

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

1	Wprowadzenie	2
1.1	Geneza powstania pracy	2
1.2	Przegląd istniejących rozwiązań	2
1.3	Cel pracy	2
2	Projekt aplikacji	3
2.1	Wymagania funkcjonalne i нефункционалне	3
2.2	Baza danych	5
2.2.1	Model konceptualny	5
2.2.2	Model logiczny	5
2.2.3	Model fizyczny	5
2.3	Przypadki użycia	5
2.3.1	Diagram przypadków użycia	5
2.3.2	Scenariusze przypadków użycia	5
2.4	Diagramy stanów	5
2.5	Diagramy aktywności	5
2.6	Diagram klas	5
2.7	Architektura aplikacji	5
3	Implementacja	6
3.1	Środowisko programistyczne	6
3.2	Technologie użyte w pracy	6
3.3	Wykorzystane wzorce projektowe	6
3.4	Model architektoniczny MVVM	6
3.5	Interfejs użytkownika	6
3.6	Wybrane aspekty implementacyjne	6
4	Testy aplikacji	7
4.1	Unit testy	7
4.2	Test case	7
4.3	Baza błędów	7
5	Podsumowanie	8
6	Bibliografia	9

7	Spis rysunków	10
8	Spis listingów	11
9	Spis tabel	12
10	Opis zawartości APD	13

Wstep

Rozdział 1

Wprowadzenie

1.1 Geneza powstania pracy

1.2 Przegląd istniejących rozwiązań

Na rynku dostępne jest wiele aplikacji dedykowanych właścicielom koni. Wiele aplikacji poświęconych jest monitorowaniu treningów. Jest to funkcjonalność, której nie ma w stworzonej aplikacji, więc aplikacje tego typu nie będą analizowane. Przygotowane aplikacje zawierają podobne funkcjonalności do aplikacji "HorseApp". Oprócz opisu działania aplikacji oraz analizy podobieństw, przeanalizowane zostaną mocne i słabe strony każdej z aplikacji.

FEI HorseApp Lician Horse Ridely The equestrian

1.3 Cel pracy

Rozdział 2

Projekt aplikacji

2.1 Wymagania funkcjonalne i нефunkcjonalne

Funkcjonalności aplikacji mobilnej oraz desktopowej nie są takie same mimo iż są podłączone do jednej bazy, więc czerpią z tego samego źródła informacji. Pomimo różnic niektóre funkcjonalności pokrywają się w obu tych produktach.

Wymagania funkcjonalne które muszą spełniać obie aplikacje:

- logowanie do aplikacji,
- resetowanie hasła przez e-mail,

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:

- zarządzanie aktywnościami (dodawanie, edytowanie, usuwanie)
- wyświetlanie dodanych aktywności oraz ich szczegółowych informacji,
- planowanie wizyt,
- zarządzanie wizytami (dodawanie, edytowanie, usuwanie)
- wyświetlanie szczegółów wizyt,
- zapisywanie zdjęć z wizyt lekarzy,
- przypomnienia o wizytach,
- sprawdzenie/wybór planu żywienia,
- sprawdzenie dat zawodów,

- potwierdzanie udziału w zawodach,
- udostępnianie swojego konia innym użytkownikom aplikacji

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.

Wymagania funkcjonalne dla aplikacji desktopowej:

- zarządzanie kontami użytkowników,
- tworzenie planów żywienia,
- zarządzanie końmi,
- tworzenie statystyk aktywności,
- przeglądanie historii wizyt,
- planowanie zawodów

Jakiś krótki tekst

Wymagania нефункциональные:

2.2 Baza danych

2.2.1 Model konceptualny

2.2.2 Model logiczny

2.2.3 Model fizyczny

2.3 Przypadki użycia

2.3.1 Diagram przypadków użycia

2.3.2 Scenariusze przypadków użycia

2.4 Diagramy stanów

2.5 Diagramy aktywności

2.6 Diagram klas

2.7 Architektura aplikacji

Jaka baza jakie połączenie itp.

Rozdział 3

Implementacja

3.1 Środowisko programistyczne

3.2 Technologie użyte w pracy

3.3 Wykorzystane wzorce projektowe

3.4 Model architektoniczny MVVM

3.5 Interfejs użytkownika

3.6 Wybrane aspekty implementacyjne

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

Rozdział 4

Testy aplikacji

4.1 Unit testy

4.2 Test case

4.3 Baza błędów

Rozdział 5

Podsumowanie

Rozdział 6

Bibliografia

Rozdział 7

Spis rysunków

Rozdział 8

Spis listingów

Rozdział 9

Spis tabel

Rozdział 10

Opis zawartości APD