

**UNIWERSYTET OPOLSKI**

**WYDZIAŁ MATEMATYKI, FIZYKI I INFORMATYKI**

**INSTYTUT INFORMATYKI**

**PRACA INŻYNIERSKA**

**Natalia Szymczak**

**Aplikacja bazodanowa dla klubów jeździeckich**

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 Visual Studio 2022 . . . . .	3
3.2	Microsoft SQL Server 2019 Express . . . . .	3
3.3	Microsoft SQL Server Management Studio . . . . .	3
3.4	Structured Query Language . . . . .	3
3.5	Windows Presentation Foundation . . . . .	4
3.6	Xamarin . . . . .	4
3.6.1	Xamarin.Android . . . . .	5
3.7	Android Device Manager . . . . .	6
3.8	NuGet . . . . .	6
3.9	Git . . . . .	7
3.10	Figma . . . . .	7
3.11	Entity Framework . . . . .	8
3.12	Ten do wykresów . . . . .	9
3.13	MVVM Toolkit . . . . .	9
3.14	Xamarin Community Toolkit . . . . .	9
3.15	ZXing.Net.Mobile.Forms . . . . .	9
3.16	Microsoft.Extensions.DependencyInjection . . . . .	9
<b>4</b>	<b>Specyfikacja wymagań</b>	<b>10</b>
4.1	Opis wycinka rzeczywistości . . . . .	10
4.2	Wymagania funkcjonalne . . . . .	11

4.3	Wymagania нефункционалне . . . . .	18
<b>5</b>	<b>Baza danych</b>	<b>22</b>
5.1	Model konceptualny . . . . .	22
5.2	Model logiczny . . . . .	46
5.3	Model fizyczny . . . . .	65
<b>6</b>	<b>Projekt systemu</b>	<b>66</b>
6.1	Model projektowanego systemu . . . . .	66
6.2	Wybrane aspekty implementacyjne . . . . .	66
<b>7</b>	<b>Testy aplikacji</b>	<b>67</b>
7.1	Unit testy . . . . .	67
7.2	Test case . . . . .	67
7.3	Baza błędów . . . . .	67
<b>8</b>	<b>Dokumentacja użytkownika</b>	<b>68</b>
8.1	Aplikacja desktopowa . . . . .	68
8.2	Aplikacja mobilna . . . . .	69
<b>9</b>	<b>Podsumowanie</b>	<b>75</b>
<b>10</b>	<b>Opis zawartości APD</b>	<b>82</b>

# Rozdział 1

## Wstęp

## Rozdział 2

### Przegląd istniejących rozwiązań

# Rozdział 3

## Technologie użyte w pracy

### 3.1 Microsoft Visual Studio 2022

Microsoft Visual Studio to środowisko IDE, za pomocą którego można edytować, debugować jak także kompilować kod. Po stworzeniu aplikacji można ją także opublikować w prostym środowisku. Środowisko to zawiera wiele funkcji wzbogacających proces tworzenia takich jak narzędzia uzupełniania kodu (Intellisense). Dzięki temu środowisku możemy programować aplikacje na dowolną platformę oraz dowolne urządzenia.

### 3.2 Microsoft SQL Server 2019 Express

Microsoft SQL Server jest to system, wspomagający zarządzanie bazą danych stworzony oraz utrzymywany przez firmę Microsoft. MS SQL wykorzystuje język zapytań Transact-SQL, który jest rozwinięciem standardu języka zapytań ANSI/SQL.

### 3.3 Microsoft SQL Server Management Studio

### 3.4 Structured Query Language

SQL czyli Structured Query Language jest to język zapytań wykorzystywany w relacyjnych bazach danych. Umożliwia on tworzenie, modyfikowanie oraz zarządzanie bazami danych. Dodatkowo dzięki SQL jesteśmy w stanie pobierać, dodawać, aktualizować oraz usuwać dane znajdujące się w naszej bazie danych. SQL wspiera również tworzenie skomplikowanych

zapytań, dzięki czemu możemy wykonywać różne operacje na danych takie jak: filtrowanie, sortowanie, grupowanie oraz łączenie.

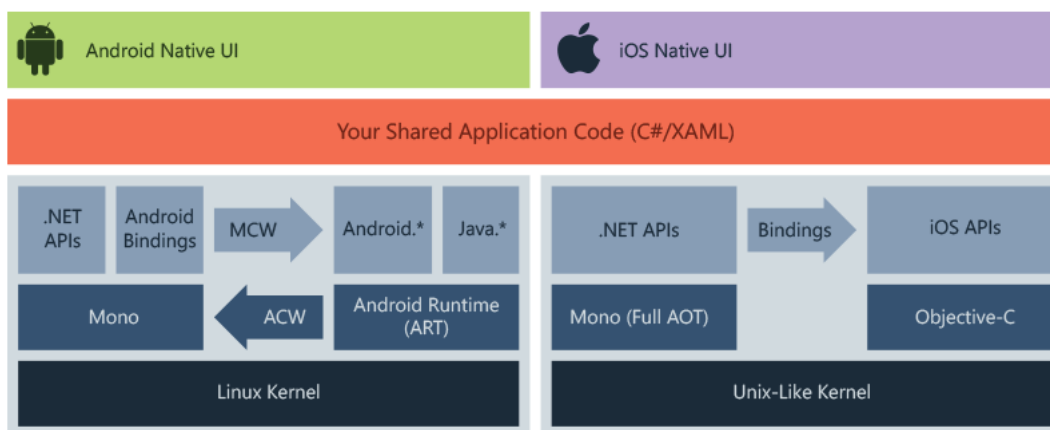
## 3.5 Windows Presentation Foundation

WPF - Windows Presentation Foundation, jest technologią opracowaną przez Microsoft. Służy ona do tworzenia aplikacji desktopowych na system Windows. Jest częścią .NET Framework i zapewnia on możliwość tworzenia zaawansowanych interfejsów użytkownika wykorzystując język XAML. Interfejs ten jest niezależny od rozdzielczości oraz używa aparatu renderowania opartego na wektorach, aby korzystać z nowoczesnego sprzętu graficznego. WPF dostarcza kontrolki, powiązanie danych, układ, grafiki 2D i 3D, animację, style, szablony, dokumenty, multimedia, tekst i typografię, jak także inne elementy interfejsu API platformy .NET. [8]

## 3.6 Xamarin

Xamarin jest to platforma do tworzenia aplikacji mobilnych za pomocą platformy .NET, która automatycznie obsługuje odzyskiwanie i alokowanie pamięci, jak także współdziałanie z platformami bazowymi. Dzięki Xamarinowi możemy pisać aplikacje na androida, iOS jak także na windows phone. Tworzy on warstwę abstrakcji komunikującą się za pomocą kodem aplikacji, a kodem bazowej platformy. Aplikacje wykorzystujące Xamarin możemy pisać nie tylko na komputerach PC z systemem Windows lub Linux, lecz także na urządzeniach z systemem MacOS. Architektura systemu Xamarin przedstawiona została na rysunku 3.1. Możemy na nim zobaczyć część architektury dotyczącą platformy android jak także, część dotyczącą platformy iOS.[2]



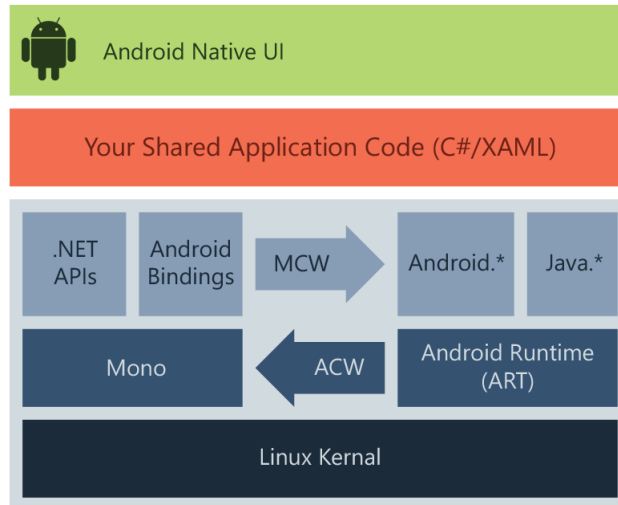


Rysunek 3.1: Architektura platformy Xamarin

*Źródło: [2]*

### 3.6.1 Xamarin.Android

Aplikacja HorseTracking dostępna będzie jedynie na platformę android. Aby dostosować ją do systemu iOS niezbędne było by urządzenie z systemem MacOS co nie było możliwe. Xamarin.Android kompilowany jest z języka C#, do języka pośredniego "just in time", często nazywanego JIT od pierwszych liter nazwy. JIT jest skompilowany do zestawu natywnego po uruchomieniu aplikacji. Xamarin.Android jest uruchamiany w środowisku mono obok maszyny wirtualnej środowiska Android Runtime. Dzięki platformie Xamarin możemy powiązać platformę .Net z przestrzeniami nazw Android.\* i Java.\*. Za pośrednictwem zarządzanych otok wywoływanych MCW środowisko mono może wywoływać przestrzenie nazw oraz udostępniać otoki wywoływane przez system Android(ACW). Dzięki temu oba środowiska mogą wywoływać kod nawzajem. Na rysunku 3.2 przedstawiona została architektura systemu Xamarin.Android[2].



Rysunek 3.2: Architektura platformy Xamarin.Android

*Źródło: [2]*

## 3.7 Android Device Menager

## 3.8 NuGet

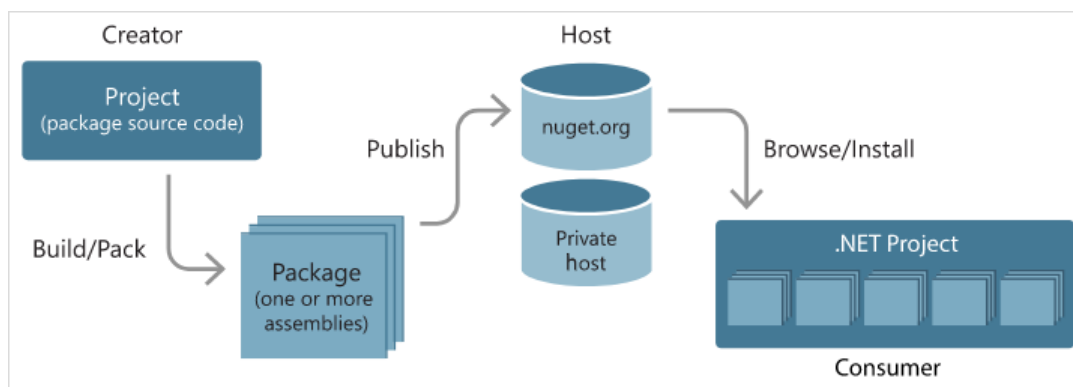
NuGet (logo systemu na rysunku 3.3) jest to mechanizm udostępniania kodu obsługiwany przez firmę Microsoft.



Rysunek 3.3: Logo systemu NuGet

*Źródło: [3]*

Służy on do współdzielenia kodu. Pakiet NuGet obsługuje hosty prywatne oraz publicznego hosta. Na hoście publicznym NuGet ma tysiące unikatowych pakietów dostępnych dla użytkowników .Net. Niezależnie od tego czy host jest prywatny czy publiczny jest on połączeniem między twórcami pakietów a ich konsumentami. Przepływ informacji między deweloperami pakietów, a ich konsumentami możemy zaobserwować na rysunku 3.4.



