

Koncepcja wykonania systemu - aplikacja Psinder

Dominik Walecko, Wojciech Woźniak

A. Scenariusze Przypadków Użycia dla Aplikacji Psinder

1. Rejestracja użytkownika lub schroniska

- 1.1. Użytkownik uruchamia aplikację Psinder.
 - 1.2. Użytkownik ma do wyboru opcje: "Zarejestruj się" i "Zaloguj się".
 - 1.3. Wybór opcji "Zarejestruj się" wyświetla formularz, gdzie należy wypełnić:
 - Login (*max. 30 znaków alfanumerycznych*)
 - Adres e-mail (*max. 70 znaków alfanumerycznych i specjalnych*)
 - Hasło (*max. 60 znaków alfanumerycznych i specjalnych*)
 - Typ konta (*osobiste lub schronisko*).
 - 1.4. Po rejestracji użytkownik otrzymuje e-mail weryfikacyjny.
 - 1.5. Konto jest aktywowane po weryfikacji.
-

2. Tworzenie profilu schroniska lub użytkownika

- 2.1. Zalogowany użytkownik wybiera opcję edycji swojego profilu.
 - 2.2. Użytkownik wypełnia informacje:
 - Dla użytkownika: nazwa użytkownika, lokalizacja, numer telefonu, zdjęcie (opcjonalne).
 - Dla schroniska: nazwa schroniska, lokalizacja, numer telefonu, zdjęcie (opcjonalne).
 - 2.3. Profil jest zapisywany i widoczny w aplikacji.
-

3. Dodawanie ogłoszenia adopcyjnego

- 3.1. Schronisko lub użytkownik wybiera opcję "Dodaj ogłoszenie" widoczną na pasku głównym.
- 3.2. Użytkownik wypełnia formularz, wprowadzając dane:
 - Rasa
 - Wiek
 - Wielkość
 - Kolor
 - Lokalizacja
 - Krótki opis psa

- 3.3. Do ogłoszenia dodawane są zdjęcia psa (opcjonalnie).
- 3.4. Ogłoszenie zostaje opublikowane i widoczne w aplikacji.
-

4. Wyszukiwanie psów do adopcji

- 4.1. Użytkownik otwiera aplikację i przechodzi do sekcji wyszukiwania.
- 4.2. Użytkownik wybiera filtry:
- Rasa
 - Wiek
 - Wielkość
 - Kolor
 - Lokalizacja
- 4.3. Po zastosowaniu filtrów użytkownik przegląda listę psów.
- 4.4. Po kliknięciu na ogłoszenie użytkownik widzi szczegóły psa.
-

5. Rejestracja jako wolontariusz

- 5.1. Zalogowany użytkownik znajduje w wyszukiwarce aplikacji schronisko, w którym byłby zainteresowany wolontariatem.
- 5.3. Na stronie profilowej schroniska, użytkownik klika w przycisk "Zostań wolontariuszem".
- 5.2. Użytkownik wypełnia formularz, będący formą kalendarza, na którym może zaznaczyć dni i godziny swojej dostępności.
- 5.3. Dane są zapisywane i udostępniane schronisku.

6. Akceptacja wolontariusza przez schronisko

- 6.1. Na swojej stronie profilowej, pracownik schroniska widzi opcję "Wolontariusze". Po kliknięciu odpala się kalendarz, w którym zaznaczone są dni i godziny, w których jakaś osoba wyraziła chęć zostania wolontariuszem.
- 6.2. Po kliknięciu w datę, w której schronisko potrzebuje wolontariusza, wyświetla się okno dialogowe "Poproś o wolontariat".
- 6.3. Wolontariusz dostaje powiadomienie w aplikacji i musi ponownie zaakceptować wolontariat w danym dniu
- 6.4. Po akceptacji lub jej braku schronisko dostaje stosowny komunikat wraz z linkiem do profilu wolontariusza.
-

7. System donacji dla schronisk (opcjonalne)

- 7.1. Na stronie profilowej schroniska, użytkownik wybiera opcję "Wesprzyj schronisko".
- 7.2. Użytkownik wybiera metodę płatności i przechodzi do płatności.
- 7.3. Po dokonaniu płatności użytkownik otrzymuje potwierdzenie.
-

8. System czatu (wirtualny spacer)

8.1. Użytkownik przegląda ogłoszenie adopcyjne.

8.2. Użytkownik klika opcję "Umów wirtualny spacer".

8.3. Otwiera się okno z wolnymi datami w których są dostępni wolontariusze.

- Istnieje opcja asynchronicznego zadania pytania na czacie.

