

**UNIwersYTET OPOLSKI**

Wydział Matematyki, Fizyki i Informatyki

Instytut Informatyki

Praca inżynierska

**Natalia Szymczak**

**Aplikacja mobilna dla posiadaczy koni**

MOBILE APPLICATION FOR HORSE OWNER

Praca wykonana pod kierunkiem

dr Jacka Iwańskiego

OPOLE 2022

**Streszczenie:**

**Abstract:**

**Keywords:**

**Klasyfikacja tematyczna wg MSC 2020:**

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>1</b>
<b>2</b>	<b>Przegląd istniejących rozwiązań</b>	<b>2</b>
<b>3</b>	<b>Technologie użyte w pracy</b>	<b>3</b>
3.1	Microsoft SQL Server 2019 . . . . .	3
3.2	Microsoft SQL Server Managemnet Studio . . . . .	3
3.3	Microsoft Visual Studio 2022 . . . . .	3
3.4	Windows Presentation Foundation . . . . .	3
3.5	Xamarin . . . . .	3
3.6	Android Device Menager . . . . .	3
3.7	NuGet . . . . .	3
3.8	Entity Framework . . . . .	3
3.9	Ten do wykresów . . . . .	3
3.10	MVVM Toolkit . . . . .	3
3.11	Xamarin Community Toolkit . . . . .	3
3.12	ZXing.Net.Mobile.Forms . . . . .	3
3.13	Structured Query Language . . . . .	3
<b>4</b>	<b>Specyfikacja wymagań</b>	<b>4</b>
4.1	Opis wycinka rzeczywistości . . . . .	4
4.2	Wymagania funkcjonalne . . . . .	5
4.3	Wymagania niefunkcjonalne . . . . .	10
<b>5</b>	<b>Baza danych</b>	<b>11</b>
5.1	Model konceptualny . . . . .	11
5.2	Model logiczny . . . . .	23
5.3	Model fizyczny . . . . .	23
<b>6</b>	<b>Projekt systemu</b>	<b>24</b>
6.1	Model projektowanego systemu . . . . .	24
6.2	Wybrane aspekty implementacyjne . . . . .	24
<b>7</b>	<b>Testy aplikacji</b>	<b>25</b>
7.1	Unit testy . . . . .	25
7.2	Test case . . . . .	25

7.3	Baza błędów . . . . .	25
<b>8</b>	<b>Dokumentacja użytkownika</b>	<b>26</b>
8.1	Aplikacja desktopowa . . . . .	26
8.2	Aplikacja mobilna . . . . .	26
<b>9</b>	<b>Podsumowanie</b>	<b>27</b>
<b>10</b>	<b>Spis listingów</b>	<b>30</b>
<b>11</b>	<b>Opis zawartości APD</b>	<b>32</b>

# Rozdział 1

## Wstęp

## Rozdział 2

### Przegląd istniejących rozwiązań

## Rozdział 3

### Technologie użyte w pracy

- 3.1 Microsoft SQL Server 2019
- 3.2 Microsoft SQL Server Managemnet Studio
- 3.3 Microsoft Visual Studio 2022
- 3.4 Windows Presentation Foundation
- 3.5 Xamarin
- 3.6 Android Device Menager
- 3.7 NuGet
- 3.8 Entity Framework
- 3.9 Ten do wykresów
- 3.10 MVVM Toolkit
- 3.11 Xamarin Community Toolkit
- 3.12 ZXing.Net.Mobile.Forms
- 3.13 Structured Query Language

# Rozdział 4

## Specyfikacja wymagań

### 4.1 Opis wycinka rzeczywistości

Aplikacja przeznaczona jest dla klubów jeździeckich, czyli organizacji zrzeszających jeźdźców startujących w danej dziedzinie sportów konnych. Aplikacja nadaje się dla klubów, których zawodnicy startują w takich dziedzinach jak:

- Skoki przez przeszkody,
- WKKW (skrót od "Wszechstronny konkurs konia wierzchowego"),
- Ujeżdżenie.

W celu jak najlepszego określenia wymagań funkcjonalnych, przed napisaniem aplikacji przeprowadzono rozmowy z kilkoma osobami zaangażowanymi w to środowisko: pracownikami stadnin państwowych, właścicielami klubów, jak także z osobami prywatnie trzymającymi konie w stadninach. Po przeprowadzonych rozmowach zdecydowano się na dwie wersje aplikacji: desktopową oraz mobilną, które będą różnić się funkcjonalnościami.