Rysunek 3.4: Przepływ informacji

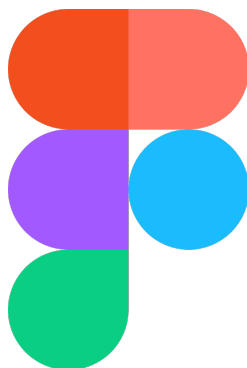
*Źródło: [3]*

## 3.9 Git

Git, github, sourcetree

## 3.10 Figma

Figma jest to narzędzie do projektowania i prototypowania interfejsów aplikacji. Dzięki narzędziom takim jak figma możemy zaplanować cały interfejs jeszcze przed jego implementacją. Stworzony w ten sposób interfejs możemy przetestować dzięki funkcji prototypowania. Jeszcze przed implementacją może on zostać udostępniony kilku osobą w celu sprawdzenia czy wszystkie funkcje aplikacji są dla użytkownika jasne i intuicyjne. Dzięki temu implementować będziemy interfejs już sprawdzony, więc będzie wymagał on mniej poprawek.

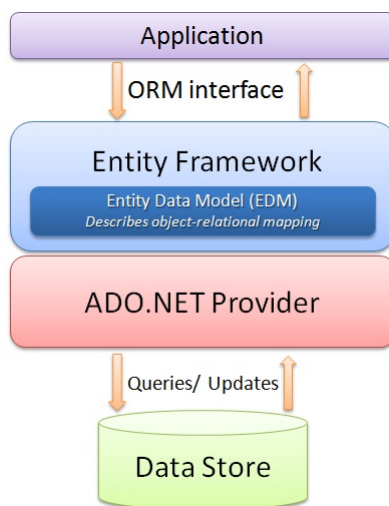


Rysunek 3.5: przepływ informacji

*Źródło: [7]*

### 3.11 Entity Framework

Entity Framework to narzędzie do mapowania obiektowo-relacyjnego, który umożliwia tworzenie przejrzystej, przenośnej i wysokopoziomowej warstwy dostępu do danych za pomocą platformy .NET (C#) dla wielu baz danych. Pozwala on na wykonywanie podstawowych operacji takich jak dodawanie, pobieranie, uaktualnianie i usuwanie danych. Możemy dzięki niemu także łatwiej zarządzać relacjami w bazie. Zasada działania tego narzędzia przedstawiona została na rysunku 3.6



Rysunek 3.6: przepływ informacji

*Źródło: [4]*

**3.12    Ten do wykresów**

**3.13    MVVM Toolkit**

**3.14    Xamarin Community Toolkit**

**3.15    ZXing.Net.Mobile.Forms**

**3.16    Microsoft.Extensions.DependencyInjection**

# 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 skierowana jest do 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 umożliwi 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 pozawalać użytkownikom na dodanie do bazy danych o następnej wizycie, czyli umożliwić zapis wizyt jedynie z datą i opisem.
Zarządzanie wizytami	Zapisywanie zdjęć 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.

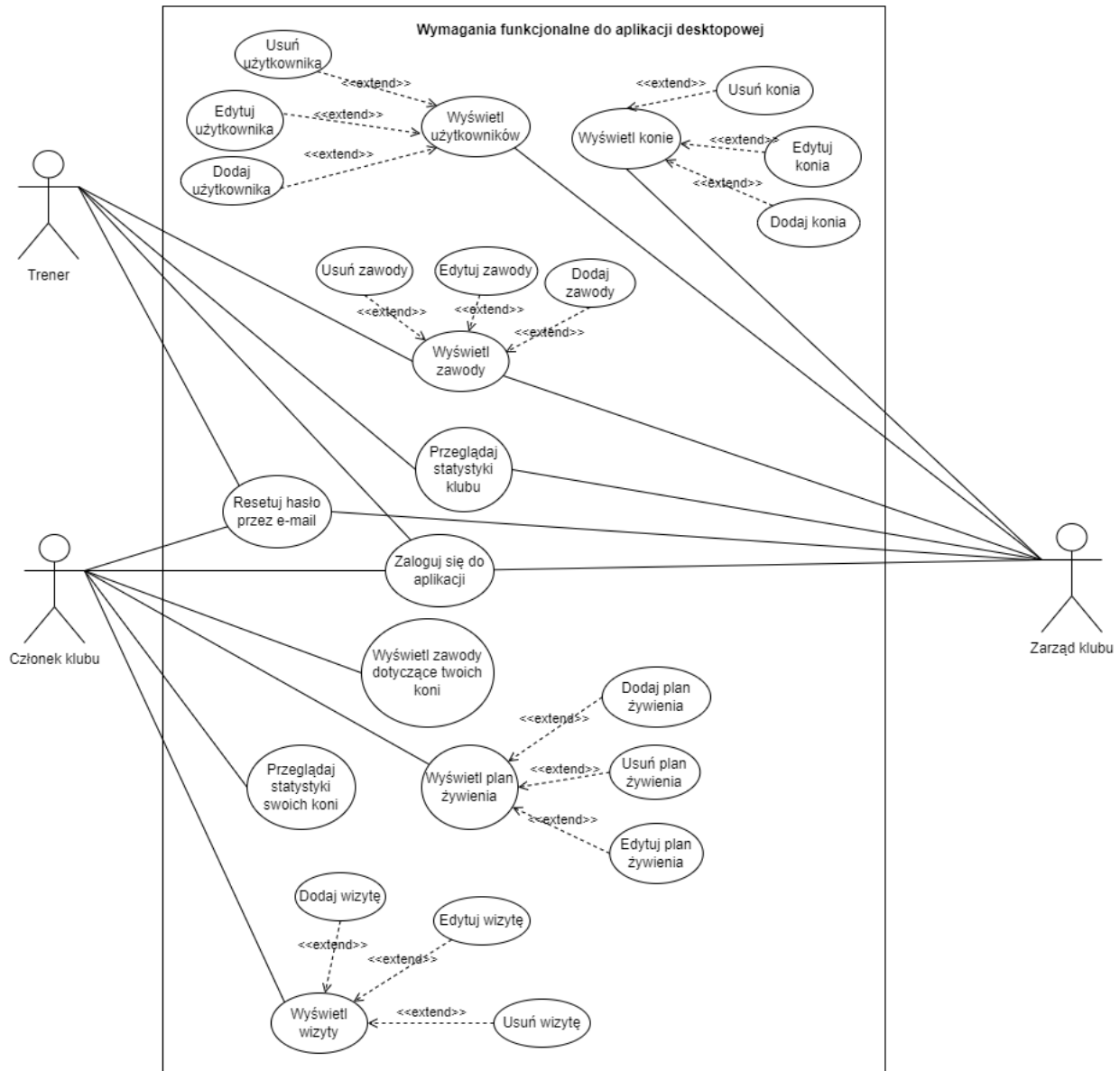
Tabela 4.4: Wymagania funkcjonalne aplikacji desktopowej

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

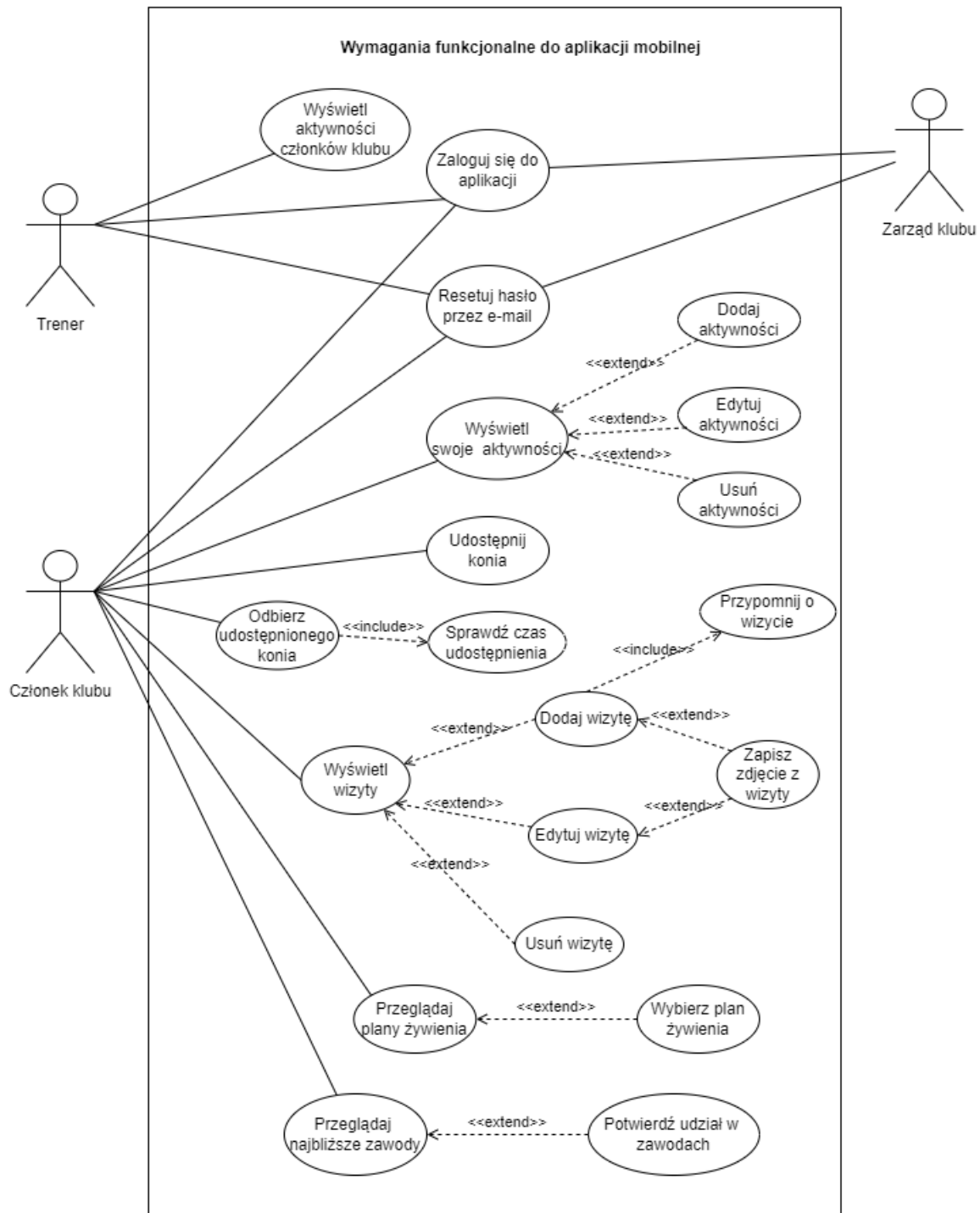
## Przypadki użycia

Wszystkie wymagania funkcjonalne zgromadzone w powyższych tabelach, możemy przedstawić na diagramie przypadków użycia UML. Poniższe rysunki zostały sporządzone według zasad języka UML, opisanych w pozycji [odnieść się do bibliografii]. Na rysunku 4.1 przedstawione zostały przypadki użycia aplikacji desktopowej, zaś na rysunku 4.2 przedstawione zostały przypadki użycia aplikacji mobilnej.



Rysunek 4.1: Diagram Use Case dla aplikacji desktopowej

*Źródło: Opracowanie własne*



Rysunek 4.2: Diagram Use Case dla aplikacji mobilnej

*Źródło: Opracowanie własne*

## 4.3 Wymagania niefunkcjonalne

W tym rozdziale przedstawione zostaną wymagania niefunkcjonalne projektowanego systemu. W osobnych tabelach przedstawione zostaną wymagania dla aplikacji mobilnej (tabela ??) oraz desktopowej (tabela 4.6). Przedstawione wymagania zostały opracowane zgodnie ze standardem ISO 9126. Określone zostały atrybuty, takie jak: niezawodność (Reliability), obsługiwalność (Usability), wydajność (Efficiency), łatwość konserwacji (Maintainability) i przenośność (Portability).

