



Politechnika
Częstochowska



Wydział Inżynierii
Mechanicznej i Informatyki

PRACA DYPLOMOWA INŻYNIERSKA

System informatyczny wspomagający zarządzanie schroniskami i hotelami dla zwierząt domowych

IT system to support pets shelters and hotels

Michał Naskręcki

Nr albumu: 129867

Kierunek: Informatyka

Forma studiów: niestacjonarne

Poziom studiów: I

Promotor pracy: dr inż. Mariusz Ciesielski

Praca przyjęta dnia:

Podpis promotora:

Częstochowa 2022

Spis treści

1. WSTĘP	5
1.1. SYSTEM INFORMATYCZNY	5
1.2. APLIKACJA INTERNETOWA	5
1.3. CEL I ZAKRES PRACY	6
1.4. PRZEGLĄD ISTNIEJĄCYCH ROZWIĄZAŃ O PODOBNEJ TEMATYCE	7
2. ZASTOSOWANE TECHNOLOGIE INFORMATYCZNE	9
2.1. MONGODB	9
2.2. BOOTSTRAP	10
2.3. REACT	11
2.4. NODE.JS	12
2.5. EXPRESS.JS	13
2.6. SCSS	14
2.7. STRIPE	14
3. PROJEKT SYSTEMU INFORMATYCZNEGO	15
3.1. STRUKTURA PROJEKTU	15
3.2. INTERFEJS UŻYTKOWNIKA	16
3.3. WYMAGANIA FUNKCJONALNE	16
3.4. WYMAGANIA NIEFUNKCJONALNE	17
3.5. DIAGRAM PRZYPADKÓW UŻYCIA	17
3.6. SCHEMAT BAZY DANYCH	19
3.7. PROBLEMY IMPLEMENTACYJNE	19
3.8. SCENARIUSZE PRZYPADKÓW UŻYCIA	20
4. PREZENTACJA PROJEKTU	27
4.1. REJESTRACJA	27
4.2. LOGOWANIE	28
4.3. PANEL UŻYTKOWNIKA	28
4.4. KONFIGURACJA PŁATNOŚCI W PANELU UŻYTKOWNIKA	29
4.5. KONFIGURACJA PŁATNOŚCI W STRIPE	30

4.6. PANEL UŻYTKOWNIKA PO SKONFIGUROWANIU DO WYSTAWIANIA OGŁOSZEŃ	31
4.7. DODAWANIE/EDYCJA OGŁOSZEŃ	31
4.8. STRONA GŁÓWNA.....	32
4.9. FILTROWANIE OGŁOSZEŃ	33
4.10. DOKONYWANIE REZERWACJI.....	34
4.11. SZCZEGÓŁY REZERWACJI.....	34
4.12. OPIEKUNOWIE	35
5. PODSUMOWANIE	37
BIBLIOGRAFIA.....	39
STRESZCZENIE	41
ABSTRACT.....	41

1. Wstęp

1.1. System Informatyczny

Jest to system, w którego skład wchodzi zazwyczaj kilka elementów które przetwarzają dane przy użyciu komputera. W jego skład wchodzi m.in:

- Hardware - są to zazwyczaj komputery jak i również różne akcesoria do przechowywania danych i urządzenia peryferyjne jak i cały szereg urządzeń służących do komunikacji między różnymi elementami bądź akcesoria do przechowywania danych.
- Software - jest to w największym uproszczeniu oprogramowanie. Wszystkie urządzenia jakie użytkujemy, użytkujemy właśnie za pomocą specjalnie napisanego oraz zaimplementowanego pod nieoprogramowania. Jest to kluczowa część całego systemu informatycznego.

Systemy informatyczne stosuje się zazwyczaj w przedsiębiorstwach w celu usprawnienia działań firmy jak i poprawienia komunikacji. Bardzo często do otrzymania odpowiedniego dla celów firmy systemu informatycznego pisze się specjalnie dedykowane pod nie aplikacje dopasowane do potrzeb klienta.

1.2. Aplikacja internetowa

Jest to program komputerowy działający najczęściej za pomocą przeglądarki bądź dedykowanej aplikacji desktopowej lub mobilnej pracujący na serwerze komunikując użytkownika z serwerem.

Zaletami takiego rozwiązania są między innymi bardzo niskie wymagania sprzętowe, gdzie użytkownik by skorzystać z danej aplikacji jedyne czego potrzebuje to zainstalowanej przeglądarki oraz wysoka kompatybilność. Zaletami aplikacji internetowej jest gromadzenie się danych na serwerze, gdzie użytkownik z dowolnego urządzenia może się zalogować i nie gromadzi danych na dysku lokalnie, lecz te są wysyłane wprost do serwera, gdzie są trzymane. W efekcie logując się z innego urządzenia nadal otrzymuje dostęp do danych, które wcześniej gromadził za pomocą swojego profilu. Relatywnie niskie koszty w utrzymaniu są niewątpliwie

dużym atutem. Również łatwość wprowadzania zmian przez dewelopera jest tutaj dużą zaletą, gdyż jest to wykonywane niezależnie od użytkowników danej aplikacji.