Aplikacja ma na celu pomóc w gromadzeniu informacji o jeźdźcach przynależących do klubu oraz ich koniach. W aplikacji gromadzone są informacje o codziennych aktywnościach koni, ich chorobach, żywieniu oraz zawodach w których biorą udział. Naturalnie chcemy także zapisywać wyniki z tych zawodów, aby móc określić czy dany trening jest skuteczny. Z aplikacji będą korzystać zawodnicy, trenerzy, jacy i zarząd klubu.

Aby skutecznie zbierać informacje o treningach i innych aktywnościach niezbędna jest aplikacja mobilna, ponieważ dane te muszą być wprowadzane na bieżąco. Informacje o wizytach różnorodnych lekarzy oraz kowala także muszą być zapisywane na bieżąco podczas danej wizyty. Dlatego funkcjonalności te dotyczą jedynie aplikacji mobilnej. W aplikacji mobilnej można również sprawdzić aktualny plan żywienia swojego konia. Do tej aplikacji będą mieć dostęp jedynie osoby posiadające konie.

W aplikacji desktopowej wyświetlane są statystyki aktywności koni danego użytkownika jak i szczegóły wizyt lekarzy i kowali. W tej aplikacji można zaplanować wyjazdy na zawody jak także szczegółowe plany żywienia swoich podopiecznych. W tej aplikacji tworzone będą także konta użytkowników, oraz ich koni. Dostęp do funkcji tworzenia kont będzie ograniczony i posiadać go będzie jedynie administrator aplikacji.



Każdy członek klubu będzie miał swoje konto z możliwością logowania zarówno do aplikacji mobilnej jak i desktopowej. Trenerzy, właściciele klubu i inne osoby związane z klubem będą miały dostęp jedynie do aplikacji desktopowej.

## 4.2 Wymagania funkcjonalne

Funkcjonalności aplikacji mobilnej oraz desktopowej nie są takie same mimo iż są połączone do jednej bazy, więc czerpią z tego samego źródła informacji. Pomimo znaczących różnic niektóre funkcjonalności pokrywają się w obu tych produktach. Wymagania funkcjonalne, które muszą spełniać obie aplikacje przedstawia tabela 4.1.

Wymaganie	Aktor	Opis wymagania
Logowanie do aplikacji	Trener, Członek klubu, Zarząd klubu	System pozwala na zalogowanie się po podaniu poprawnego loginu oraz hasła.
Resetowanie hasła przez email	Trener, Członek klubu, Zarząd klubu	System umożliwia resetowanie hasła przez adres e-mail.

Tabela 4.1: Wymagania funkcjonalne obu aplikacji

Aplikacja mobilna będzie służyć użytkownikom głównie do zapisu aktualnych wydarzeń z życia stajni. Jej głównym celem jest szybkie zapisanie informacji o aktywnościach koni i ich wizytach u lekarzy, bądź kowali. Można w niej także szybko sprawdzić przygotowany plan żywienia, oraz daty zbliżających się zawodów. Wymagania funkcjonalne dla aplikacji mobilnej zawierają poniższe tabele 4.2 i 4.3.

Wymaganie		Aktor	Opis wymagania
Zarządzanie aktywnościami	Dodawanie aktywności	Członek klubu	System umożliwia zapis danych wprowadzonych przez zalogowanego użytkownika do bazy danych.
Zarządzanie aktywnościami	Edytowanie aktywności	Członek klubu	System umożliwia edytowanie dodanych wcześniej danych o aktywnościach.
Zarządzanie aktywnościami	Usuwanie aktywności	Członek klubu	System pozwala na usuwanie dodanych wcześniej aktywności.
Zarządzanie aktywnościami	Wyświetlanie aktywności	Trener, Członek klubu, Zarząd klubu	System umożliwia na przeglądanie wszystkich danych o aktywnościach danego konia zgromadzonych w bazie danych.
Zarządzanie wizytami	Dodawanie wizyt	Członek klubu	System pozwala na zapisanie danych z wizyty konia u lekarza/kowala do bazy danych.
Zarządzanie wizytami	Edytowanie wizyt	Członek klubu	System powinien umożliwić zapis zaktualizowanych danych o wizycie do bazy.
Zarządzanie wizytami	Usuwanie wizyt	Członek klubu	System powinien umożliwiać usuwanie danych o dodanych wcześniej wizytach.
Zarządzanie wizytami	Wyświetlanie wizyt	Trener, Członek klubu, Zarząd klubu	System powinien umożliwić przeglądanie danych o wizytach zgromadzonych w bazie.
Zarządzanie wizytami	Planowanie wizyt	Członek klubu	System powinien pozwalać użytkownikom na dodanie do bazy danych o następnej wizycie, czyli umożliwić zapis wizyt jedynie z datą i opisem.
Zarządzanie wizytami	Zapisywanie zdjęcia z wizyty	Członek klubu	System powinien pozwalać na zapisywanie zdjęć z wizyt.