Tabela 4.6: Wymagania niefunkcjonalne aplikacji desktopowej

*Źródło: Opracowanie własne*

Nr	Nazwa wymagania	Opis wymagania
1	Interfejsy programowe	System operacyjny Microsoft Windows 7 lub nowszy. Baza danych zainstalowana na platformie Microsoft SQL Server 2019 lub nowszej oraz dostępna dla aplikacji zgodnie z zasadami określonymi w dokumentacji serwera.
2	Interfejsy sprzętowe	Komputer osobisty lub laptop, obsługujący system Windows 7 lub nowszy. Minimum 4 GB pamięci RAM i xGB wolnej przestrzeni na dysku, procesor 1GHz lub szybszy, 32 bitowy (x86) lub 64 bitowy (x64).
3	Obsługiwalność (Usability)	Aplikacja powinna mieć prosty i intuicyjny interfejs użytkownika. Interfejs powinien być dostosowany do pracy na monitorach o małej rozdzielczości i laptopach.
4	Niezawodność (Reliability)	Aplikacja powinna walidować wszystkie pola, do których wprowadzane są dane. W przypadku błędnych danych powinny wyświetlać się komunikaty informujące o nieprawidłowościach w klarowny sposób.
5	Język i narzędzia programowania	Aplikacja została napisana w C# na silniku graficznym Windows Presentation Foundation przy użyciu środowiska Microsoft Visual Studio Community 2022. Baza danych została napisana w języku SQL, przy pomocy środowiska Microsoft SQL Server Management Studio 18 i zainstalowana na serwerze bazy danych Microsoft SQL Server 2019

6	Aspekty prawne	W bazie danych przechowywane będą dane członków klubu, zarządu, specjalistów oraz których korzystają. Poza danymi osobowymi tych osób przechowywane będą także dane o koniach, ich aktywnościach oraz stanie zdrowia. Aplikacja nie będzie przekazywać danych osobowych poszczególnych członków innym użytkownikom. Dane specjalistów będą rozpowszechniane między użytkownikami, dlatego przed wpisaniem do bazy będą musieli wyrazić na to pisemną zgodę.
7	Wydażność (Efficiency)	Używanie aplikacji na sprzęcie o minimalnych wymaganiach powinno przebiegać w sposób płynny, to znaczy wyświetlanie, dodawanie, edytowanie, usuwanie, kategoryzowanie i filtrowanie danych powinno nie zajmować dłużej niż kilka sekund.
8	Łatwość konserwacji (Maintainability)	Działanie aplikacji będzie kontrolowane przy pomocy dziennika debugowania. Kolejne wersje systemu (kopie zapasowe) będą zapisywane za pomocą systemu kontroli wersji GIT.
9	Przenośność (Portability)	Aplikacja udostępniana będzie w pliku wykonywalnym .exe. Baza danych musi zostać zainstalowana i skonfigurowana na serwerze w danym klubie jeździeckim. Po zainstalowaniu aplikacja i baza danych powinny zostać skonfigurowane ze sobą.



Tabela 4.7: Wymagania niefunkcjonalne aplikacji mobilnej

*Źródło: Opracowanie własne*

Nr	Nazwa wymagania	Opis wymagania
1	Interfejsy programowe	Android ?? lub nowszy
2	Interfejsy sprzętowe	Telefon lub tablet z systemem operacyjnym Android ?? lub nowszym
3	Obsługiwalność (Usability)	Aplikacja powinna dobrze skalować się na różnej wielkości ekrany. Na telefonie najważniejsze przyciski powinny znajdować się "w zasięgu kciuka". Ikony i przyciski powinny być widoczne i możliwe do kliknięcia nawet małych ekranach (minimalny ekran ??), w aplikacji dostępny jest jedynie tryb jasny.
4	Niezawodność (Reliability)	Aplikacja powinna walidować wszystkie pola, do których wprowadzane są dane. W przypadku błędnych danych powinny wyświetlać się komunikaty informujące o nieprawidłowościach w klarowny sposób.
5	Język i narzędzia programowania	Aplikacja została napisana na platformie Xamarin przy użyciu środowiska Microsoft Visual Studio Community 2022. Używa tej samej bazy co aplikacja desktopowa.
6	Aspekty prawne	Aspekty prawne aplikacji mobilnej i desktopowej pokrywają się, ponieważ korzystają one z tych samych danych.
7	Wydajność (Efficiency)	Używanie aplikacji na sprzęcie o minimalnych wymaganiach powinno przebiegać w sposób płynny, to znaczy wyświetlanie, dodawanie, edytowanie, usuwanie danych powinno nie zajmować dłużej niż kilka sekund.
8	Łatwość konserwacji (Maintainability)	Konserwacja aplikacji mobilnej będzie przebiegać analogicznie do desktopowej.
9	Przenośność (Portability)	Aplikacja udostępniana w postaci pakietu instalacyjnego .apk oraz powinna spełniać wszystkie wymagania do umieszczenia jej w sklepie Google Play.

# Rozdział 5

## Baza danych

W tym rozdziale przedstawimy model konceptualny, logiczny oraz fizyczny bazy danych. Rozdział ten został opracowany na podstawie [1].

### 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 jedne 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.

### ENC\01 ACTIVITY

*Semantyka encji* - Encja zawierająca aktywności, które koń wykonuje w ciągu dnia. Każda aktywność oprócz typu, zawiera opis, czas trwania oraz ocenę satysfakcji i intensywności jej wykonania.

Opis atrybutów encji znajduje się w tabeli 5.1.

Tabela 5.1: Wykaz atrybutów encji typu Activity

*Źródło: Opracowanie własne*

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
<i>activityID</i>	Numer identyfikujący aktywności	Liczba naturalna	+
<i>date</i>	Data wykonania aktywności	Data	+
<i>description</i>	Opis aktywności	Typ znakowy	-
<i>time</i>	Czas trwania aktywności	Czas	+
<i>intensity</i>	Intensywność wykonanej aktywności	Liczba naturalna	+
<i>satisfaction</i>	Satysfakcja wykonanej aktywności	Liczba naturalna	+
<i>activityType</i>	Typ wykonanej aktywności	Liczba naturalna	+

Klucze kandydujące: activityID

Klucz główny: activityID

Charakter encji: encja słaba

### ENC\02 COMPETITION

*Semantyka encji* - encja zawierająca dane o zawodach jeździeckich.

Opis atrybutów znajduje się w tabeli 5.2.

Tabela 5.2: Wykaz atrybutów encji typu Competition

*Źródło: Opracowanie własne*

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
<i>competitionID</i>	Numer identyfikujący zawody.	Liczba naturalna	+
<i>spot</i>	Miejsce, w którym odbywają się zawody.	Max. znaków 200	-
<i>description</i>	Opis zawodów.	Typ znakowy	-
<i>rank</i>	Ranga zawodów.	Max. znaków 50	-

Klucze kandydujące: competitionID

Klucz główny: competitionID

Charakter encji: encja silna

## ENC\03 NOTIFICATION

*Semantyka encji* - Encja zawierająca powiadomienia utworzone przez użytkowników.

Opis atrybutów encji znajduje się w tabeli 5.3.

Tabela 5.3: Wykaz atrybutów encji typu Notification

*Źródło: Opracowanie własne*

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
<i>notificationID</i>	Numer identyfikujący powiadomienie	Liczba naturalna	+
<i>title</i>	Tytuł powiadomienia	Max. znaków 30	+
<i>description</i>	Opis powiadomienia	Typ znakowy	+
<i>sendDate</i>	Data wysłania	Data	+
<i>createdDate</i>	Data stworzenia	Data	+

Klucze kandydujące: notificationID

Klucz główny: notificationID

Charakter encji: encja słaba

## ENC\04 PROFESSIONALS

*Semantyka encji* - encja opisująca specjalistów przyjeżdżających do konia takich jak lekarze (np. gastrolog, kardiolog, lekarz ogólny), fizjoterapeuci, kowale itp.

Opis atrybutów encji znajduje się w tabeli 5.4.

Tabela 5.4: Wykaz atrybutów encji typu Doctor

*Źródło: Opracowanie własne*

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
<i>professionalsID</i>	Numer identyfikujący specjalistę	Liczba naturalna	+
<i>degree</i>	Stopień naukowy doktora	Liczba naturalna	+

Klucze kandydujące: professionalsID

Klucz główny: professionalsID

Charakter encji: encja słaba

## ENC\05 SPECIALISATION

*Semantyka encji* - encja słownikowa zawiera nazwy specjalizacji specjalistów.

Opis atrybutów encji znajduje się w tabeli 5.5.

Tabela 5.5: Wykaz atrybutów encji typu Specialization

*Źródło: Opracowanie własne*

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
<i>specialisationID</i>	Numer identyfikujący specjalizację	Liczba naturalna	+
<i>name</i>	Nazwa specjalizacji	Max. znaków 85	+

Klucze kandydujące: specializationID

Klucz główny: specializationID

Charakter encji: encja silna

## ENC\06 DIET

*Semantyka encji* - Encja zawierająca informacje o aktywnej diecie konia.

Opis atrybutów encji znajduje się w tabeli 5.6.

Tabela 5.6: Wykaz atrybutów encji typu Diet

*Źródło: Opracowanie własne*

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
<i>dietID</i>	Numer identyfikujący jedzenie	Liczba naturalna	+
<i>isActive</i>	Określenie czy plan jest w użyciu	Prawda/Fałsz	+

Klucze kandydujące: dietID

Klucz główny: dietID

Charakter encji: encja słaba

## ENC\07 PORTION

*Semantyka encji* - Encja zawierająca informacje o porcji jedzenia.

Opis atrybutów encji znajduje się w tabeli 5.6.

Tabela 5.7: Wykaz atrybutów encji typu Portion

*Źródło: Opracowanie własne*

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
<i>portionID</i>	Numer identyfikujący jedzenie	Liczba naturalna	+
<i>amount</i>	Ilość jedzenia w porcji	Liczba zmiennoprzecinkowa	+

Klucze kandydujące: portionID

Klucz główny: portionID

Charakter encji: encja słaba

## ENC\08 FORAGE

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

Opis atrybutów encji znajduje się w tabeli 5.8.

Tabela 5.8: Wykaz atrybutów encji typu Forage

*Źródło: Opracowanie własne*

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
<i>forageID</i>	Numer identyfikujący paszy	Liczba naturalna	+
<i>name</i>	Nazwa paszy	Max. znaków 92	+
<i>producent</i>	Producent paszy	max. znaków 57	-
<i>capacity</i>	Ilość paszy w jednym worku	Liczba naturalna	-

Klucze kandydujące: forageID

Klucz główny: forageID

Charakter encji: encja słaba

## ENC\09 HORSE

*Semantyka encji* - Encja zawierająca informacje o koniach.

Opis atrybutów encji znajduje się w tabeli 5.9.