Do niewątpliwych wad zaliczamy wymagany stały dostęp do łącza internetowego podczas korzystania z aplikacji i komunikacji z serwerem, gdzie jednocześnie szybkość posiadanego łącza internetowego wpływa na szybkość i wygodę w użytkowaniu aplikacji. Brak bezpośredniej komunikacji z urządzeniami peryferyjnymi. Na przykład, aby wydrukować dany plik należy go uprzednio pobrać na komputer. Kolejną z wad jest mimo wszystko wolniejsze działanie niż w dedykowanej aplikacji okienkowej, dużo mniejsza funkcjonalność, gorsza optymalizacja, mniejsza wygoda w użytkowaniu. Kwestie bezpieczeństwa stanowią także dla deweloperów aplikacji internetowych nie lada wyzwanie. Wycieki informacji mogą być katastrofalne w skutkach, gdzie mogą wyciec bardzo wrażliwe dane różnych użytkowników.

1.3. Cel i zakres pracy

Celem pracy jest zaprojektowanie projektu systemu informatycznego (aplikacji internetowej) służącej do zarządzania hotelami i schroniskami dla zwierząt w za pomocą technologii webowych w postaci MERN (skrót od MongoDB, Express.js, React, Node.js). Aplikacja pozwala użytkownikowi na utworzenie własnego konta, gdzie będzie mieć dostęp do wystawiania ogłoszeń oraz zarezerwować daną usługę w określonym czasie. Aplikacja umożliwia ponadto przy wystawianiu ogłoszenia ustawienia lokalizacji danego obiektu, ustawienia kategorii zwierzęcia, do którego bezpośrednio będzie skierowane ogłoszenie, dodanie zdjęcia obiektu, lub lokalizacji. Ponadto wyświetla saldo danego użytkownika oraz umożliwia wejście w statystyki przychodów danego konta. Kupującemu pozwala na wybranie dat we własnym zakresie oraz podglądu poprzednich zamówień. Aplikacja umożliwia użytkownikowi dokonywania płatności internetowych za pomocą karty debetowej, systemu płatności Blik, Google Pay, Apple Pay, czy innych popularnych serwisów transakcji internetowych takich jak Przelewy24 bądź PayU.

Całość opiera się na architekturze klient-serwer z warstwą frontendową po stronie klienta oraz backendową po stronie serwera. Po stronie klienta znajduje się cała funkcjonalność aplikacji, gdzie klient może utworzyć swoje konto, zalogować się, dokonywać różnych operacji, z kolei serwer rejestruje dane wprowadzone przez klienta i dokonuje ich odpowiedniego zapisu w bazie danych.

1.4. Przegląd istniejących rozwiązań o podobnej tematyce

W Internecie można znaleźć wiele systemów aplikacji hotelowych w tym rozdziale zostanie przedstawione kilka najpopularniejszych systemów oraz porównanie ich funkcjonalności.

Airbnb: Jednym z najpopularniejszych rozwiązań na rynku najmu jest Airbnb. Jest to platforma działająca na bardzo prostej zasadzie dająca możliwość dodatkowego zarobku dla użytkowników poprzez wystawianie ogłoszeń własnych mieszkań czy pokoi dla turystów chcących odwiedzić dane miejsce oraz możliwość rezerwacji przez innych użytkowników, gdzie użytkownik zważając na własną kieszeń może w zależności od dostępnych środków finansowych wynająć „łóżko” w danym miejscu, bądź cały obiekt. W ten sposób serwis Airbnb jest w stanie zarabiać nie posiadając nawet danej nieruchomości, a jedynie pobierając opłatę serwisową od właściciela danej nieruchomości.

Zalety:

- wysoka przejrzystość platformy, minimalizm aplikacji,
- łatwość użytkowania,
- szeroki wybór usług od pokoi jednoosobowych po całe obiekty,
- ustalanie własnej ceny przez danego użytkownika,
- możliwość dostosowania wyszukiwania,
- wysoki poziom ochrony konsumenta,
- elastyczna rezerwacja i możliwości anulacji,

Wady:

- niektórzy sprzedawcy potrafią być nieuczciwi wystawiając ogłoszenia lepiej wyglądające niż są w rzeczywistości,
- opłaty serwisowe potrafią być dosyć wysokie,
- kwestie podatkowe,
- kwestie legalności i licencji: niektóre kraje blokują całkowicie dostęp do Airbnb, inne zaś nakładają różne wymogi licencyjne na gospodarzy,
- jakakolwiek kwestia naprawy zepsutego sprzętu zależy na dobrym gościu właściciela,
- brak profesjonalnej obsługi.

Booking: Innym systemem jest platforma Booking. Jest to jeden z najpopularniejszych serwisów służących do rezerwacji obiektów hotelowych. W przeciwieństwie do Airbnb,

Booking w swojej ofercie bazy noclegowej posiada głównie obiekty hotelowe, pensjonaty czy sanatoria. Jednakże podobnie jak Airbnb jego rola polega na znacznym ułatwieniu komunikacji i transakcji na osi obiekt-klient. Unikalną cechą tej platformy jest potwierdzanie automatyczne potwierdzanie rezerwacji. Dzięki temu zamiast potwierdzać każdego pojedynczego gościa, może on samodzielnie zarezerwować swój pobyt w danym obiekcie. Jest to niezwykle istotne dla zarządzającego danym obiektem, gdyż pozwala na prawidłowe zarządzanie kalendarzem oraz pozwala uniknięcia podwójnej rezerwacji obiektu.

Booking oferuje również możliwości zarządzania obiektem w kwestiach finansowych takich jak przedpłata częściowa lub całościowa kosztu rezerwacji, albo kaucja na poczet potencjalnych zniszczeń, która zostaje zwracana gościowi, gdy ten już opuści dany obiekt i nic nie uszkodzi. Ponadto umożliwia w bardzo rozbudowany sposób ustalać przez zarządzającego obiektem różne promocje czy opłaty za usługi dodatkowe.