Tabela 4.2: Wymagania funkcjonalne aplikacji mobilnej

Wymaganie		Aktor	Opis wymagania
Zarządzanie wizytami	Przypomnienia o wizytach	Członek klubu	System powinien wysłać powiadomienie o zbliżającej się wizycie
Zarządzanie żywieniem	Przeglądanie planów żywienia	Członek klubu	System powinien umożliwiać przeglądanie planów żywienia umieszczonych w bazie.
Zarządzanie żywieniem	Wybór planu żywienia	Członek klubu	System powinien umożliwiać wybór jednego z planów żywienia umieszczonych w bazie jako tego aktualnie używanego.
Zarządzanie zawodami	Wyświetlanie najbliższych zawodów	Członek klubu	System powinien umożliwić wyświetlanie dat najbliższych zawodów umieszczonych w bazie.
Zarządzanie zawodami	Potwierdzenie udziału w zawodach	Członek klubu	System powinien umożliwić użytkownikowi potwierdzenie swojego udziału w zawodach.
Udostępnianie koni		Członek klubu	System powinien umożliwić udostępnianie koni między użytkownikami.

Tabela 4.3: Wymagania funkcjonalne aplikacji mobilnej

Aplikacja desktop-owa przeznaczona jest zarówno dla użytkowników posiadających swoje konie jak i dla osób zarządzających klubem jeździeckim. W aplikacji desktop-owej posiadacze koni będą mogli obejrzeć zgromadzone informacje w przystępniejszej formie na dużym ekranie, stworzyć plan żywienia swojego konia, jak także przeanalizować statystyki swoich koni. Osoby zarządzające klubem będą miały możliwość dodawania nowych użytkowników i koni jak także sprawdzania statystyk wszystkich koni klubowych. Szczegółowe wymagania funkcjonalne dla aplikacji desktopowej zostały przedstawione w tabelach 4.4 oraz 4.5.

Wymaganie		Aktor	Opis wymagania
Zarządzanie planami żywienia	Tworzenie planów żywienia	Członek klubu	System umożliwia użytkownikowi stworzenie planu żywienia i zapisanie go do bazy.
Zarządzanie planami żywienia	Edytowanie planów żywienia	Członek klubu	System pozwala aktualizować stworzone wcześniej plany żywienia.
Zarządzanie planami żywienia	Usuwanie planów żywienia	Członek klubu	System umożliwia usuwanie danych o stworzonych wcześniej planach żywienia.
Zarządzanie końmi	Dodawanie koni	Zarząd klubu	System umożliwia wprowadzenie danych o koniach i dodanie ich do konkretnego użytkownika
Zarządzanie końmi	Usuwanie koni	Zarząd klubu	System umożliwia usuwanie koni
Zarządzanie końmi	Edytowanie koni	Zarząd klubu	System umożliwia edycję danych o koniach zgromadzonych już w bazie.
Zarządzanie użytkownikami	Dodawanie użytkowników	Zarząd klubu	System umożliwia dodawanie danych o użytkownikach i tworzenie ich kont.
Zarządzanie użytkownikami	Edytowanie użytkowników	Zarząd klubu	System umożliwia edytowanie danych użytkownika
Zarządzanie użytkownikami	Usuwanie użytkowników	Zarząd klubu	System umożliwia usuwanie użytkowników
Zarządzanie użytkownikami	Zmiana hasła	Zarząd klubu, Członek klubu, Trener	System umożliwia zmianę hasła przez użytkownika.