Tabela 5.9: Wykaz atrybutów encji typu Horse

*Źródło: Opracowanie własne*

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
<i>horseID</i>	Numer identyfikujący konia	Liczba naturalna	+
<i>name</i>	Imie konia	max. znaków 60	+
<i>mother</i>	Imie klaczy	max. znaków 60	-
<i>father</i>	Imie ogiera	Max. znaków 60	-
<i>birthday</i>	Data urodzenia konia	Date	-
<i>race</i>	Rasa konia	Max. znaków 50	-
<i>breeder</i>	Hodowca koni	Max. znaków 60	-
<i>passport</i>	Paszport konia	Max. znaków 20	-
<i>photo</i>	Zdjęcie konia	Typ znakowy	-

Klucze kandydujące: horseID

Klucz główny: horseID

Charakter encji: encja słaba

**ENC\10 HORSEGENDER***Semantyka encji* - Encja słownikowa zawierająca płeć koni.

Opis atrybutów encji znajduje się w tabeli 5.10.

Tabela 5.10: Wykaz atrybutów encji typu HorseGender

*Źródło: Opracowanie własne*

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
<i>genderID</i>	Numer identyfikujący płeć konia	Liczba naturalna	+
<i>gender</i>	Nazwa płci konia	Max. znaków 10	+

Klucze kandydujące: genderID

Klucz główny: genderID

Charakter encji: encja silna

## ENC\11 STATUS

*Semantyka encji* - Encja słownikowa zawierająca statusy koni.

Opis atrybutów encji znajduje się w tabeli 5.11.

Tabela 5.11: Wykaz atrybutów encji typu Status

*Źródło: Opracowanie własne*

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
<i>statusID</i>	Numer identyfikujący status konia	Liczba naturalna	+
<i>name</i>	Nazwa statusu konia	Max. znaków 20	+

Klucze kandydujące: statusID

Klucz główny: statusID

Charakter encji: encja silna

## ENC\12 MEALNAME

*Semantyka encji* - Encja słownikowa zawierająca nazwy posiłków.

Opis atrybutów encji znajduje się w tabeli 5.12.

Tabela 5.12: Wykaz atrybutów encji typu MealName

*Źródło: Opracowanie własne*

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
<i>mealNameID</i>	Numer identyfikujący posiłek	Liczba naturalna	+
<i>mealName</i>	Nazwa posiłku	Max. znaków 20	+

Klucze kandydujące: mealNameID

Klucz główny: mealNameID

Charakter encji: encja słaba

## ENC\13 NUTRITIONPLAN

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

Opis atrybutów encji znajduje się w tabeli 5.13.

Tabela 5.13: Wykaz atrybutów encji typu NutrtrionPlan

*Źródło: Opracowanie własne*

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
<i>nutritionPlanID</i>	Numer identyfikujący plan żywienia	Liczba naturalna	+
<i>title</i>	Tytuł planu żywienia	Max. znaków 50	+
<i>descriptiption</i>	Ilość jedzenia w porcji	Typ znakowy	-
<i>icon</i>	Ikona dołączona do planu żywienia	Liczba naturalna	+

Klucze kandydujące: nutritionPlanID

Klucz główny: nutritionPlanID

Charakter encji: encja silna

## ENC\14 PEOPLEDETAILS

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

Opis atrybutów encji znajduje się w tabeli 5.14.



Tabela 5.14: Wykaz atrybutów encji typu PeopleDetails

*Źródło: Opracowanie własne*

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
<i>detailsID</i>	Numer identyfikujący dane użytkowników	Liczba naturalna	+
<i>name</i>	Imie	Max. znaków 40	-
<i>surname</i>	Nazwisko	Max. znaków 40	+
<i>phonNumber</i>	Numer telefonu	Max. znaków 20	-
<i>email</i>	Adres e-mailowy	Max. znaków 320	-
<i>city</i>	Miasto zamieszkania	Max. znaków 200	-
<i>street</i>	Ulica zamieszkania	Max. znaków 90	-
<i>number</i>	Numer domu zamieszkania	Max. znaków 10	-

Klucze kandydujące: detailsID

Klucz główny: detailsID

Charakter encji: encja silna

**ENC\15 PARTICIPATION***Semantyka encji* - encja zawierająca dane o uczestnictwie konia w zawodach.

Opis atrybutów encji znajduje się w tabeli 5.15.

Tabela 5.15: Wykaz atrybutów encji typu Participation

*Źródło: Opracowanie własne*

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
<i>participationID</i>	Numer identyfikujący udział w zawodach	Liczba naturalna	+
<i>level</i>	Poziom konkursu	Max. znaków 30	+
<i>result</i>	Wynik zawodów	Typ znakowy	+
<i>place</i>	Zajęte miejsce	Liczba naturalna	+

Klucze kandydujące: participationID

Klucz główny: participationID

Charakter encji: encja słaba

## ENC\16 SHARED

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

Opis atrybutów encji znajduje się w tabeli 5.16.

Tabela 5.16: Wykaz atrybutów encji typu Shared

*Źródło: Opracowanie własne*

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

Klucze kandydujące: sharedID

Klucz główny: sharedID

Charakter encji: encja słaba

## ENC\17 UNITOFMEASURE

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

Opis atrybutów encji znajduje się w tabeli 5.17.

Tabela 5.17: Wykaz atrybutów encji typu UnitOfMeasure

*Źródło: Opracowanie własne*

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
<i>unitID</i>	Numer identyfikujący jednostkę miary	Liczba naturalna	+
<i>unitName</i>	Nazwa jednostek miary	Max. znaków 30	+

Klucze kandydujące: unitID

Klucz główny: unitID

Charakter encji: encja silna

**ENC\18 UserAccount**

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

Opis atrybutów encji znajduje się w tabeli 5.18.

Tabela 5.18: Wykaz atrybutów encji typu UserAccount

*Źródło: Opracowanie własne*

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
<i>userID</i>	Numer identyfikujący użytkownika	Liczba naturalna	+
<i>login</i>	Login użytkownika	max. znaków 50	+
<i>hash</i>	Hash hasła użytkownika	max. znaków 50	+
<i>salt</i>	Salt hasła użytkownika	max. znaków 50	+
<i>createdDateTime</i>	Data utworzenia konta	Data	+

Klucze kandydujące: userID

Klucz główny: userID

Charakter encji: encja słaba

## ENC\19 USERTYPE

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

Opis atrybutów encji znajduje się w tabeli 5.19.

Tabela 5.19: Wykaz atrybutów encji typu UserType

*Źródło: Opracowanie własne*

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
<i>userTypeID</i>	Numer identyfikujący typ użytkownika	Liczba naturalna	+
<i>typeName</i>	Nazwa typu użytkownika	max. znaków 20	+

Klucze kandydujące: *userTypeID*

Klucz główny: *userTypeID*

Charakter encji: encja silna

## ENC\20 VISIT

*Semantyka encji* - encja zawierająca dane o wizytach.

Opis atrybutów encji znajduje się w tabeli 5.20.

Tabela 5.20: Wykaz atrybutów encji typu Visit

*Źródło: Opracowanie własne*

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

Klucze kandydujące: visitID

Klucz główny: visitID

Charakter encji: encja słaba

## ENC\21 MEAL

*Semantyka encji* - Encja słownikowa zawierająca posiłki.

Opis atrybutów encji znajduje się w tabeli 5.21.

Tabela 5.21: Wykaz atrybutów encji typu Meal

*Źródło: Opracowanie własne*

Nazwa atrybutu	Opis atrybutu	Typ	OBL(+) OPC(-)
<i>mealID</i>	Numer identyfikujący posiłek	Liczba naturalna	+
<i>hour</i>	Godzina podawania posiłku	Max. znaków 10	-

Klucze kandydujące: mealID

Klucz główny: mealID

Charakter encji: encja słaba

Po zaprojektowaniu encji możemy zapisać predykatowe definicje typów encji:

ENC\01 ACTIVITY(activityID, date, description, time, intensivity, satisfaction, activityType)

ENC\02 COMPETITION (competitionID, spot, description, rank)

ENC\03 NOTIFICATION (notificationID, title, description, sendDate, createdDate)

ENC\04 PROFESTIONALS (profesionalsID, degree)

ENC\05 SPECIALISATION (specialisationID, name)

ENC\06 DIET (dietID, isActive)

ENC\07 PORTION (portionID, amount)

ENC\08 FORAGE (forageID, name, producent, capacity)

ENC\09 HORSE (horseID, name, mother, father, birthday, race, breeder, passport, photo)

ENC\10 HORSEGENDER (genderID, gender)

ENC\11 STATUS (statusID, name)

ENC\12 MEALNAME (mealNameID, mealName)

ENC\13 NUTRITIONPLAN (nutritionPlanID, title, description, icon)

ENC\14 PEOPLEDETIALS (detailsID, name, surname, phonNumber, email, city, street,  
number)

ENC\15 PARTICIPATION (participationID, level, result, place)

ENC\16 SHARED (sharedID, code, endDate, startDate)

ENC\17 UNITOFMEASURE (unitID, unitName)

ENC\18 USERACCOUNT (userID, accountLogin, hash, salt, createdDateTime)

ENC\19 USERTYPE (userTypeID, typeName)

ENC\20 VISIT (careID, cost, summary, artefactImage, visitDate)

ENC\21 MEAL (mealID, hour)

Predykatowe definicje związków encji:

**ZWI\xx Związek** (ENCJA1(min, max) ENCJA2(min, max))

**ZWI\01 Has** (HORSE(0,N) HORSEGENDER(1,1))

**ZWI\02 Has** (HORSE(0,N) HORSESTATUS(1,1))

**ZWI\03 Take** (HORSE(1,1) PARTICIPATION(0,N))

**ZWI\04 Refers to** (PARTICIPATION(0,N) COMPETITION(1,1))

**ZWI\05 Performs** (HORSE(1,1) ACTIVITY(0,N))

**ZWI\06 Train** (ACTIVITY(1,1) USERACOUNT(0,N))

**ZWI\07 Ride** (ACTIVITY(1,1) USERACOUNT(0,N))

**ZWI\08 Refers to** (CUSTOMNOTIFICATION(0,N) USERACOUNT(1,1))

**ZWI\09 Gets** (USERACOUNT(1,1) SHARE(0,N))

**ZWI\10 Shares** (USERACCOUNT(1,1) SHARE(0,N))

**ZWI\11 IsShared** (HORSE(1,1) SHARE(0,N))

**ZWI\12 Has** (USERACCOUNT(0,N), USERTYPE(1,1))

**ZWI\13 Concern** (PEOPLEDDETAILS(1,1) USERACCOUNT(0,N))

**ZWI\14 Attend** (HORSE(1,1) VISIT(0,N))

**ZWI\15 CareOn** (VISIT(0,N) PROFESSIONALS(1,1))

**ZWI\16 Has** (PROFESSIONALS(0,N) SPECIALISATION(1,1))

**ZWI\17 IsOn** (HORSE(1,1) DIET(0,N))

**ZWI\18 Contains** (DIET(0,N) NUTRITIONPLAN(1,1))

**ZWI\19 BelongsTo** (MEAL(0,N) NUTRITIONPLAN(0,1))

**ZWI\20 Has** (MEALNAME(1,1) MEAL(0,N))

**ZWI\21 ConsistOf** (MEAL(1,1) PORTION(0,N))

**ZWI\22 Contains** (PORTION(0,N) FORAGE(1,1))

**ZWI\23 Has** (FORAGE(0,N) UNITOFMEASURE(1,1))

**ZWI\24 Has** (PORTION(0,N) UNITOFMEASURE(1,1))

Model konceptualny przedstawiony został na diagramie ERD, który został przedstawiony na rysunku 5.1.





## 5.2 Model logiczny

Po stworzeniu modelu konceptualnego, należy przetransformować go do modelu logicznego. Poniżej przedstawiono tabele opisujące schematy relacji oraz znaczenia atrybutów tych relacji.

