODO,
* Dokładność informacji, które są niezbędne do prawidłowego oraz bezpiecznego dla wszystkich funkcjonowania hotelu,
* Szybki czas reakcji na zapytania użytkownika
* Łatwość korzystania z aplikacji
* Mała liczba błędów lub ich brak w trakcie korzystania z aplikacji

# 4. Analiza wymagań funkcjonalnych - model use case'ów

## 4.1 Specyfikacja aktorów

We wszystkich use casach występuje jeden aktor - Pracownik;

## 4.2 Lista use case'ów

* Logowanie
* PlanowanieWizyty
* PrzyjazdZwierzęcia
* ProwadzenieDziennikaAktywności
* WyjazdZwierzącia

## 4.3 Diagram use case'ów UML

* DiagramUCLogowanie.jpg
* DiagramUCPlanowanieWizyty.jpg
* DiagramUCPrzyjazdZwierzącia.jpg
* DiagramUCProwadzenieDziennikaAktywności.jpg
* DiagramUCWyjazdZwierzącia.jpg

## 4.4 Specyfikacja use case’ów

* Logowanie.docx
* PlanowanieWizyty.docx
* PrzyjazdZwierzęcia.docx
* ProwadzenieDziennikaAktywności.docx
* WyjazdZwierzęcia.docx

## 4.5 Diagramy aktywności UML dla use case’ów

* DiagramAktywnościLogowanie.jpg
* DiagramAktywnościPlanowanieWizyty.jpg
* DiagramAktywnościPrzyjazdZwierzęcia.jpg
* DiagramAktywnosciDziennik.jpg
* DiagramAktywnościWyjazdZwierzęcia.jpg

# 5. Analiza wymagań niefunkcjonalnych

## 5.1 Interfejsy użytkownika

Komunikacja urządzenie - użytkownik odbywa się przy pomocy urządzenia wskazującego, jest to interfejs graficzny, który ma na celu umożliwienie użytkownikowi łatwego oraz szybkiego odczytywania danych.

## 5.2 Interfejsy programowe

interfejsy programistyczne - WIN WPF

# 6. Analiza wymagań dotyczących jakości modelowanego systemu

Głównym wymaganiem klienta było wymaganie dotyczące użytkowania systemu, to znaczy łatwość wdrożenia systemu w hotelu dla zwierząt oraz łatwość późniejszej obsługi oraz wykonywania zadań na danym oprogramowaniu.

Dodatkowym wymaganiem było wymaganie odporności na awarie danego oprogramowania oraz odporność systemu informatycznego na błędy oraz awarie.

Na samym końcu wzięto pod uwagę efektywność systemu w odniesieniu do zachowywania w czasie, pobierania danych z bazy danych oraz wykorzystania zasobów komputera.

# 7. Analiza warunków serwisowania

Warunki serwisowania zawarte z hotelem są zawarte na podstawie umowy serwisowej oprogramowania. Oprogramowanie będzie regularnie serwisowane w celu eliminacji wszelkich występujących w trakcie użytkowania oprogramowania błędów. W umowie serwisowej zawarto również możliwość dalszego rozwoju oprogramowania w zależności od zapotrzebowania klienta.

# 8. Analiza ograniczeń architektury systemu

Do ograniczeń systemu należą:

* brak metod analitycznych, które jednoznacznie rozstrzygają, czy system spełnia oczekiwania
* problemy komunikacyjne między interesariuszami
* brak dedykowanego architekta oprogramowania.

# 9. Model bazy danych

## 9.1 Specyfikacja atrybutów bazy danych

W bazie danych możemy wyróżnić następujące tabele, które zawierają następujące atrybuty:

TABLE Owner

ID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

Name VARCHAR(100),

Surname VARCHAR(100),

Address VARCHAR(255),

PhoneNumber VARCHAR(20),

Email VARCHAR(100)

TABLE Animal

ID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

Name VARCHAR(100),

OwnerID INT NOT NULL,

FOREIGN KEY (OwnerID) REFERENCES Owner(ID)

TABLE Details

ID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

Breed VARCHAR(100),

Color VARCHAR(50),

AnimalID INT,

FOREIGN KEY (AnimalID) REFERENCES Animal(ID)

TABLE FeedingRules

ID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

FoodType VARCHAR(100),

Frequency INT,

Hours VARCHAR(100),

AdditionalNotes VARCHAR(255),

AnimalID INT,

FOREIGN KEY (AnimalID) REFERENCES Animal(ID)

TABLE Health

ID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

HealthStatus VARCHAR(100),

Vaccination DATETIME,

AdditionalNotes VARCHAR(255),

AnimalID INT,

FOREIGN KEY (AnimalID) REFERENCES Animal(ID)

TABLE DailyActivity

ID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

Date DATE,

Hour VARCHAR(20),

AdditionalNotes VARCHAR(255),

AnimalID INT,

FOREIGN KEY (AnimalID) REFERENCES Animal(ID)

TABLE [User]

ID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

Login VARCHAR(100),

Password VARCHAR(100)

TABLE Visit

ID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

BeginDate DATETIME,

EndDate DATETIME,

AnimalID INT,

FOREIGN KEY (AnimalID) REFERENCES Animal(ID)

TABLE WalksRules

ID INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

Hours VARCHAR(100),

Length INT,

AnimalID INT,

FOREIGN KEY (AnimalID) REFERENCES Animal(ID)

## 9.2 Specyfikacja operacji na danych

Przy pomocy oprogramowania możemy przeprowadzić następujące operacje na bazach danych:

addDailyInfoDate - dodajemy do bazy danych informacje na temat dziennej aktywności zwierząt w hotelu

addDateAnimal - dodajemy informacje na temat zwierząt przebywających w hotelu

addVisit - dodajemy wizytę danego zwierzęcia w hotelu

deleteDailyInfoDate - usuwamy informację na temat dziennej aktywności

deleteAnimal - usuwamy dane na temat zwierzęcia

deleteVisit - usuwamy informacje na temat wizyty

editDailyInfoDate - edycja informacji na temat dziennej aktywności

editDateAnimal - edycja informacji na temat zwierzęcia

editVisit - edycja informacji na temat wizyty zwierzęcia

getDailyInfoDate - pobieranie danych z tabeli DayliInfoDate

getAnimalInfo - pobiera danych z tabeli na temat dodanych zwierząt

getAnimalName - pobiera informacje na temat imienia danego zwierzęcia

getVisists - pobiera informacje na temat wizyt w hotelu danego zwierzęcia

## 9.3 Specyfikacja reguł poprawności i zgodności typów danych

Nie dotyczy