Zalety:

- profesjonalna obsługa w przeciwieństwie do Airbnb,
- rozbudowane możliwości konfiguracji dla zarządzającego obiektem,
- rozbudowana opcja dodawania dodatkowych usług,
- ogromna baza hotelowa,
- wysoki standard,
- bardziej rozbudowane wyszukiwanie za pomocą filtrów,
- wysoka wszechstronność.

Wady:

- znacznie wyższe opłaty serwisowe niż w przypadku Airbnb dochodzące nawet do 25% ceny wystawionej przez zarządzającego obiektem,
- dużo mniej przyjazny interfejs dla użytkownika.

Wyżej wymienione rozwiązania posiadają bardzo dużo funkcjonalności, natomiast posiadają też wady. Jednakże w tym przypadku jedną z podstawowych wad jest, iż skupiają się jedynie na samej podróży turysty do miejsca docelowego, a w ogóle nie biorą pod uwagę kwestii, rozwiązania problemów turysty jakie zostawi za sobą w domu w związku z wyjazdem, zwłaszcza w przypadku posiadania zwierzęcia domowego.

2. Zastosowane technologie informatyczne

W tym rozdziale zostały przedstawione technologie zastosowane w systemie. Technologie te są dostępne na licencji “free and open-source”. Gwarantuje to dostęp do używanych technologii i możliwości rozwoju aplikacji przez deweloperów. Główne cechy to:

- darmowy dostęp,
- popularność wśród użytkowników,
- dostęp do szeroko dostępnej dokumentacji,
- częste aktualizacje,
- otwarty kod źródłowy,
- szeroki dostęp do dodatkowych bibliotek.

Najwięcej czasu pracy software developera zajmuje rozwiązywanie problemów, jednakże bardzo często zdarza się, że ktoś inny rozwiązał na podobny problem w czasie realizowania swojego projektu. Jedną z kluczowych decyzji o zastosowaniu nowych technologii jest fakt ogromnej społeczności internetowej, gdzie wpisanie odpowiednich fraz w przeglądarce umożliwia szybko znaleźć rozwiązanie danego problemu. Dodatkowym czynnikiem jest fakt, iż dzisiaj rozwijane technologie niejako same podpowiadają w przypadku problemu, co jest nie tak i gdzie należy szukać problemu. Jest to ogromna zaleta, gdyż projektując wielkie projekty w zespołach bardzo ułatwia to pracę, gdzie występuje bardzo często przed samą kompilacją ostrzeżenie w składni kodu.

Zastosowane w projekcie technologie noszą nazwę MERN [1], jest to skrót od MongoDB [2], Express.js [3], React [4] i Node.js [5] (Czasami nosi nazwę MEAN, gdzie Angular zastępuje React). Jest to zbiór obecnie jednych z najbardziej popularnych technologii webowych służących do tworzenia aplikacji. Posiadając bazę użytkowników i stałe wsparcie deweloperów oraz rozwijanie przez samych użytkowników posiada możliwości rozwojowe co do dalszego rozwoju obecnych bądź przyszłych projektów.

2.1. MongoDB

Technologią bazodanową wykorzystaną w projekcie jest MongoDB. MongoDB to baza dokumentów NoSQL, jest to najpopularniejsze obecnie dostępne rozwiązanie NoSQL. Jest to

narzędzie oparte na dokumentach typu open source służące do przechowywania dużych ilości danych. Bazy NoSQL od relacyjnych baz danych SQL różnią się tym, że zamiast zapisywać dane w formie w stałej liczbie wierszy i kolumn przechowują je w dokumentach JSON lub klucz-wartość, oraz zamiast skalować się wertykalnie(pionowo) skalują się horyzontalnie(poziomo), dzieląc się na mniejsze fragmenty, więc poprzez zwiększenie liczby serwerów, z tego względu idealnie nadają się do projektów wykorzystujących duże ilości danych.

Zalety baz MongoDB:

- łatwe środowisko i szybka konfiguracja,
- elastyczność - schemat nie jest predefiniowany, oznacza to, że bazy MongoDB posiadają dynamiczną architekturę schematu,
- fragmentacja - bazy MongoDB używają „shardingu” podczas obsługi dużych obiektów. Sharding to proces dzielenia danych z dużego zestawu i dystrybucji ich na wiele serwerów. W przypadku problemu, gdy serwer nie może obsłużyć danych ze względu na ich rozmiar, automatycznie dzieli je dalej bez wstrzymywania aktywności,
- skalowalność - używanie shardingu znacząco zwiększa pojemność baz,
- rozbudowana dokumentacja,

Wady:

- baza nie jest zaprojektowana do obsługi operacji JOIN,
- ograniczony rozmiar do 16MB,
- duplikacja danych,
- wysokie wykorzystanie pamięci - przez brak łączy, dane się duplikują doprowadzając do wzrostu wykorzystania pamięci,

2.2. Bootstrap

CSS-owy [6] framework służący do tworzenia responsywnych [6] aplikacji webowych na bazie siatki w podejściu mobile-first. Korzysta jednocześnie z HTML [6], CSS jak i JavaScript [7] dodając kilka usprawnień. Umożliwia stronie internetowej lub aplikacji wykrywanie rozmiaru i orientacji ekranu odwiedzającego oraz automatyczne dostosowywanie wyświetlacza.

Jego główną zaletą jest przyspieszenie czasu pracy developera. Pozwala zachować większą czystość w kodzie i rozwiązuje wiele problemów z kompatybilnością dla wielu przeglądarek.