### REL\01 Activities\ACTIVITY

Opis schematu relacji znajduje się w tabeli 5.22.

Tabela 5.22: Opis schematu relacji Activities

*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>activityID</i>	Integer+		+			+	PR		BD
<i>date</i>	Date	dd-mm-rrrr	+						USER
<i>description</i>	String		-						USER
<i>time</i>	Integer		+						USER
<i>intensivity</i>	Integer+		+						USER
<i>satisfaction</i>	Integer+		+						USER
<i>activityType</i>	Integer+		+						USER
<i>userID</i>	Integer+		+				FK	User	BD
<i>horseID</i>	Integer+		+				FK	Horse	BD
<i>trainerID</i>	Integer+		-				FK	User	BD

Tabela 5.23: Opis atrybutów relacji Activities

Źródło: Opracowanie własne

Nazwa atrybutu	Znaczenie
<i>activityID</i>	Unikalne ID aktywności, generowane przez aplikację, klucz główny tabeli.
<i>date</i>	Data wykonywanej aktywności.
<i>description</i>	Opis aktywności, dowolne polskie znaki.
<i>time</i>	Liczba naturalna, oznaczająca czas trwania wykonywanej aktywności.
<i>intensivity</i>	Liczba naturalna od 0 do 5 oznaczająca poziom intensywności treningu.
<i>satisfaction</i>	Liczba naturalna od 0 do 5 oznaczająca poziom satysfakcji z treningu.
<i>activityType</i>	Liczba naturalna oznaczająca typ aktywności.
<i>userID</i>	Indentyfikator użytkownika, który wpisuje aktywność.
<i>horseID</i>	Identyfikator konia, którego dotyczy aktywność.
<i>trainerID</i>	Identyfikator użytkownika typu trener, który przeprowadzał trening.

**REL\02 Competitions\COMPETITION**

Opis schematu relacji znajduje się w tabeli 5.24.

Tabela 5.24: Opis schematu relacji Competitions

Źródło: Opracowanie własne

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>competitionID</i>	Integer+		+			+	PR		BD
<i>spot</i>	String		-						USER
<i>rank</i>	String		-						USER
<i>date</i>	Date		+						USER
<i>description</i>	String		+						USER

Tabela 5.25: Opis atrybutów relacji Competitions

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>competitionID</i>	Unikalny numer ID identyfikujący zawody, generowany przez aplikację.
<i>spot</i>	Miejsce, w którym odbędą się zawody.
<i>rank</i>	Ranga zawodów np. regionalne/międzynarodowe itp.
<i>date</i>	Dzień, w którym odbędą się zawody.
<i>description</i>	Opis zawodów.

**REL\03 Notifications\NOTIFICATION**

Tabela 5.26: Opis schematu relacji Notifications

*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>notificationID</i>	Integer+		+			+	PR		BD
<i>title</i>	String		+						USER
<i>description</i>	String		+						USER
<i>sendDate</i>	Date		+						USER
<i>createdDate</i>	Date		+						USER
<i>userID</i>	Integer+		+				FK	User	BD

Tabela 5.27: Opis atrybutów relacji Notifiactions

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>notificationID</i>	Unikalny numer ID identyfikujący powiadomienie
<i>title</i>	Tytuł powiadomienia
<i>description</i>	Opis pojawiający się na powiadomieniu
<i>sendDate</i>	Data i godzina informująca kiedy ma zostać wysłane powiadomienie
<i>createdDate</i>	Data i godzina stworzenia powiadomienia
<i>userID</i>	Numer ID identyfikujący użytkownika wysyłającego powiadomienie

**REL\04 Professionals\PROFESSIONAL**

Tabela 5.28: Opis schematu relacji Professionals

*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>professionalID</i>	Integer+		+			+	PR		BD
<i>degree</i>	String		-						USER
<i>detailsID</i>	Integer+		+				FK	Details	BD
<i>specialisationID</i>	Integer+		+				FK	Sepcialisation	BD

Tabela 5.29: Opis atrybutów relacji Professionals

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>professionalID</i>	Unikalny numer ID identyfikujący profesjonalistę
<i>degree</i>	Stopień naukowy profesjonalisty
<i>detailsID</i>	Numer ID identyfikujący dane personalne profesjonalistę
<i>specialisationID</i>	Numer ID identyfikujący specjalizację profesjonalisty

## REL\05 Specialisations\SPECIALISATION

Tabela 5.30: Opis schematu relacji Specialisations

*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>specialisationID</i>	Integer+		+			+	PK		BD
<i>name</i>	String		+						USER

Tabela 5.31: Opis atrybutów relacji Professionals

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>specialisationID</i>	Unikalny numer ID identyfikujący specjalizację.
<i>name</i>	Nazwa specjalizacji

## REL\06 Diets\DIET

Tabela 5.32: Opis schematu relacji Diets

*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>dietID</i>	Integer+		+			+	PR		BD
<i>isActive</i>	Boolean		+	true					
<i>horseID</i>	Integer+		+				FK	Horse	BD
<i>nutritionPlanID</i>	Integer+		+				FK	NutritionPlan	BD

Tabela 5.33: Opis atrybutów relacji Diets

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>dietID</i>	Unikalny numer ID identyfikujący daną dietę
<i>isActive</i>	Zmienna przyjmująca wartości true/false
<i>horseID</i>	Numer ID identyfikujący konia
<i>nutritionPlanID</i>	Numer ID identyfikujący plan żywienia

**REL\07 Portions\PORTION**

Tabela 5.34: Opis schematu relacji Portions

*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>portionID</i>	Integer+		+			+	PR		BD
<i>amount</i>	Float+		+	1	0<				USER

Tabela 5.35: Opis atrybutów relacji Portions

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>portionID</i>	Unikalny numer ID identyfikujący porcję jedzenia dla konia
<i>amount</i>	Ilość jedzenia w porcji

**REL\08 Forges\FORAGE** (*forageID*, *name*, *producent*, *capacity*)

Tabela 5.36: Opis schematu relacji Forges

*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>forageID</i>	Integer+		+			+	PR		BD
<i>name</i>	String		+						USER
<i>producent</i>	String		+						USER
<i>capacity</i>	String		+						USER
<i>unitID</i>	Integer+		+				FK	UnitOfMeasure	BD

Tabela 5.37: Opis atrybutów relacji Forges

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>forageID</i>	Unikalny numer ID identyfikujący paszę
<i>name</i>	Nazwa paszy
<i>producent</i>	Nazwa producenta paszy
<i>capacity</i>	Liczba naturalna oznaczająca ilość paszy w jednej paczce paszy
<i>unitID</i>	Numer ID identyfikujący jednostkę miary

**REL\09 Horses\HORSE**



Tabela 5.38: Opis schematu relacji Horses

*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>horseID</i>	Integer+		+			+	PR		BD
<i>name</i>	Integer+		+						USER
<i>mother</i>	Integer+		+						USER
<i>father</i>	Integer+		+						USER
<i>birthday</i>	Date		+						USER
<i>race</i>	String		+						USER
<i>breeder</i>	String		+						USER
<i>passport</i>	String		+						USER
<i>photo</i>	String		+						USER
<i>statusID</i>	Integer+		+				FK	Status	USER
<i>genderID</i>	Integer+		+				FK	HorseGender	USER

Tabela 5.39: Opis atrybutów relacji Horses

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>horseID</i>	Unikalny numer ID identyfikujący konia
<i>name</i>	Imię konia
<i>mother</i>	Imię matki konia
<i>father</i>	Imię ojca konia
<i>birthday</i>	Data urodzenia
<i>race</i>	Rasa konia
<i>breeder</i>	Nazwa hodowli lub imię i nazwisko hodowcy
<i>passport</i>	Numer paszportu
<i>photo</i>	URL zdjęcia
<i>statusID</i>	Numer ID identyfikujący status konia
<i>genderID</i>	Numer ID identyfikujący płeć konia

**REL\10 HorseGenders\HORSEGENDER**

Tabela 5.40: Opis schematu relacji HorseGenders

*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>genderID</i>	Integer+		+			+	PK		USER
<i>gender</i>	String								USER

Tabela 5.41: Opis atrybutów relacji HorseGenders

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>genderID</i>	Unikalny numer ID identyfikujący płeć konia
<i>gender</i>	Nazwa płci

## REL\11 Status\STATUS

Tabela 5.42: Opis schematu relacji Specialisations

*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>statusID</i>	Integer+		+			+	PK		BD
<i>name</i>	String		+						USER

Tabela 5.43: Opis atrybutów relacji Status

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>statusID</i>	Unikalny numer ID identyfikujący status konia bądź użytkownika
<i>name</i>	Nazwa statusu

## REL\12 MealNames\MEALNAME

Tabela 5.44: Opis schematu relacji MealNames

*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>mealNameID</i>	Integer+		+			+	PK		BD
<i>mealName</i>	String		+						USER

Tabela 5.45: Opis atrybutów relacji MealNames

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>mealNameID</i>	Unikatowy numer ID identyfikujący nazwę posiłku
<i>mealName</i>	Nazwa posiłku

**REL\13 NutritionPlans\NUTRITIONPLAN**

Tabela 5.46: Opis schematu relacji NutritionPlans

*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>nutritionPlanID</i>	Integer+		+			+	PK		BD
<i>title</i>	String		+						USER
<i>description</i>	String		-						USER
<i>icon</i>	Integer+		+	1					USER

Tabela 5.47: Opis atrybutów relacji NutritionPlans

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>nutritionPlanID</i>	Unikatowy numer ID identyfikujący plan żywienia
<i>title</i>	Tytuł planu żywienia
<i>description</i>	Opis planu żywienia
<i>icon</i>	Id ikonki wyświetlanej koło planu żywienia

**REL\14 PeopleDetails\PEOPLEDETAILS**

Tabela 5.48: Opis schematu relacji PeopleDetails

*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>detailsID</i>	Integer+		+			+	PK		BD
<i>name</i>	String		+						USER
<i>surname</i>	String		-						USER
<i>phonNumber</i>	String	+_____	+						USER
<i>email</i>	Integer+	_____@____	+						USER
<i>city</i>	Integer+		+						USER
<i>street</i>	Integer+		+						USER
<i>number</i>	Integer+		+						USER

Tabela 5.49: Opis atrybutów relacji PeopleDetails

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>detailsID</i>	Unikalny numer ID identyfikujący detale ludzi
<i>name</i>	Imie
<i>surname</i>	Nazwisko
<i>phonNumber</i>	Numer telefonu
<i>email</i>	Adres e-mail
<i>city</i>	Miasto
<i>street</i>	Ulica
<i>number</i>	Numer domu

REL\15 Participations\PARTICIPATION

Tabela 5.50: Opis schematu relacji Participations

*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>ParticipationID</i>	Integer+		+			+	PK		BD
<i>level</i>	Integer+		+						BD
<i>result</i>	String		+						USER
<i>place</i>	String		+						USER
<i>competitionID</i>	Integer+		+						BD
<i>horseID</i>	Integer+		+						BD

Tabela 5.51: Opis atrybutów relacji Participations

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>ParticipationID</i>	Unikalny numer ID identyfikujący start w zawodach.
<i>level</i>	Poziom konkursu, w którym koń brał udział
<i>result</i>	Wynik z danego konkursu
<i>place</i>	Miejsce uzyskane w danym konkursie
<i>competitionID</i>	Numer ID zawodów, w których koń bierze udział
<i>horseID</i>	Numer ID konia biorącego udział w zawodach