Tabela 4.4: Wymagania funkcjonalne aplikacji desktopowej

Wymaganie		Aktor	Opis wymagania
Zarządzanie zawodami	Dodawanie zawodów	Zarząd klubu	System pozwala na tworzenie zawodów, oraz zapraszanie do udziału w nich poszczególnych członków klubu
Zarządzanie zawodami	Edytowanie zawodów	Zarząd klubu	System pozwala na edycję danych o dodanych wcześniej zawodach
Zarządzanie zawodami	Usuwanie zawodów	Zarząd klubu	System pozwala na usuwanie danych o dodanych wcześniej zawodach.
Przeglądanie historii wizyt		Członek klubu, Trener, Zarząd klubu	
Przeglądanie statystyk		Członek klubu, Trener, Zarząd klubu	

Tabela 4.5: Wymagania funkcjonalne aplikacji desktopowej

## Przypadki użycia

Wszystkie wymagania funkcjonalne zgromadzone w powyższych tabelach, możemy przedstawić na diagramie przypadków użycia.

UML-

Przypadek użycia - jest to

Diagram przypadków użycia -

## 4.3 Wymagania нефункциональные

Wymagania нефункциональные:

# Rozdział 5

## Baza danych

### 5.1 Model konceptualny

Proces tworzenia bazy danych zaczynamy od modelu konceptualnego. W pierwszej fazie tworzenia go ważne jest określenie słownika pojęć, które będą następnie używane w projekcie bazy danych.

#### Słownik pojęć

- **Użytkownik** - wszyscy członkowie klubu, trenerzy oraz zarząd klubu.
- **Koń** - koń należący do któregoś z członków klubu jeździeckiego, lub dzierżawiony przez niego.
- **Atywności** - są to czynności wykonywane przez konia w ciągu dnia, należą do nich jazdy, skoki przez przeszkody, kross, ujeżdżenie, lonża, wyjazd w teren, karuzela, padok, wyjazd na zawody, spacer, skoki luzem, padok.
- **Wizyty** - to wizyty wszelkich lekarzy, jak także wizyty kowali.
- **Udostępnianie konia** - jest to przekazanie możliwości wprowadzania danych o danym koniu przez jego właściciela innemu członkowi klubu.

Po określeniu definicji poszczególnych pojęć używanych w projekcie możemy przystąpić do tworzenia kategorii.

#### Kategorie

Po przeanalizowaniu wycinku rzeczywistości możemy określić jakie dane chcemy zbierać i zapisywać do bazy danych. Dane te możemy podzielić na kategorie i opisać językiem naturalnym ich cechy charakterystyczne.

#### KAT:1

Po określeniu kategorii możemy określić reguły funkcjonowania naszej aplikacji.

## **Reguły funkcjonowania**

Reguły funkcjonowania określają zasady, procedury i wytyczne jakie musi spełniać projektowana aplikacja.

- REG\001** Konta użytkowników tworzy jedynie użytkownik "administrator".
- REG\002** Każdy użytkownik ma określony swój typ.
- REG\003** Każdy użytkownik może zmienić swoje hasło.
- REG\004** O każdym użytkowniku, jak także o lekarzu i kowalu zbieramy podstawowe dane personalne.
- REG\005** Tylko użytkownik "administrator" dodaje konie do kont użytkowników.
- REG\006** Każdy koń ma przypisaną płęć.
- REG\007** Każdy koń ma przypisany status.
- REG\008** Jeden użytkownik może posiadać wiele koni.
- REG\009** Aktywności konia może dodać jego właściciel lub osoba której właściciel udostępni konia.
- REG\010** Koń może mieć wiele aktywności każdego dnia.
- REG\011** Wizyty konia może dodawać tylko jego właściciel.
- REG\012** Na wizycie jest jeden koń i jedno lekarz/kowal.
- REG\013** Każdy lekarz ma określoną specjalizację.
- REG\014** Plan żywienia konia może ustalać tylko właściciel.
- REG\015** Koń może posiadać wiele planów żywienia, ale aktualnie może jeść tylko jeden.
- REG\016** Plan żywienia zawiera wiele żywien.
- REG\017** Żywienie dotyczy konkretnego typu jedzenia, podawanego o konkretnej porze (rano, południe, wieczór), który swoją jednostkę miary.
- REG\018** Użytkownicy, którym ktoś udostępnił konia mogą tylko wyświetlić plan żywienia.
- REG\019** Statystyki mają być tworzone na podstawie aktywności.
- REG\020** Użytkownik "członek klubu" może przeglądać statystyki tylko swoich koni.
- REG\021** Użytkownik "trener" lub "administrator" może przeglądać statystyki wszystkich koni.
- REG\022** Użytkownik "trener" lub "administrator" może dodawać wyjazd na zawody dla całego klubu i zapraszać poszczególnych użytkowników.
- REG\023** Użytkownik "członek klubu" może dodawać swoje wyjazdy na zawody.