Został wydany przez programistów Twittera i jest obecnie najpopularniejszym tego typu frameworkiem na świecie. Zawiera w sobie wiele nowych elementów takich jak:

- paski postępu,
- okna dialogowe,
- formularze,
- karty,
- okna pop-up,
- zestaw przycisków.

2.3. React

React to obecnie najpopularniejsza biblioteka JavaScriptowa, choć trwa dyskusja czy jest to nadal biblioteka, czy to już framework. React przede wszystkim pozwala na tworzenie rozbudowanych i szybkich interfejsów użytkownika. Jest przy tym łatwiejszym w obsłudze niż Angular, framework wydany przez Google do tworzenia aplikacji front-endowych.

React pozwala na dzielenie kodu na komponenty, co jest zaletą tej biblioteki, gdyż przy budowie dużych projektów, gdzie praca przez wiele osób nad jednym plikiem byłaby uciążliwa i z czasem rozrostu aplikacji bardzo nieczytelne. Code splitting (pol. dzielenie kodu) na poziomie komponentów przynosi jeszcze inne zalety. Podczas ładowania strony z jednego pliku wszystkie komponenty zostaną załadowane nawet gdy nie są potrzebne użytkownikowi w danym momencie, a jedynie obciążają aplikację.

Poprzez dzielenie kodu na komponenty można załadować użytkownikowi jedynie te komponenty, które są w danym momencie potrzebne i znacznie uwydatnić pracę aplikacji. Można to wręcz porównać do budowania aplikacji poprzez tworzenie i składanie klocków.

Kilka znaczących usprawnień wprowadzonych przez React to:

- VirtualDOM - aplikacja po wprowadzeniu zmian jest renderowana ponownie, sprawdza różnicę pomiędzy poprzednim DOM (Document Object Model) i nowym, następnie jest aktualizowany jedynie ze zmianami,
- JSX - jest to skrót od JavaScript XML i pozwala nam pisać HTML w kodzie React. Nie jest to wymagane, lecz znacznie usprawnia pracę,
- renderowanie po stronie serwera SSR.

Kryteriami, które zaważyły na użyciu tej technologii do implementacji systemu są takie, iż jest to technologia języka JavaScript, który jest obecnie najpopularniejszym narzędziem do tworzenia stron i aplikacji internetowych. Jest to stosunkowo prosty język, którego można się nauczyć bardzo szybko i jest on również jednym z zainteresowań autora. Bardzo wysoka wydajność i szybkość działania aplikacji w tej technologii jest perfekcyjnym narzędziem do tego typu projektów. Z react korzysta obecnie wiele dużych firm takich jak Meta (dawniej Facebook), Netflix, Dropbox, Airbnb czy Uber. Jest przy tym wszystkim wysoce skalowalny, gdzie wdrażając tysiące komponentów na raz nadal nie występują problemy z wydajnością aplikacji. Wszystko to daje duże możliwości rozwoju dla dalszego rozwoju aplikacji, jednocześnie zmniejszając znacząco koszty produkcji.

2.4. Node.js

Jest to środowisko uruchomieniowe JavaScript, służące do tworzenia aplikacji sieciowych wykonując zapytania po stronie serwera. Pozwala to tworzyć całe aplikacje w obrębie jednego języka zamiast korzystania z zewnętrznych rozwiązań. Z Node.js obecnie korzysta wiele znanych marek takich jak LinkedIn, IBM, Microsoft, Netflix, Airbnb czy AWS (Amazon Web Services). Posiada bogatą bibliotekę modułów npm, która znacznie ułatwia budowanie aplikacji internetowych i jest rozbudowywana przez użytkowników. Różnicą między Node.js, a PHP lub ASP jest fakt, iż Node.js pomija oczekiwanie na odpowiedź i wysyła od razu następne zapytanie. Node służy do generowania dynamicznej zawartości strony, dodawania, usuwania lub edycji danych w bazie danych lub potrafi zbierać dane formularzy czy pracować na plikach na serwerze.

Cechy Node.js to:

- asynchroniczne i sterowane zdarzeniami - Wszystkie interfejsy API biblioteki Node.js są asynchroniczne. Oznacza to, że serwer oparty na Node.js nigdy nie czeka na zwrócenie danych przez API. Serwer przechodzi do następnego API po jego wywołaniu, a mechanizm powiadamiania o zdarzeniach Node.js pomaga serwerowi uzyskać odpowiedź z poprzedniego wywołania API,
- szybkość działania,
- wysoce skalowalny - Node.js używa modelu jednowątkowego z pętlą zdarzeń. Mechanizm zdarzeń pomaga serwerowi odpowiadać w sposób nie blokujący oraz

sprawia, że serwer jest wysoce skalowalny w przeciwieństwie do tradycyjnych serwerów, które tworzą ograniczone wątki do obsługi żądań. Node.js wykorzystuje jednowątkowy program i ten sam program może obsłużyć znacznie większą liczbę żądań niż tradycyjne serwery, takie jak Apache HTTP Server,

- brak buforowania - aplikacje Node.js nie buforują danych, a wyprowadzają dane w porcjach.

Kryterium jakie zdecydowało nad zastosowaniem tej technologii do projektu była chęć zebrania wszystkiego wokół jednego języka programowania, co pozwala na dużą oszczędność czasu i zasobów. Dodatkowo szybkość działania jest tutaj zdecydowanie ważnym kryterium.