8.4. Informacja o rozmowie jest zapisywana i widoczna w historii czatu. Wiadomości są zapisywane.

9. Zarządzanie systemem reklam

9.1. Administrator aplikacji dodaje reklamy zewnętrznych reklamodawców.

- Możliwe jest korzystanie z ad providerów (np. [Google Ad Services](#))

9.2. Użytkownicy widzą reklamy w różnych sekcjach aplikacji.

9.3. Reklamy są dostosowane do lokalizacji i zainteresowań użytkownika.

10. Adopcja

10.1. Użytkownik wyświetla ogłoszenie.

10.1. Użytkownik klika przycisk "Adoptuj".

10.2. Użytkownik wypełnia formularz adopcyjny udostępniony przez schronisko.

10.3. Pracownik schroniska / uprawniony przez nich wolontariusz zatwierdza prośbę.

10.4. Użytkownik otrzymuje wiadomość potwierdzeniem. Odblokowuje się opcja wybrania terminu.

10.5. Użytkownik wybiera jeden z podanych terminów.

10.6. Schronisko otrzymuje rezerwację.

10.7. Po terminie spotkania schronisko/wolontariusz aktualizuje status ogłoszenia.

- Użytkownik widząc ogłoszenie ma do dyspozycji dane kontaktowe wystawiającego, więc nic nie stoi na przeszkodzie, aby większość z tych kroków zadziała się poza aplikacją.

B. Projekty ekranów

Logowanie




E-mail/login

Hasło

Zaloguj

[Nie masz jeszcze konta? Zarejestruj się!](#)

Rejestracja



Login

E-mail

Typ konta

Hasło

Powtórz hasło

Zakończ rejestrację

Już prawie!

Jeszcze tylko parę kroków dzieli cię od korzystania z serwisu Psinder! Prosimy o wpisanie poniższych danych.


Nazwa schroniska

Adres schroniska

Numer kontaktowy

Zdjęcie (opcjonalne)

**KLIKNIJ LUB PRZECIĄGNIJ,
ABY PRZESŁAĆ ZDJĘCIE**



Przejdź do serwisu



Dodawanie ogłoszenia

Tytuł

Wprowadź tytuł ogłoszenia (np. Husky 11 lat, Zielona Góra)

Rasa

Wybierz rasę rozwijając listę przyciskiem Znajdź

Znajdź ▼

Wiek

Wprowadź wiek psa lub pozostaw pole puste, jeśli go nie znasz

Wielkość

Wybierz wielkość rozwijając listę przyciskiem Wybierz

Wybierz ▼

Zdjęcie

Przeciągnij zdjęcia z folderu w ten pasek lub kliknij w przycisk dodaj, aby dodawać zdjęcia

Dodaj

Lokalizacja

Ustaw lokalizację zapisaną w profilu ✓

Miejscowość

Kod pocztowy

Ulica

Numer domu

Numer mieszkania

Informacje dodatkowe

Opis

Opowiedz nam o swoim pupilu ☺ Jaki ma charakter? Co lubi robić? Jaka jest jego historia?

Wystaw ogłoszenie



Schronisko XYZ

[Zostań wolontariuszem!](#)

Dane schroniska XYZ, w tym adres, numery kontaktowe itp.

Zdjęcie 1

Ogłoszenie 1 schroniska XYZ

Zdjęcie 2

Ogłoszenie 2 schroniska XYZ...



Formularz wolontariusza

Kalendarz dostępności

01.12.2024	02.12.2024	03.12.2024	04.12.2024	05.12.2024	06.12.2024	07.12.2024
08.12.2024	09.12.2024	10.12.2024	11.12.2024	12.12.2024	13.12.2024	14.12.2024
15.12.2024	16.12.2024	17.12.2024	18.12.2024	19.12.2024	20.12.2024	21.12.2024
22.12.2024	23.12.2024	24.12.2024	25.12.2024	26.12.2024	27.12.2024	28.12.2024
29.12.2024	30.12.2024	31.12.2024				

Poprzedni miesiąc

Następny miesiąc

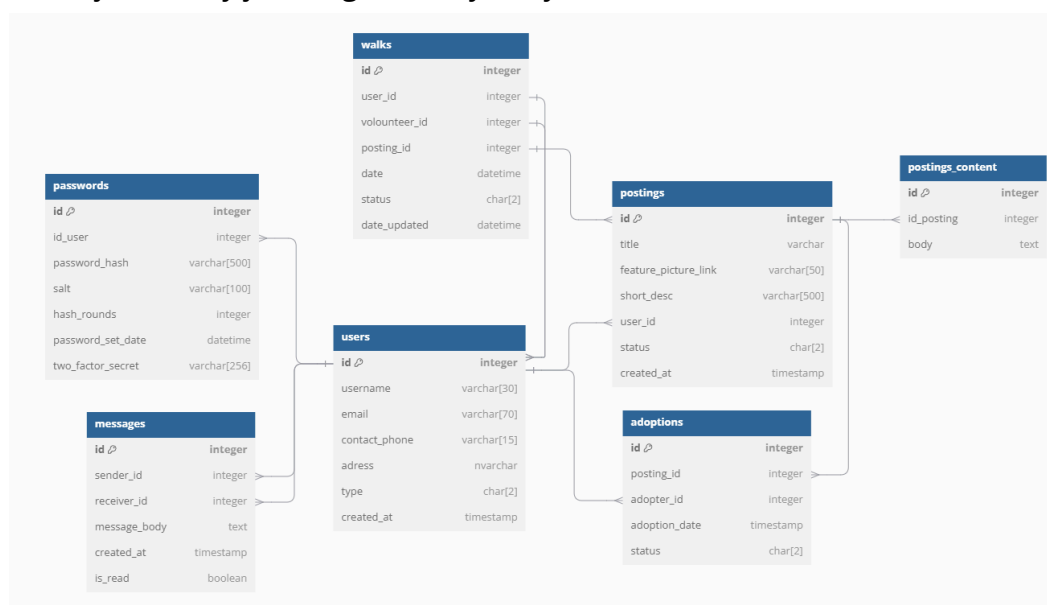
[Zgłoś chęć wolontariatu!](#)