## Ograniczenia dziedzinowe

Ograniczenia dziedzinowe to ograniczenia, które nakładane są na atrybuty w powyższych kategoriach. Wynikają one z analizy wycinka rzeczywistości i należy je uwzględnić podczas projektowania bazy danych oraz implementacji systemu.

**OGR\001** Paszport konia składa się ze znaków i cyfr postaci xxx-aaa-bb-cccc-dd, gdzie

- xxx - określa kraj pochodzenia konia,
- aaa - oznacza kod hodowli konia,
- bb- oznacza rok urodzenia konia,
- ccccc - to numer paszportu konia,
- dd - to numer identyfikacyjny konia w ramach hodowli.

**OGR\002** Data wizyty konia jest wcześniejsza niż data jego urodzenia.

**OGR\003**

## Transakcje

Transakcje są to operacje, które możemy wykonywać na danych. Mają one cztery własności, które w skrócie nazywamy ACID (ang. Atomicity, Consistency, Isolation, Durability). Transakcje mają więc następujące własności:

- atomowość, inaczej niepodzielność oznacza, że transakcje muszą być wykonywane na bazie w całości. Jeśli transakcja nie zostanie poprawnie przeprowadzona należy przywrócić stan bazy z przed jej wykonania.
- spójność, po wykonaniu transakcji baza powinna być nadal spójna.
- izolacja, oznacza że transakcje nie mogą być od siebie zależne.
- trwałość, oznacza że dane po transakcji zostają zapisane w bazie i są zachowane na stałe.

Transakcje występujące w aplikacji:

### **TRA\001 Dodanie aktywności**

*Opis:* Zadaniem transakcji jest dodanie danych o aktywności konia. Aktywności konia może dodać jedynie członek klubu, który jest jego właścicielem lub osoba której został on udostępniony.

*Uwarunkowania:* Aktywność musi zawierać dane o tym kto ją wprowadził, jakiego konia ona dotyczy, w jakim dniu została wykonana, oraz czas jej trwania.

*Wejście:*

- U - Dane nowej aktywności
- BD - Dane aktywności

*Wyjście:*

- U - Komunikat
- BD - Dane aktywności

## **TRA\002 Edycja aktywności**

*Opis:*

*Uwarunkowania:*

*Wejście:*

- U -
- BD -

*Wyjście:*

- U -
- BD -

## Encje

Po określeniu kategorii, reguł funkcjonowania, ograniczeń dziedzinowych i transakcji należy przystąpić do tworzenia encji i relacji między nimi. Definicje??

### ENC\001 Activity

*Semantyka encji* - Encja zawierająca aktywności konia.

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
activityID	Numer identyfikujący aktywności	Liczba naturalna	+
date	Data aktywności	Data	+
description	Opis aktywności	typ znakowy	-
time	Czas trwania aktywności	Czas	+
intensivity	Intensywność aktywności	Liczba naturalna	+
satisfaction	Satysfakcja aktywności	Liczba naturalna	+
activityType	Typ aktywności	Liczba naturalna	+

Tabela 5.1: Wykaz atrybutów encji typu Activity

Klucze kandydujące: activityID

Klucz główny: activityID

Charakter encji: encja silna

### ENC\002 Competition

*Semantyka encji* - encja zawierająca dane o zawodach.