2.5. Express.js

Express.js jest to framework Node.js przeznaczony do szybkiego tworzenia interfejsu API do aplikacji internetowych oraz mobilnych. Node.js jak już wspomniano jest środowiskiem uruchomieniowym JavaScript służącym do tworzenia aplikacji po stronie serwera. Jednakże nie wie, jak obsługiwać pliki, żądania i metody HTTP, dlatego właśnie został stworzony Express.js. Jest to narzędzie bardzo proste w użyciu, aby szybko i łatwo pisać aplikacje internetowe co w innym wypadku zajmuje bardzo dużo czasu.

Zalety:

- asynchroniczność,
- oszczędność czasu,
- zmniejszenie kosztów,
- możliwość napisania aplikacji w obrębie jednego języka.

Kilka usprawnień, które wprowadza Express.js tj:

- szablony,
- routing - służy do tego jak adresy URL endpointów odnoszą się do zapytań klienta,
- middleware - obsługa zapytań, która ma dostęp do cyklu zapytanie-odpowiedź.

Jest to kolejne rozwiązanie zastosowane w projekcie, aby napisać wszystko w obrębie jednego języka.

2.6. SCSS

Składnia SASS (Syntactically Awesome Style Sheets) [8], preprocesora języka CSS, wprowadza on mnóstwo usprawnień pozwalających na dużo lepszą przejrzystość i organizację kodu. Obecnie jest to standardowe narzędzie wykorzystywane przez programistów do pisania kaskadowych arkuszy stylów. Jest w pełni kompatybilny z CSS, więc programista piszący kod w zwykłym CSS pisze go poprawnie, nie mając obowiązku korzystać ze składni SCSS.

Kluczowe usprawnienia wprowadzone przez SCSS:

- zmienne,
- wprowadzenie instrukcji takich jak mixins czy include,
- zagnieżdżenia,
- dziedziczenie stylów za pomocą extends,
- funkcje,
- pętle.

2.7. Stripe

Stripe [9] to dostawca usług płatniczych, Stripe jest wykorzystywany do płatności internetowych obsługujących karty płatnicze, karty kredytowe, portfele cyfrowe. Jest to narzędzie, z którego korzystają właściciele biznesów do zapewnienia możliwości płatności klientom za dany produkt lub usługę. Posiada obsługę najpopularniejszych serwisów płatności takich jak PayU, Przelewy24, Google Pay, Apple Pay i wiele innych. Jest używany przez wiele znanych firm, na przykład: Amazon, Spotify, Uber, Booking, Shopify. Stripe to skontrolowany i certyfikowany dostawca usług PCI Level 1, co oznacza, że musi przejść roczny raport zgodności oraz rutynowe skanowanie i testy bezpieczeństwa. Stripe szyfruje numery kart kredytowych wszystkich klientów i osobno przechowuje informacje o odszyfrowaniu, co oznacza, że Stripe nie widzi numerów kart kredytowych. Ponadto Stripe wymaga, aby wszystkie transakcje online odbywały się za pośrednictwem sieci HTTPS.

3. Projekt systemu informatycznego

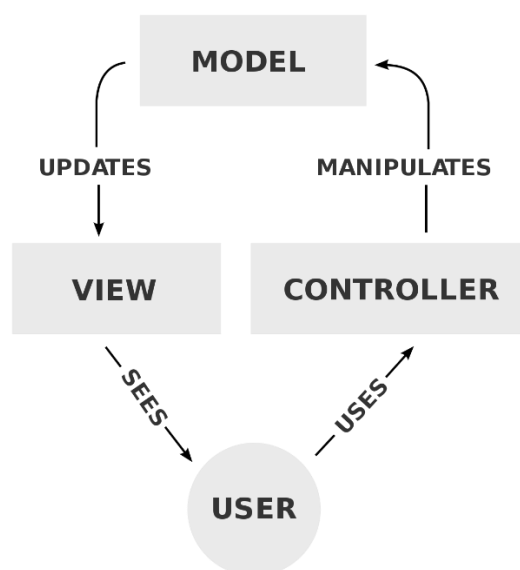
W niniejszym rozdziale zostały przedstawione wybrane elementy całego projektu, wymagania stawiane przed projektem, omawiane są wymagania funkcjonalne i нефункционалне całego projektu. Następnie rodzaje użytkowników, ich funkcje i możliwości. Diagramy przypadków użycia oraz scenariusze przypadków użycia.

3.1. Struktura projektu

Struktura projektu opiera się o wzorzec projektowy zwany: MVC [10] - Model, Widok, Kontroler (*ang. Model, View, Controller*). Jest to wzorzec projektowy, który stosuje się do podziału aplikacji na trzy moduły.

- Model - odpowiedzialny za dostarczanie danych w projekcie
- Widok - odpowiedzialny za dostarczanie treści użytkownikowi
- Kontroler - odpowiedzialny za interakcję użytkownika z systemem. Przyjmuje on zapytania od użytkownika i przekazuje je dalej do modelu oraz zarządza widokiem

Na Rys. 1. Przedstawiono wzorzec MVC.



Rysunek 1 Wzorzec Model View Controller

3.2. Interfejs użytkownika

Interfejs użytkownika jest oparty na operacjach **CRUD** (ang. Create, Read, Update, Delete). Interfejs użytkownika został stworzony za pomocą JavaScriptowej biblioteki React, a za wygląd aplikacji odpowiadała CSSowa biblioteka Bootstrap wraz z domieszką CSS napisanego za pomocą preprocesora SASS w składni SCSS oraz biblioteki Antd [14] oraz Algolia [13]. Również do dodania niektórych elementów wykorzystano zestaw narzędzi Font Awesome [11] dodający dodatkowe ikony do wyglądu strony.