REL\16 Shareds\SHARED

Tabela 5.52: Opis schematu relacji Shareds

*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>sharedID</i>	Integer+		+			+	PK		BD
<i>code</i>	String		+						USER
<i>endDate</i>	Date		+						USER
<i>startDate</i>	Date		+						USER
<i>horseID</i>	Integer+		+						USER
<i>userSharedID</i>	Integer+		+				FK	USER	BD
<i>userScanID</i>	Integer+		+				FK	USER	BD

Tabela 5.53: Opis atrybutów relacji Shareds

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>sharedID</i>	Unikalny numer ID identyfikujący pojedyncze udostępnienie konia między dwoma użytkownikami
<i>code</i>	Kod Qr dzięki któremu użytkownicy mogą udostępniać między sobą konie.
<i>endDate</i>	Data kończąca udostępnianie
<i>startDate</i>	Data od której koń będzie udostępniony
<i>horseID</i>	Numer ID identyfikujący udostępnianego konia
<i>userSharedID</i>	Numer ID identyfikujący użytkownika, który udostępnia konia
<i>userScanID</i>	Numer ID identyfikujący użytkownika, któremu zostanie udostępniony koń

REL\17 UnitOfmeasures\UNITOFMEASURE

Tabela 5.54: Opis schematu relacji UnitOfmeasures

*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>unitID</i>	Integer+		+			+			BD
<i>unitName</i>	String		+						USER

Tabela 5.55: Opis atrybutów relacji UnitOfmeasures

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>unitID</i>	Unikalny numer ID identyfikujący jednostkę miary
<i>unitName</i>	Nazwa jednostki miary

**REL\18 UserAccounts\USERACCOUNT**

Tabela 5.56: Opis schematu relacji UserAccounts

*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>userID</i>	Integer+		+			+	PK		BD
<i>accountLogin</i>	String		+			+			USER
<i>hash</i>	String		+						USER
<i>salt</i>	String		+						USER
<i>createdDateTime</i>	Datetime		+						USER
<i>typeID</i>	Integer+		+				FK	UserTypes	BD
<i>detailsID</i>	Integer+		+				FK	PeopleDetails	BD



Tabela 5.57: Opis atrybutów relacji UserAccounts

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>userID</i>	Unikalny numer ID identyfikujący użytkownika
<i>accountLogin</i>	Login użytkownika
<i>hash</i>	Hash powstający z hasła użytkownika
<i>salt</i>	Do hasła użytkownika
<i>createdDateTime</i>	Data utworzenia konta
<i>typeID</i>	Numer identyfikujący typ użytkownika
<i>detailsID</i>	Numer identyfikujący detale osobowe użytkownika

**REL\19 UserTypes\USERTYPE**

Tabela 5.58: Opis schematu relacji UserTypes

*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>userTypeID</i>	Integer+		+			+	PK		BD
<i>typeName</i>	String		+						USER

Tabela 5.59: Opis atrybutów relacji UserTypes

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>userTypeID</i>	Unikalny numer ID identyfikujący typ użytkownika
<i>typeName</i>	Nazwa typu użytkownika

**REL\20 Visits\VISIT**

Tabela 5.60: Opis schematu relacji Visits

*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>careID</i>	Integer+		+			+	PK		BD
<i>cost</i>	Float		+						USER
<i>summary</i>	String		-						USER
<i>image</i>	String		-						USER
<i>visitDate</i>	Date		+						USER
<i>horseID</i>	Integer+		+				FK	Horse	DB
<i>professionalID</i>	Integer+		+				FK	Professional	DB

Tabela 5.61: Opis atrybutów relacji Visits

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>visitID</i>	Unikalny numer ID identyfikujący wizytę
<i>visitDate</i>	Data wizyty
<i>summary</i>	Podsumowanie wizyty, opis przepisanych leków i innych zaleceń
<i>image</i>	Obraz z wizyty
<i>cost</i>	Cena wizyty
<i>horseID</i>	Numer ID identyfikujący konia
<i>professionalID</i>	Numer ID identyfikujący profesjonalistę, który przeprowadza wizytę

REL\21 Meals\MEAL

Tabela 5.62: Opis schematu relacji Meals

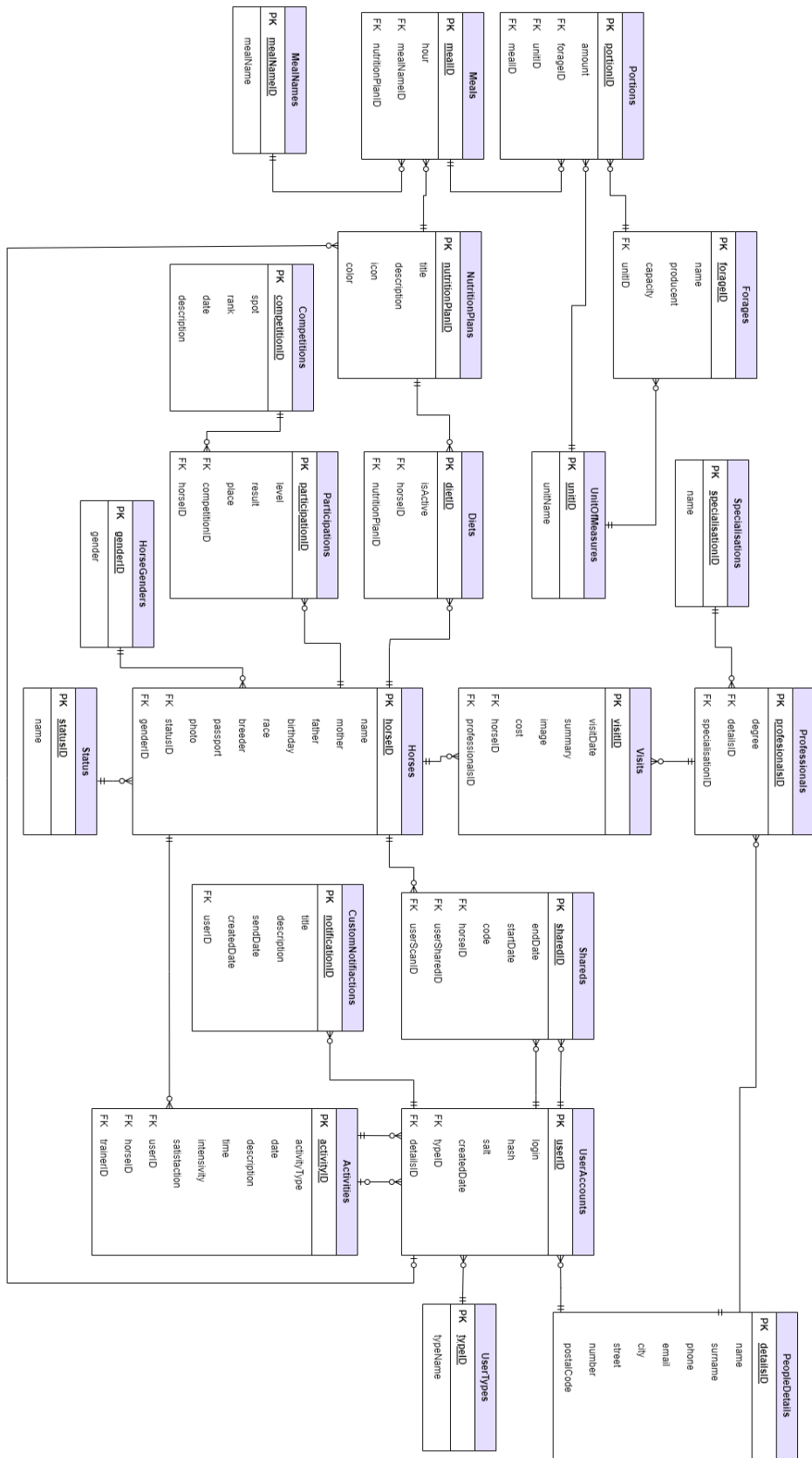
*Źródło: Opracowanie własne*

Nazwa atrybutu	Dziedzina	Maska	OBL(+) OPC(-)	Wartość domyślna	Ograniczenia	Unikalność	Klucz	Referencje	Źródło danych
<i>mealID</i>	Integer+		+			+			BD
<i>hour</i>	String		+						USER
<i>mealNameID</i>	Integer+		+				FK	MealName	BD
<i>nutritionPlanID</i>	Integer+		+				FK	NutritionPlan	BD

Tabela 5.63: Opis atrybutów relacji Meals

*Źródło: Opracowanie własne*

Nazwa atrybutu	Znaczenie
<i>mealID</i>	Unikalny numer ID identyfikujący posiłek
<i>hour</i>	Godzina, w której jedzony jest posiłek
<i>mealNameID</i>	Numer identyfikujący nazwę posiłku
<i>nutritionPlanID</i>	Numer identyfikujący plan żywienia

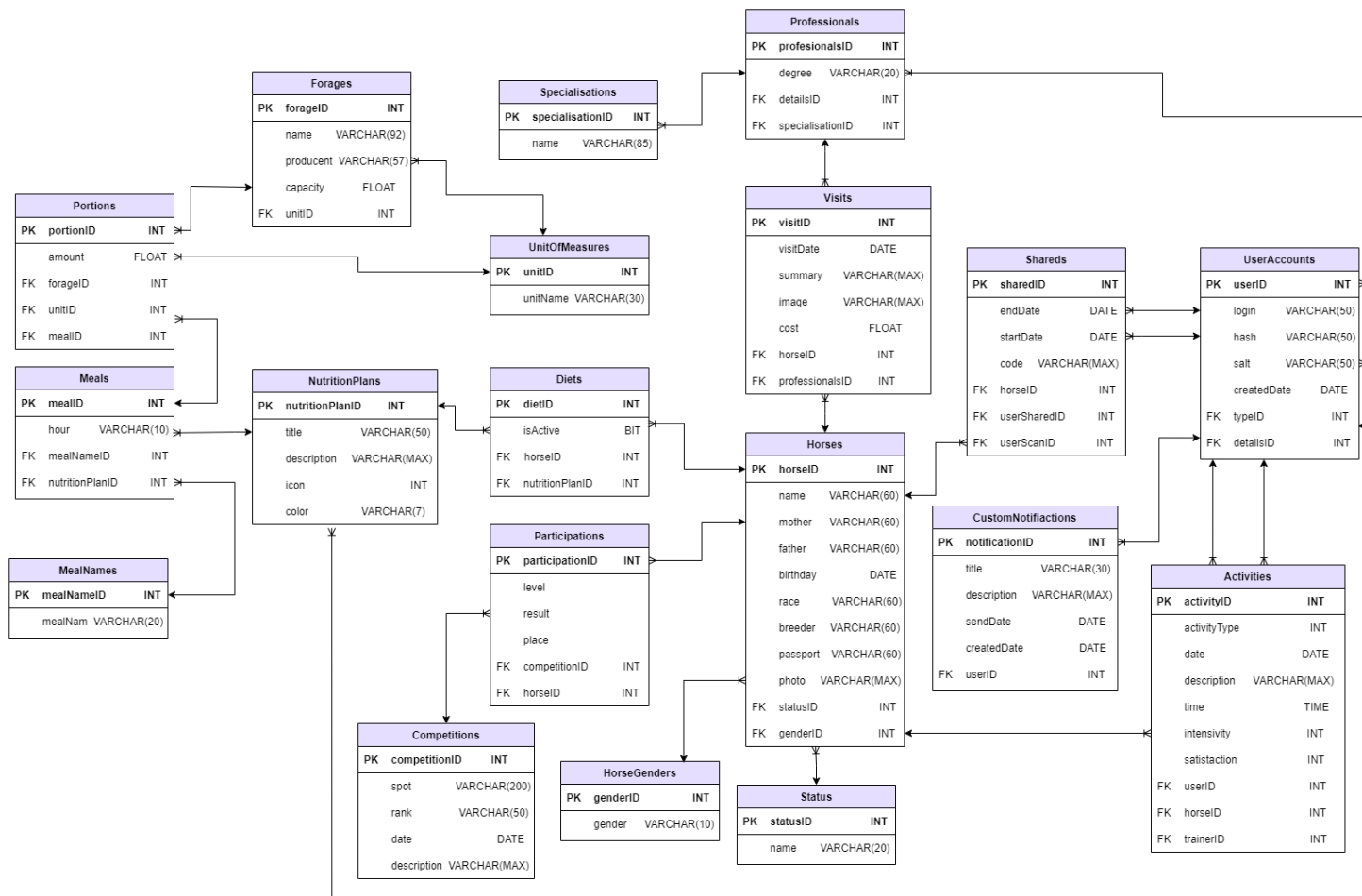


Rysunek 5.2: Logiczny schemat bazy danych

Źródło: Opracowanie własne

## 5.3 Model fizyczny

Po stworzeniu logicznego modelu bazy danych, możemy przetransformować go w model fizyczny. Model fizyczny składa się z plików i rekordów. Pilki bazy danych złożony jest z poszczególnych rekordów, przy czym rekordy te mają // dokończyć na podstawie książki Sharon Allen



Rysunek 5.3: Fizyczny schemat bazy danych

*Źródło: Opracowanie własne*

# 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

Aplikacja mobilna "HorseTracking" służy do zapisywania dziennych aktywności koni, ich wizyt u lekarzy, kowali jak także do zarządzania zawodami.