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
competitionID	Numer identyfikujący zawody	Liczba naturalna	+
spot	Miejsce wizyty	max. znaków 50	-
description	Opis zawodów	Typ znakowy	-
rank	Ranga zawodów	max. znaków 50	-

Tabela 5.2: Wykaz atrybutów encji typu Competition

Klucze kandydujące: competitionID  
 Klucz główny: competitionID  
 Charakter encji: encja silna

#### ENC\003 CustomNotification

*Semantyka encji* - Encja zawierająca powiadomienia.

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
notificationID	Numer identyfikujący powiadomienie	Liczba naturalna	+
title	Tytuł powiadomienia	max. znaków 50	+
description	Data kończąca udostępnienie	typ znakowy	+
sendDate	Data wysłania	Data	+
createdDate	Data stworzenia	Data	+

Tabela 5.3: Wykaz atrybutów encji typu CustomNotification

Klucze kandydujące: notificationID  
 Klucz główny: notificationID  
 Charakter encji: encja silna

#### ENC\004 Doctor

*Semantyka encji* - encja opisująca lekarzy łącząca ich dane kontaktowe ze specjalizacją.

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
doctorID	Numer identyfikujący doktora	Liczba naturalna	+

Tabela 5.4: Wykaz atrybutów encji typu Doctor

Klucze kandydujące: doctorID  
 Klucz główny: doctorID  
 Charakter encji: encja silna

#### ENC\005 DoctorSpecialisation

*Semantyka encji* - encja słownikowa zawiera nazwy specjalizacji lekarzy jak także kowali.

Klucze kandydujące: specializationID

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
specialisationID	Numer identyfikujący specjalizacje	Liczba naturalna	+
name	Nazwa specjalizacji	max. znaków 50	+

Tabela 5.5: Wykaz atrybutów encji typu DoctorSpecialization

Klucz główny: specializationID

Charakter encji: encja silna

## ENC\006 Eat

*Semantyka encji* - Encja zawierająca informacje o planach żywienia i ich przynależności do koni.

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
eatID	Numer identyfikujący jedzenie	Liczba naturalna	+
isActive	Czy obecny plan jest w użyciu?		+

Tabela 5.6: Wykaz atrybutów encji typu Feed

Klucze kandydujące: eatID

Klucz główny: eatID

Charakter encji: encja silna

## ENC\007 Feed

*Semantyka encji* - Encja zawierająca informacje o jedzeniu koni.

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
feedID	Numer identyfikujący jedzenie	Liczba naturalna	+
amount	Ilość jedzenia w porcji	Liczba zmiennoprzecinkowa	+

Tabela 5.7: Wykaz atrybutów encji typu Feed

Klucze kandydujące: feedID  
 Klucz główny: feedID  
 Charakter encji: encja silna

## ENC\008 Forage

*Semantyka encji* - Encja zawierająca informacje o paszy dla koni.

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
forageID	Numer identyfikujący paszy	Liczba naturalna	+
name	Nazwa paszy	max. znaków 50	+
producent		max. znaków 50	-
capacity	Ilość paszy w jednym worku	Liczba naturalna	-

Tabela 5.8: Wykaz atrybutów encji typu Meal

Klucze kandydujące: forageID  
 Klucz główny: forageID  
 Charakter encji: encja silna

## ENC\009 Horse

*Semantyka encji* - Encja zawierająca informacje o koniach.

Klucze kandydujące: horseID  
 Klucz główny: horseID  
 Charakter encji: encja silna

## ENC\010 HorseGender

*Semantyka encji* - Encja słownikowa zawierająca płeć koni.

Klucze kandydujące: genderID  
 Klucz główny: genderID  
 Charakter encji: encja silna

## ENC\011 HorseStatus

*Semantyka encji* - Encja słownikowa zawierająca statusy koni.

Klucze kandydujące: statusID  
 Klucz główny: statusID  
 Charakter encji: encja silna

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
horseID	Numer identyfikujący konia	Liczba naturalna	+
name	Imie konia	max. znaków 50	+
mother	Imie klaczy	max. znaków 50	+
father	Imie ogiera	Max. znaków 50	-
birthday	Data urodzenia konia	Datetime	-
race	Rasa konia	Max. znaków 50	-
breeder	Hodowca koni	Max. znaków 50	-
passport	Paszport konia	Max. znaków 50	-
photo	Zdjęcie konia	Typ znakowy	-

Tabela 5.9: Wykaz atrybutów encji typu Horse

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
genderID	Numer identyfikujący płć konia	Liczba naturalna	+
gender	Nazwa płci konia	max. znaków 50	+

Tabela 5.10: Wykaz atrybutów encji typu HorseGender

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
statusID	Numer identyfikujący status konia	Liczba naturalna	+
name	Nazwa statusu konia	max. znaków 50	+

Tabela 5.11: Wykaz atrybutów encji typu HorseStatus

## ENC\012 Meal

*Semantyka encji* - Encja słownikowa zawierająca nazwy posiłków.