Dodawanie elementów do bazy odbywa się za pomocą funkcji **Create** i pozwala nam na utworzenie nowego dokumentu w bazie danych.

Odczyt odbywa się za pomocą wywołania funkcji **Read** i pozwala na sprawdzenie czy dane zostały poprawnie zapisane w bazie i wyświetlanie danych użytkownikowi.

Edytowanie wywołuje się za pomocą funkcji **Update** i pozwala nadpisać stworzony wcześniej dokument o nowe zmienne i przekazuje do bazy danych

Usuwanie odbywa się za pomocą wywołania funkcji **Delete** i jak sama nazwa wskazuje daje możliwość usunięcia elementu.

3.3. Wymagania funkcjonalne

1. założenie przez użytkownika własnego konta,
2. logowanie się do aplikacji za pomocą hasła i podanego wcześniej adresu email,
3. dodawanie przez użytkownika ogłoszenia,
4. możliwość edycji własnego ogłoszenia,
5. widok salda własnego konta,
6. możliwość sprawdzenia statystyk przychodów,
7. wybranie dowolnej lokalizacji na całym świecie,
8. możliwość dodania ceny z polskiej walucie za dobę,
9. podgląd poprzednich opłaconych zamówień z dokładnymi danymi co do zrealizowanego zamówienia,
10. wyszukiwanie ogłoszeń z danego miejsca,
11. wyszukiwanie ogłoszeń ze względu na kategorię,
12. wyszukiwanie ogłoszeń za pomocą kategorii i miejsca jednocześnie,

13. ustalanie dat przy opłacie danego zamówienia,
14. automatyczne zliczanie końcowej ceny dla użytkownika licząc cenę wystawioną przez wystawiającego dane ogłoszenie jak i zakres dat, który wybrał użytkownik,
15. możliwość usuwania ogłoszenia,
16. dodawanie zdjęć do ogłoszenia,
17. komunikaty przy udanej akcji użytkownika lub nieudanej takiej jak dodanie ogłoszenia, edycja, usunięcie, logowanie etc,
18. wybór przez klienta godzin karmienia, rodzaju karmy i dodatkowego świadczenia medycznego,
19. wprowadzenie danych zwierzęcia do bazy danych,
20. przypisanie do zwierzęcia właściciela jak i opiekuna na czas pobytu,
21. dodanie opiekuna do bazy,
22. usunięcie opiekuna z bazy.

3.4. Wymagania niefunkcjonalne

1. Aplikacja powinna korzystać z szeroko dostępnych darmowych rozwiązań do celów komercyjnych.
2. Responsywność na różnych rozdzielczościach na urządzeniach mobilnych i urządzeniach desktopowych,
3. Poprawne działanie na najpopularniejszych przeglądarkach internetowych tj:
 - Google Chrome,
 - Safari,
 - Microsoft Edge,
 - Mozilla Firefox.