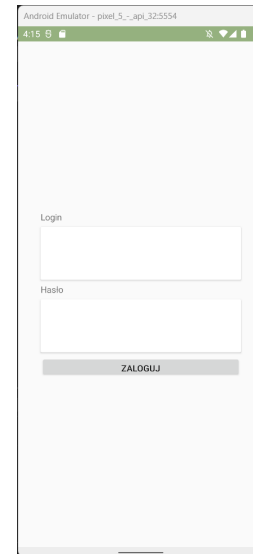
Po pierwszym otwarciu aplikacji użytkownikowi ukaże się okno logowania przedstawione na rysunku 8.1.

Na tym ekranie użytkownik może zalogować się do aplikacji. **Możliwa jest także opcja zresetowania swojego hasła, w przypadku zapomnienia hasła.** Rejestracja do aplikacji nie jest możliwa, ponieważ ta funkcja jest dostępna jedynie dla administratora w aplikacji desktopowej. Po wprowadzeniu loginu i hasła, a następnie kliknięciu przycisku "Zaloguj" sprawdzana jest poprawność wprowadzonych danych.

Jeśli przy próbie zalogowania podane zostaną nieprawidłowe dane logowania, bądź użytkownika nie ma w systemie, zostanie on o tym poinformowany (patrz. rys 8.2) Użytkownik nieposiadający koni (użytkownik typu appOwner) nie może zalogować się do aplikacji mobilnej, ponieważ służy ona tylko do wpisywania danych o swoich koniach.

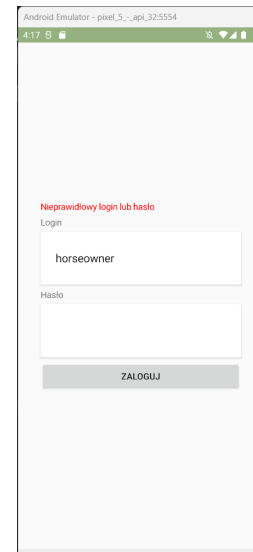
Po poprawnym zalogowaniu się dane użytkownika zostają zapamiętane, więc przy kolejnym otwarciu aplikacji użytkownik będzie już zalogowany. Aby wylogować się z aplikacji użytkownik musi otworzyć menu boczne i wybrać opcję "Wyloguj".

Po zalogowaniu do aplikacji użytkownik zostaje przeniesiony na okno główne. Zawie-



Rysunek 8.1: Logowanie do aplikacji.

*Źródło: Opracowanie własne*



Rysunek 8.2: Błędne dane logowania.

*Źródło: Opracowanie własne*

ra ono menu dolne pozwalające na nawigację pomiędzy czterema głównymi sekcjami aplikacji: aktywności, wizyty, żywienie, zawody. Na początek wyświetlona zostaje strona dotycząca aktywności. W tym widoku można przeglądać informacje o wszystkich aktywnościach koni posiadanych lub tych które trenujemy.

Aktywności dotyczą konia, którego imię podane jest w polu powyżej. Aby zmienić konia wystarczy kliknąć w to pole i wybrać innego konia. Strzałki lewo-prawo widoczne na dole ekranu umożliwiają nawigację między kolejnymi tygodniami. W przypadku braku aktywności w danym dniu, wyświetlony jest napis informujący o braku aktywności tego dnia. Każdy typ aktywności ma inny kolor i ikonę, aby ułatwić identyfikację. Po kliknięciu w daną aktywność możemy przejść do detali dotyczących tej aktywności. Ekran szczegółów został przedstawiony na rysunku 8.9 i zostanie omówiony później.


Pomiędzy strzałkami nawigującymi między tygodniami znajduje się okrągły przycisk z ikoną "+". Umożliwia on dodawanie aktywności dla aktualnie wybranego konia. Przycisk ten dostępny jest jedynie dla właściciela konia oraz osób którym dany koń został udostępniony. Oznacza to, że jeśli użytkownik jest trenerem ta opcja jest dla niego zablokowana, a przycisk nie jest widoczny.



Rysunek 8.3: Ekran aktywności.

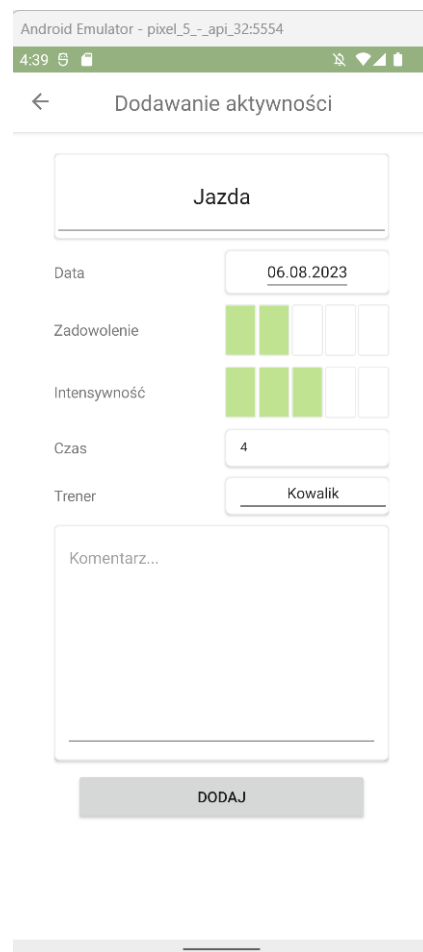
*Źródło: Opracowanie własne*

Po kliknięciu w przycisk "+" użytkownik zostaje przeniesiony na okno "Dodawanie aktywności".



Rysunek 8.4: Okno dodawania aktywności.

*Źródło: Opracowanie własne*



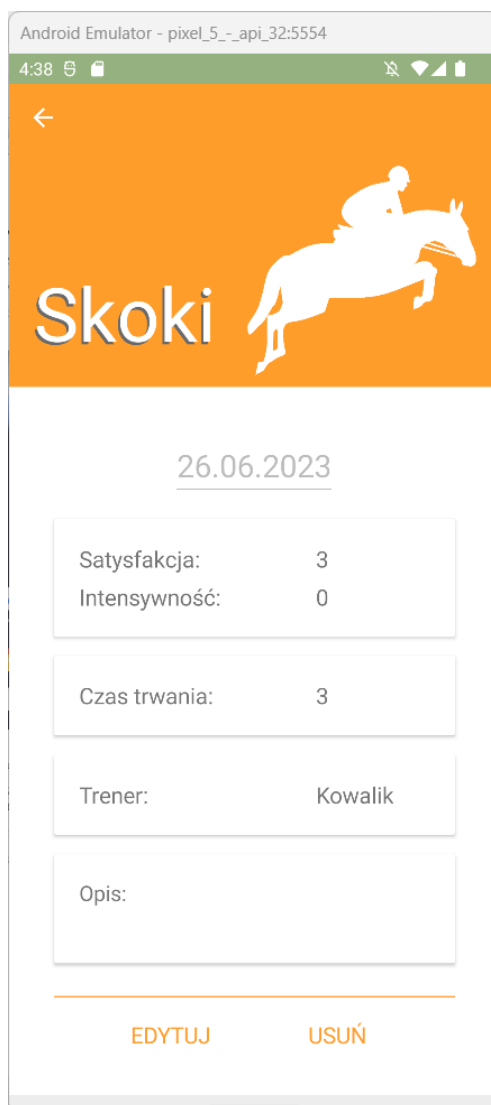
Rysunek 8.5: Okno dodawania aktywności rozszerzone.

*Źródło: Opracowanie własne*

Okno to wygląda różnie w zależności od tego jaki typ aktywności chcemy dodać (patrz rys. 8.4 i rys. 8.5). Na początku wyświetlony jest prostszy model okna, a po wybraniu typu aktywności dostosowuje się. Dla aktywności: jazda, skoki, zawody, kros czy skoki w oknie dochodzą nowe opcje takie jak satysfakcja, intensywność oraz wybór trenera (patrz. rys 8.5). Po uzupełnieniu wszystkich niezbędnych informacji aktywność zostaje dodana i pojawia się na ekranie głównym. Jeśli któraś z niezbędnych informacji nie zostanie uzupełniona aktywność nie doda

się, użytkownik zostanie poinformowany o nieprawidłowościach i będzie mógł je poprawić. Po dodaniu aktywności użytkownik zostaje przeniesiony z powrotem na okno główne, gdzie po kliknięciu w wybraną aktywność może zobaczyć jej szczegóły (patrz rys.8.9 ).

W oknie szczegółów oprócz przeczytania wszystkich informacji dotyczących aktywności można także przejść do edycji lub usunąć daną aktywność. Przy edycji otwiera się to samo okno co przy dodawaniu aktywności jednakże tym razem jest ono wypełnione aktualnymi danymi wybranej aktywności. Po zakończonej edycji użytkownik zostaje przeniesiony na okno główne. Po kliknięciu przycisku usuń, wyświetla się komunikat proszący o potwierdzenie wykonania akcji. Jeśli użytkownik potwierdzi, że akcje, to aktywność zostanie usunięta, a użytkownik zostanie przeniesiony na ekran główny. **Usunięcie aktywności jest także możliwe poprzez długie przytrzymanie wybranej aktywności na ekranie głównym, a następnie potwierdzenie akcji na pojawiającym się komunikacie.**



Rysunek 8.6: Szczegóły aktywności.