Klucze kandydujące: mealID

Klucz główny: mealID

Charakter encji: encja silna

## ENC\013 NutritionPlan

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
mealID	Numer identyfikujący posiłek	Liczba naturalna	+
mealName	Nazwa posiłku	max. znaków 50	+

Tabela 5.12: Wykaz atrybutów encji typu Meal

*Semantyka encji* - Encja zawierająca informacje o planie żywienia koni.

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
nutritionPlanID	Numer identyfikujący plan żywienia	Liczba naturalna	+
title	Tytuł planu żywienia	max. znaków 50	+
description	Ilość jedzenia w porcji	Liczba zmiennoprzecinkowa	-
icon	Ikona dołączona do planu żywienia	Liczba naturalna	+

Tabela 5.13: Wykaz atrybutów encji typu Feed

Klucze kandydujące: nutritionPlanID

Klucz główny: nutritionPlanID

Charakter encji: encja silna

## ENC\014 PeopleDetails

*Semantyka encji* - encja zawiera szczegółowe dane użytkowników (członków klubu, trenerów i zarządu klubu) jak i lekarzy oraz kowali.

Klucze kandydujące: detailsID

Klucz główny: detailsID

Charakter encji: encja słaba

## ENC\015 TakePart

*Semantyka encji* - encja zawierająca dane o zawodach.

Klucze kandydujące: competitionID

Klucz główny: competitionID

Charakter encji: encja silna

## ENC\016 Shared



Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
detailsID	Numer identyfikujący dane użytkowników	Liczba naturalna	+
name	Imię	max. znaków 50	-
surname	Nazwisko	max. znaków 50	+
phonNumber	Numer telefonu	max. znaków 50	-
email	Adres e-mailowy	max. znaków 50	-
city	Miasto zamieszkania	max. znaków 50	-
street	Ulica zamieszkania	max. znaków 50	-
number	Numer domu zamieszkania	max. znaków 50	-

Tabela 5.14: Wykaz atrybutów encji typu PeopleDetails

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
takePartID	Numer identyfikujący udział w zawodach	Liczba naturalna	+
level	Poziom konkursu	max. znaków 50	+
result	Wynik zawodów	Typ znakowy	+
place	Zajęte miejsce	Liczba naturalna	+

Tabela 5.15: Wykaz atrybutów encji typu Competition

*Semantyka encji* - Encja zawierająca wpisy o udostępnianiu koni.

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
sharedID	Numer identyfikujący status konia	Liczba naturalna	+
code	Kod z kodu QR	max. znaków 50	+
endDate	Data kończąca udostępnienie	max. znaków 50	+
startDate	Data udostępnienia	max. znaków 50	+

Tabela 5.16: Wykaz atrybutów encji typu Shared

Klucze kandydujące: sharedID

Klucz główny: sharedID

Charakter encji: encja silna

### ENC\017 UnitOfMeasure

*Semantyka encji* - Encja słownikowa zawierająca nazwy jednostek miary.

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
unitID	Numer identyfikujący jednostkę miary	Liczba naturalna	+
unitName	Nazwa jednostek miary	max. znaków 50	+

Tabela 5.17: Wykaz atrybutów encji typu Meal

Klucze kandydujące: unitID

Klucz główny: unitID

Charakter encji: encja silna

### ENC\018 UserAccount

*Semantyka encji* - encja zawiera dane użytkownika (członków klubu, trenerów i zarządu klubu). Klucze kandydujące: userID

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
userID	Numer identyfikujący użytkownika	Liczba naturalna	+
accountLogin	Login użytkownika	max. znaków 50	+
hash	???	max. znaków 50	+
salt	???	max. znaków 50	+
createdDateTime	Data utworzenia konta	Data	+

Tabela 5.18: Wykaz atrybutów encji typu UserAccount

Klucz główny: userID

Charakter encji: encja słaba

### ENC\019 UserType

*Semantyka encji* - encja zawiera typy użytkowników: zwykły użytkownik (standard), trener (trainer), zarząd klubu (admin). Klucze kandydujące: userTypeID

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
userId	Numer identyfikujący typ użytkownika	Liczba naturalna	+
typeName	Nazwa typu użytkownika	max. znaków 50	+

Tabela 5.19: Wykaz atrybutów encji typu UserType

Klucz główny: userId  
 Charakter encji: encja silna

## ENC\020 Visit

*Semantyka encji* - encja zawierająca dane o wizytach.