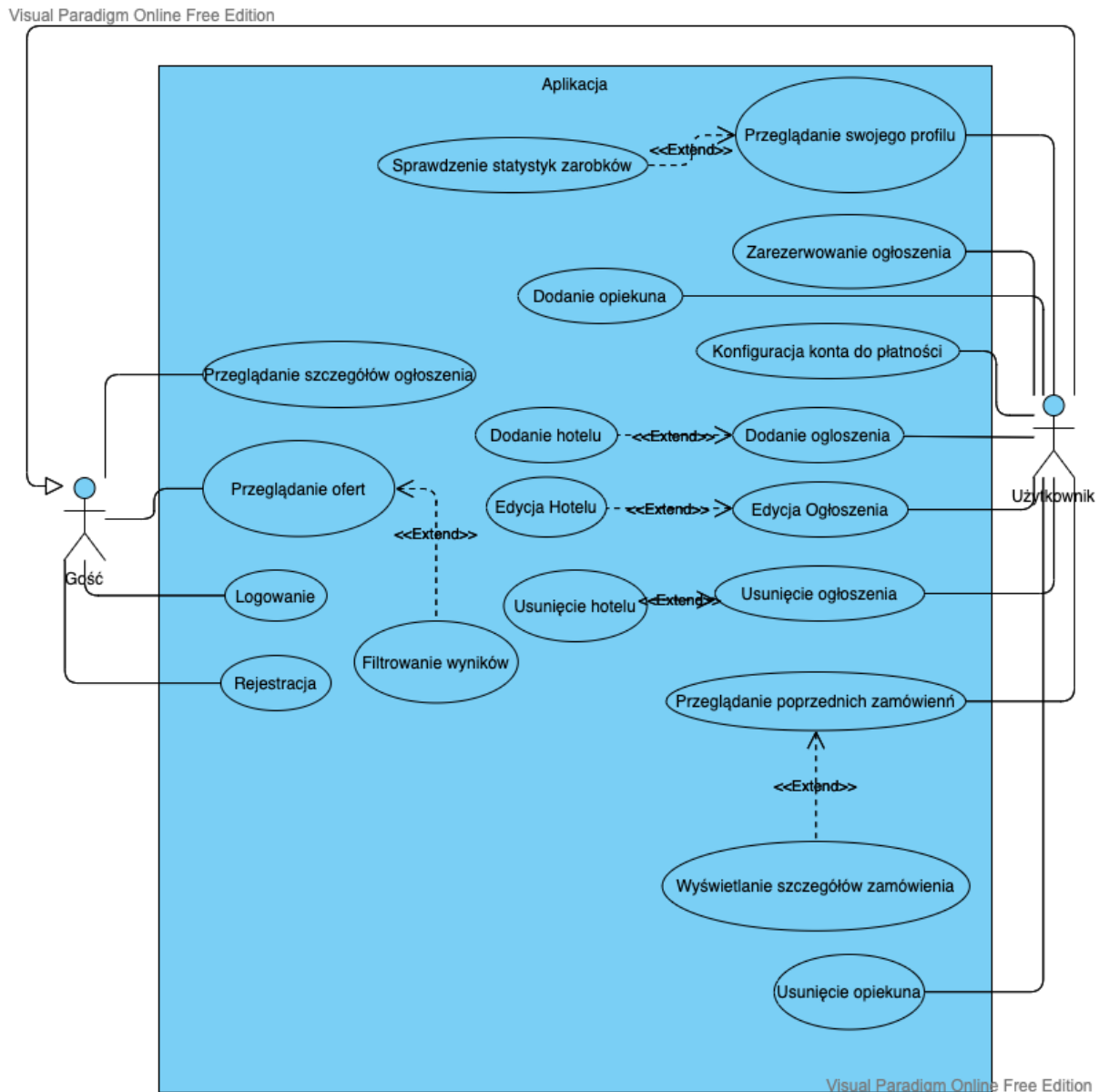
3.5. Diagram przypadków użycia

W projekcie jest dwóch aktorów:

1. Gość - Aktor, który bez zalogowania się do aplikacji może głównie przeglądać wystawione ogłoszenia lub założyć konto w aplikacji, albo zalogować się.

2. Użytkownik - Aktor, który po uprzednim założeniu konta i zalogowaniu się ma dostęp do wszystkich funkcjonalności aplikacji. Może dokonać rezerwacji wystawionego ogłoszenia, a po odpowiednim skonfigurowaniu swojego konta może zacząć wystawiać własne, edytować je, usuwać i sprawdzać na bieżąco statystyki swojego konta. Może także dodawać i usuwać opiekunów.

Diagram został przedstawiony na Rys. 2.

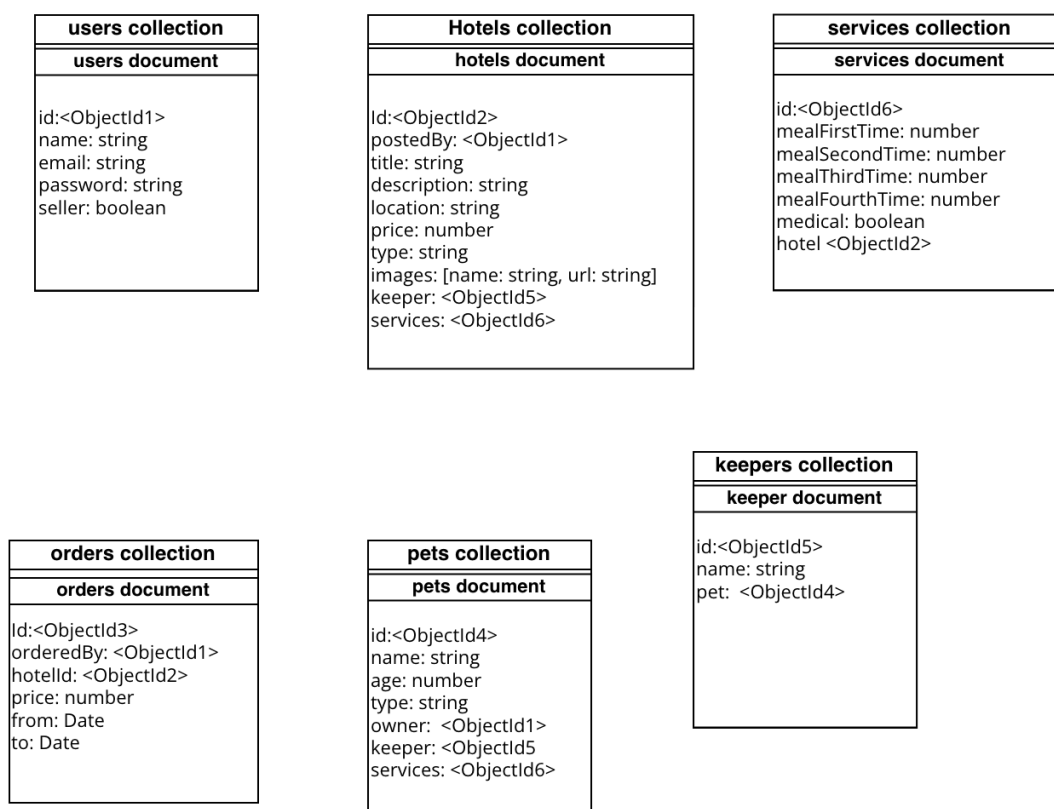


Rysunek 2 Diagram przypadków użycia

3.6. Schemat bazy danych

W projekcie została zastosowana NoSQL'owa baza danych MongoDB, która przechowuje dane w postaci dokumentów typu JSON.

W bazie zdefiniowano następujące kolekcje: Użytkownicy, hotele i zamówienia. W każdej z nich zawarto odpowiednie dokumenty o danych polach. Schemat został przedstawiony na Rys. 3.