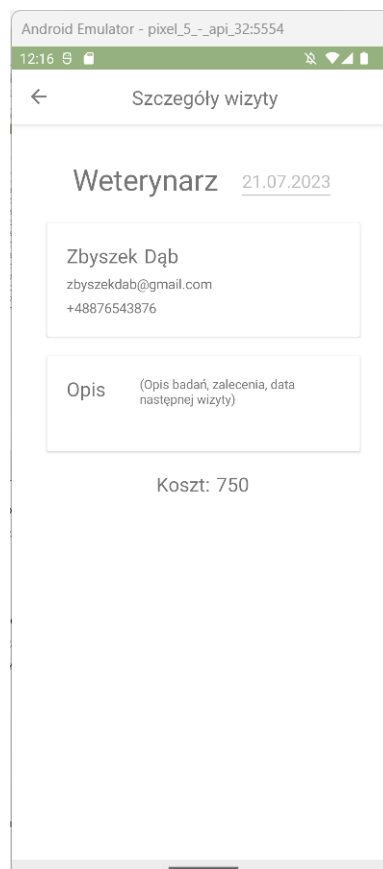
*Źródło: Opracowanie własne*

Kolejną opcją w menu dolnym są wizyty. Na tym oknie podobnie jak w oknie aktywności mamy pole pozwalające wybrać konia o którym informacje chcemy obejrzeć. Koń wybrany na oknie aktywności przenosi się na okno wizyt i na odwrót. W tym oknie można sprawdzić jakie wizyty odbył ostatnio wybrany koń.



Rysunek 8.7: Okno wizyt.

*Źródło: Opracowanie własne*

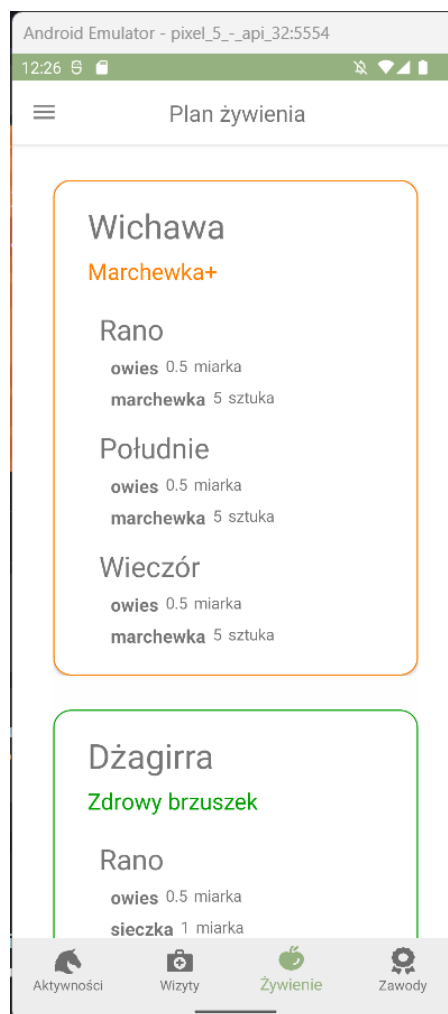


Rysunek 8.8: Szczegóły wizyty.

*Źródło: Opracowanie własne*

Na rysunku 8.7 widzimy listę wizyt wybranego konia. Po kliknięciu w wizytę zostaniemy przeniesieni do szczegółów wizyty, gdzie możemy znaleźć dane kontaktowe do lekarza/kowala, który przeprowadził wizytę oraz szczegóły takie jak opis wizyty i jej koszt. Dzięki temu widokowi użytkownik może sprawdzić jakie zalecenia były na poprzednich wizytach, jaki był ich koszt i kiedy dokładnie się odbyły.

Kolejną opcją dostępną w menu dolnym są plany żywienia. Po kliknięciu w ikonę "jabłka" użytkownik zostanie przeniesiony na stronę z planami żywienia wszystkich jego koni. W widoku tym można jedynie obejrzeć plan żywienia, nie jest możliwe ich dodanie, edycja bądź usunięcie. Całość zarządzania planami żywienia została zaimplementowana w aplikacji desktopowej.



Rysunek 8.9: Szczegóły aktywności.

*Źródło: Opracowanie własne*

## Rozdział 9

## Podsumowanie

# Bibliografia

- [1] Hanna Mazur, Zygmunt Mazur, *Projektowanie relacyjnych baz danych*. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2004.
- [2] profexorgeek, alexbuckgit, v-hearya, davidbritch, conceptdev *Co to jest środowisko Xamarin?* <https://learn.microsoft.com/pl-pl/xamarin/get-started/what-is-xamarin> [Dostęp: 06.08.2023]
- [3] JonDouglas, alexbuckgit, Mikejo5000, v-hearya, zivkan, chrisraygill, loic-sharma, karann-msft, NickKruger, mairaw, kraigb, alfredmyers, *Wprowadzenie do narzędzia NuGet* <https://learn.microsoft.com/pl-pl/nuget/what-is-nuget> [Dostęp: 06.08.2023]
- [4] Paweł Łukasiewicz *C# - Entity Framework* <https://www.plukasiewicz.net/Artykuly/EntityFramework> [Dostęp: 06.08.2023]
- [5] Juris Lavrinovics, *Figma - narzędzie do projektowania interfejsu użytkownika* <https://blog.consdata.tech/2023/02/15/uiux-tools.html> [Dostęp 05.05.2023]
- [6] <https://www.figma.com/about/> [Dostęp 05.05.2023]
- [7] Figma *Figma-logo* <https://commons.wikimedia.org/wiki/File:Figma-logo.svg> [Dostęp 05.05.2023]
- [8] adegeo, ihsansfd, alexbuckgit, v-trisshores, DCtheGeek, *Przewodnik dotyczący aplikacji klasycznych (WPF .NET)* <https://learn.microsoft.com/pl-pl/dotnet/desktop/wpf/overview/?view=netdesktop-7.0> [Dostęp: 06.08.2023]



# Spis rysunków

3.1	Architektura platformy Xamarin . . . . .	5
3.2	Architektura platformy Xamarin.Android . . . . .	6
3.3	Logo systemu NuGet . . . . .	6
3.4	Przepływ informacji . . . . .	7
3.5	przepływ informacji . . . . .	8
3.6	przepływ informacji . . . . .	8
4.1	Diagram Use Case dla aplikacji desktopowej . . . . .	16
4.2	Diagram Use Case dla aplikacji mobilnej . . . . .	17
5.1	Diagram ERD . . . . .	45
5.2	Logiczny schemat bazy danych . . . . .	64
5.3	Fizyczny schemat bazy danych . . . . .	65
8.1	Logowanie do aplikacji. . . . .	69
8.2	Błędne dane logowania. . . . .	69
8.3	Ekran aktywności. . . . .	70
8.4	Okno dodawania aktywności. . . . .	71
8.5	Okno dodawania aktywności rozszerzone. . . . .	71
8.6	Szczegóły aktywności. . . . .	72
8.7	Okno wizyt. . . . .	73
8.8	Szczegóły wizyty. . . . .	73
8.9	Szczegóły aktywności. . . . .	74

# Listings

# Spis tabel

4.1	Wymagania funkcjonalne obu aplikacji . . . . .	11
4.2	Wymagania funkcjonalne aplikacji mobilnej . . . . .	12
4.3	Wymagania funkcjonalne aplikacji mobilnej . . . . .	13
4.4	Wymagania funkcjonalne aplikacji desktopowej . . . . .	14
4.5	Wymagania funkcjonalne aplikacji desktopowej . . . . .	15
4.6	Wymagania нефункционалне aplikacji desktopowej . . . . .	19
4.7	Wymagania нефункционалне aplikacji mobilnej . . . . .	21
5.1	Wykaz аtrybutów encji typu Activity . . . . .	29
5.2	Wykaz аtrybutów encji typu Competition . . . . .	30
5.3	Wykaz аtrybutów encji typu Notification . . . . .	30
5.4	Wykaz аtrybutów encji typu Doctor . . . . .	31
5.5	Wykaz аtrybutów encji typu Specialization . . . . .	31
5.6	Wykaz аtrybutów encji typu Diet . . . . .	32
5.7	Wykaz аtrybutów encji typu Portion . . . . .	32
5.8	Wykaz аtrybutów encji typu Forage . . . . .	33
5.9	Wykaz аtrybutów encji typu Horse . . . . .	34
5.10	Wykaz аtrybutów encji typu HorseGender . . . . .	34
5.11	Wykaz аtrybutów encji typu Status . . . . .	35
5.12	Wykaz аtrybutów encji typu MealName . . . . .	35
5.13	Wykaz аtrybutów encji typu NutrtrionPlan . . . . .	36
5.14	Wykaz аtrybutów encji typu PeopleDetails . . . . .	37
5.15	Wykaz аtrybutów encji typu Participation . . . . .	37
5.16	Wykaz аtrybutów encji typu Shared . . . . .	38
5.17	Wykaz аtrybutów encji typu UnitOfMeasure . . . . .	39
5.18	Wykaz аtrybutów encji typu UserAcount . . . . .	39

5.19 Wykaz atrybutów encji typu UserType . . . . .	40
5.20 Wykaz atrybutów encji typu Visit . . . . .	40
5.21 Wykaz atrybutów encji typu Meal . . . . .	41
5.22 Opis schematu relacji Activities . . . . .	46
5.23 Opis atrybutów relacji Activities . . . . .	47
5.24 Opis schematu relacji Competitions . . . . .	47
5.25 Opis atrybutów relacji Competitions . . . . .	48
5.26 Opis schematu relacji Notifications . . . . .	48
5.27 Opis atrybutów relacji Notifiactions . . . . .	49
5.28 Opis schematu relacji Professionals . . . . .	49
5.29 Opis atrybutów relacji Professionals . . . . .	49
5.30 Opis schematu relacji Specialisations . . . . .	50
5.31 Opis atrybutów relacji Professionals . . . . .	50
5.32 Opis schematu relacji Diets . . . . .	50
5.33 Opis atrybutów relacji Diets . . . . .	51
5.34 Opis schematu relacji Portions . . . . .	51
5.35 Opis atrybutów relacji Portions . . . . .	51
5.36 Opis schematu relacji Forges . . . . .	52
5.37 Opis atrybutów relacji Forges . . . . .	52
5.38 Opis schematu relacji Horses . . . . .	53
5.39 Opis atrybutów relacji Horses . . . . .	54
5.40 Opis schematu relacji HorseGenders . . . . .	54
5.41 Opis atrybutów relacji HorseGenders . . . . .	54
5.42 Opis schematu relacji Specialisations . . . . .	55
5.43 Opis atrybutów relacji Status . . . . .	55
5.44 Opis schematu relacji MealNames . . . . .	55
5.45 Opis atrybutów relacji MealNames . . . . .	56
5.46 Opis schematu relacji NutritionPlans . . . . .	56
5.47 Opis atrybutów relacji NutritionPlans . . . . .	56
5.48 Opis schematu relacji PeopleDetails . . . . .	57
5.49 Opis atrybutów relacji PeopleDetails . . . . .	57
5.50 Opis schematu relacji Participations . . . . .	58
5.51 Opis atrybutów relacji Participations . . . . .	58

5.52	Opis schematu relacji Shareds . . . . .	59
5.53	Opis atrybutów relacji Shareds . . . . .	59
5.54	Opis schematu relacji UnitOfmeasures . . . . .	60
5.55	Opis atrybutów relacji UnitOfmeasures . . . . .	60
5.56	Opis schematu relacji UserAccounts . . . . .	60
5.57	Opis atrybutów relacji UserAccounts . . . . .	61
5.58	Opis schematu relacji UserTypes . . . . .	61
5.59	Opis atrybutów relacji UserTypes . . . . .	61
5.60	Opis schematu relacji Visits . . . . .	62
5.61	Opis atrybutów relacji Visits . . . . .	62
5.62	Opis schematu relacji Meals . . . . .	63
5.63	Opis atrybutów relacji Meals . . . . .	63

## Rozdział 10

### Opis zawartości APD