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
careID	Numer identyfikujący wizyte	Liczba naturalna	+
cost	Cena wizyty	Liczba rzeczywista dodatnia	+
summary	Opis podsumowujący wizytę	Typ znakowy	-
artefactImage	Zdjęcie z wizyty	Typ znakowy	-
visitDate	Data wizyty	Data	+

Tabela 5.20: Wykaz atrybutów encji typu Visit

Klucze kandydujące: careID  
 Klucz główny: careID  
 Charakter encji: encja silna

Po zaprojektowaniu encji możemy zapisać predykatowe definicje typów encji.

## ENC\001

### 5.2 Model logiczny

### 5.3 Model fizyczny

# Rozdział 6

## Projekt systemu

### 6.1 Model projektowanego systemu

Diagramy stanów

Diagramy aktywności

Diagram klas

Architektura aplikacji

Jaka baza jakie połączenie itp.

Wykorzystane wzorce projektowe

Model architektoniczny MVVM

### 6.2 Wybrane aspekty implementacyjne

jeden viewmodel obsługuje dwa widoki (dodawanie aktywności i szczegóły aktywności)  
kontrolki

# Rozdział 7

## Testy aplikacji

7.1 Unit testy

7.2 Test case

7.3 Baza błędów

## Rozdział 8

# Dokumentacja użytkownika

8.1 Aplikacja desktopowa

8.2 Aplikacja mobilna

## Rozdział 9

## Podsumowanie

# Bibliografia

- [1] Hanna Mazur, Zygmunt Mazur, *Projektowanie relacyjnych baz danych*. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2004.



## Spis rysunków

## Rozdział 10

### Spis listingów

# Spis tabel

4.1	Wymagania funkcjonalne obu aplikacji . . . . .	5
4.2	Wymagania funkcjonalne aplikacji mobilnej . . . . .	6
4.3	Wymagania funkcjonalne aplikacji mobilnej . . . . .	7
4.4	Wymagania funkcjonalne aplikacji desktopowej . . . . .	9
4.5	Wymagania funkcjonalne aplikacji desktopowej . . . . .	10
5.1	Wykaz atrybutów encji typu Activity . . . . .	15
5.2	Wykaz atrybutów encji typu Competition . . . . .	15
5.3	Wykaz atrybutów encji typu CustomNotification . . . . .	16
5.4	Wykaz atrybutów encji typu Doctor . . . . .	16
5.5	Wykaz atrybutów encji typu DoctorSpecialization . . . . .	17
5.6	Wykaz atrybutów encji typu Feed . . . . .	17
5.7	Wykaz atrybutów encji typu Feed . . . . .	17
5.8	Wykaz atrybutów encji typu Meal . . . . .	18
5.9	Wykaz atrybutów encji typu Horse . . . . .	19
5.10	Wykaz atrybutów encji typu HorseGender . . . . .	19
5.11	Wykaz atrybutów encji typu HorseStatus . . . . .	19
5.12	Wykaz atrybutów encji typu Meal . . . . .	20
5.13	Wykaz atrybutów encji typu Feed . . . . .	20
5.14	Wykaz atrybutów encji typu PeopleDetails . . . . .	21
5.15	Wykaz atrybutów encji typu Competition . . . . .	21
5.16	Wykaz atrybutów encji typu Shared . . . . .	21
5.17	Wykaz atrybutów encji typu Meal . . . . .	22
5.18	Wykaz atrybutów encji typu UserAccount . . . . .	22
5.19	Wykaz atrybutów encji typu UserType . . . . .	23
5.20	Wykaz atrybutów encji typu Visit . . . . .	23

## Rozdział 11

### Opis zawartości APD