C. Projekt architektury

Aplikacja będzie napisana w JS korzystając z biblioteki React. Narzędziem używanym do pisania kodu będzie Visual Studio Code. Do budowy frontendu użyjemy biblioteki komponentów **MUI**. Do potwierdzania tożsamości będą używane tokeny JWT. Planujemy skorzystać z opcji OAuth2 od Google, Meta oraz Apple. Do dostarczania mediów skorzystamy z systemu CDN od cloudflare.

Baza danych będzie wykorzystywała relacyjny serwer PostgreSQL. Programista będzie mógł bezpośrednio pisać wszystkie zapytania do bazy. Pośrednio do zapytań **SELECT**, **INSERT**, **UPDATE**, **DELETE** będzie miał także użytkownik (rejestracja, aktualizacja swoich danych, usunięcie konta).

Poniżej widoczny jest diagram bazy danych:



D. Główne zasady kodowania

Kod, komentarze i dokumentacja muszą być pisane w języku angielskim. React jest wykorzystywany jako frontend, Node.js (Express.js) jako backend, a PostgreSQL jako baza danych. Docker służy do konteneryzacji, a autoryzacja i uwierzytelnianie opierają się na systemie JWT.

Nazwy klas i komponentów są w PascalCase (np. **AdoptionForm**), zmienne i funkcje w camelCase (np. **fetchDogDetails**), a pliki komponentów w PascalCase z rozszerzeniem **.tsx** (np. **ProfileCard.tsx**). Stałe globalne zapisuje się w UPPER_SNAKE_CASE (np. **API_BASE_URL**). Kod formatuje się z wcięciami na 2 spacje i

maksymalną długością linii 120 znaków. Używane są podwójne cudzysłowy (") w JavaScript. Preferowane są funkcjonalne komponenty React z hookami, a logika biznesowa jest oddzielona od komponentów. Stany globalne zarządza się za pomocą Redux Toolkit.

Kod przechowywany jest w prywatnym repozytorium GitHub. Zmiany wprowadzane są na gałęziach o nazwach `feature/<nazwa-funkcjonalności>` lub `bugfix/<opis-poprawki>`. Merge z gałęzią `main` wymaga pull requestu zatwierdzonego przez recenzenta. Commity są zwięzłe, np. `Add search functionality for dogs by breed`.

Dane uwierzytelnienia przechowywane są w plikach `.env`. ORM Sequelize zapobiega SQL Injection, a zależności są audytowane za pomocą `npm audit` lub `yarn audit`. Testy jednostkowe pisane są w Jest, integracyjne i e2e w Cypress. Nowy kod musi być testowany przed integracją z `main`.

Dokumentacja API tworzona jest za pomocą Swaggera. Wzorzec MVC jest stosowany w backendzie, a Separation of Concerns rozdziela logikę prezentacji od biznesowej. Komponenty React projektuje się zgodnie z zasadą "Smart and Dumb Components".

E. Identyfikacja i zasady zarządzania ryzykiem

Ryzyka jakie mogą wystąpić to:

- **Przeciążenie serwerów, początkowy wzrost kosztów niewspółmierny do zysków z reklam:**

Możliwe przy eksplozji liczby użytkowników. Na początku prawdopodobieństwo jest niskie, lecz z racji poruszanych przez naszą aplikację problemów nastąpi w marketingu punkt, po którym popularność aplikacji eksploduje. Zwiększy się wtedy wielokrotnie liczba pobrań, dodawanych mediów, aktywnych użytkowników. Konieczna jest możliwość natychmiastowego skalowania infrastruktury, aby obsłużyć to zapotrzebowanie gdy tylko nastąpi.

- **Opoźnienie ukończenia aplikacji:**

Prawdopodobieństwo niskie - prowadzony jest ciągły nadzór nad pracami i ich zaawansowaniem. Oczywiście konieczne jest bieżące śledzenie postępów, a w razie wystąpienia poważnych komplikacji może być konieczny crunch lub przesunięcie terminu oddania projektu.

F. Zgodność prac z tablicą koncepcyjną i specyfikacją

Po przeanalizowaniu wykonanej dotychczas pracy i porównaniu jej z tablicą koncepcyjną oraz specyfikacją wymagań możemy stwierdzić, że otrzymane wyniki w pełni spełniają oczekiwania. Nie przewidujemy żadnych znaczących zmian w stosunku do wizji przedstawionej na tablicy koncepcyjnej oraz w specyfikacji wymagań.