Rysunek 3 Schemat bazy danych

3.7. Problemy implementacyjne

W trakcie tworzenia aplikacji pojawiło się kilka problemów implementacyjnych. Pierwszym z nich był problem unikalności nazwy użytkownika. Użytkownik tworząc konto podawał swój adres e-mail oraz nazwę użytkownika, zdecydowano więc by nazwa nie była wymagana unikalna, gdyż może on posługiwać się swoim imieniem, które wśród wielu użytkowników może się powtarzać. Zamiast tego został wprowadzony wymóg unikalnego

adresu e-mail, który jest już unikalny i za pomocą prostej instrukcji warunkowej, która sprawdza za pomocą zmiennej czy w bazie danych istnieje dany adres e-mail udało się rozwiązać ten problem.

Drugim problemem implementacyjnym w procesie tworzenia aplikacji był problem z zarządzaniem obsługą danego zwierzęcia podczas pobytu w danym hotelu przez czas rezerwacji danego ogłoszenia, gdyż użytkownik rezerwujący jedynie opłacał daną rezerwację nie wprowadzając żadnych informacji co do swojego zwierzęcia, co powodowało problemy jak sprostac oczekiwaniom rezerwacji i rozwiązać problem organizacyjnie. Postanowiono dodać w bazie nowe dokumenty: zwierzęcia, opiekuna i usług, a także wprowadzić wymóg uzupełnienia przez rezerwującego dane ogłoszenie krótkiego formularza

3.8. Scenariusze przypadków użycia

W Tabelach 1-16 przedstawiono wybrane scenariusze przypadków użycia w systemie.

Tab. 1 Przeglądanie ofert

Nazwa	Przeglądanie ofert
Aktorzy	Gość, Użytkownik
Warunki początkowe	Aktor znajduje się na stronie startowej
Opis	Aktor zamierza przejrzeć ogłoszenia zamieszczone na stronie
Ścieżka główna	Aktor po wejściu na stronę główną przegląda ogłoszenia przechodząc w dół lub w górę
Efekt końcowy	Aktor znajduje się na stronie z ogłoszeniami

Tab. 2 Filtrowanie wyników

Nazwa	Filtrowanie wyników
Aktorzy	Gość, Użytkownik
Warunki początkowe	Aktor znajduje się na stronie startowej
Opis	Aktor zamierza przefiltrować ogłoszenia, wedle własnych wytycznych

Ścieżka główna	<ol style="list-style-type: none"> 1. Aktor wybiera interesujące go filtry 2. Aktor filtruje wyniki za pomocą przycisku "szukaj"
Efekt końcowy	Aktor znajduje wybrane ogłoszenia według filtrów

Tab. 3 Przeglądanie szczegółów ogłoszenia

Nazwa	Przeglądanie szczegółów ogłoszenia
Aktorzy	Gość, Użytkownik
Warunki początkowe	Aktor znajduje się na stronie startowej
Opis	Aktor zamierza wejść w szczegóły wystawionego ogłoszenia
Ścieżka główna	Aktor naciska przycisk „Zobacz więcej”
Ścieżka Alternatywna	<ul style="list-style-type: none"> • Aktor naciska na zdjęcie przy ogłoszeniu • Aktor naciska na tytuł ogłoszenia
Efekt końcowy	Aktor wchodzi w szczegóły ogłoszenia

Tab. 4 Logowanie

Nazwa	Logowanie
Aktorzy	Gość
Warunki początkowe	Aktor znajduje się na stronie startowej
Opis	Aktor zamierza zalogować się do swojego konta
Ścieżka główna	<ol style="list-style-type: none"> 1. Aktor naciska centralny przycisk menu nawigacji z napisem „Zaloguj się” 2. Aktor zostaje przekierowany do panelu logowania 3. Aktor wypełnia pola dotyczące adresu email i hasła 4. Aktor naciska przycisk „zaloguj” lub naciska enter”
Ścieżka Alternatywna	<ul style="list-style-type: none"> • Aktor wpisuje niepoprawne dane użytkownika i zostaje wyświetlony odpowiedni komunikat • Aktor wpisuje dane, lecz te zostają nie odnalezione w bazie

	danych, więc zostaje wyświetlony odpowiedni komunikat
Efekt końcowy	Aktor zalogował się do swojego konta

Tab. 5 Rejestracja

Nazwa	Rejestracja
Aktorzy	Gość
Warunki początkowe	Aktor znajduje się na stronie startowej
Opis	Aktor zamierza utworzyć konto w aplikacji
Ścieżka główna	<ol style="list-style-type: none"> 1. Aktor naciska przycisk menu nawigacji po prawej stronie z napisem „Zarejestruj się” 2. Aktor wypełnia pola dotyczące adresu email, nazwy i hasła, a następnie zatwierdza przyciskiem.
Ścieżka Alternatywna	<ul style="list-style-type: none"> • Aktor wprowadził dane nie spełniające wymagań utworzenia konta • Aktor wprowadził dane, które uprzednio znajdowały się w bazie
Efekt końcowy	Konto zostaje utworzone

Tab. 6 Przeglądanie własnego profilu

Nazwa	Przeglądanie własnego profilu
Aktorzy	Użytkownik
Warunki początkowe	Aktor zalogował się na własne konto
Opis	Aktor zamierza wejść w panel swojego konta
Ścieżka główna	Aktor po klika przycisk znajdujący się w centralnym miejscu menu nawigacyjnego
Efekt końcowy	Aktor zostaje przekierowany do panelu użytkownika

Tab. 7 Konfiguracja konta do płatności

Nazwa	konfiguracja konta do płatności
Aktorzy	Użytkownik
Warunki początkowe	Aktor po zalogowaniu się w panelu użytkownika przechodzi do zakładki „Twoje ogłoszenia”
Opis	Aktor zamierza skonfigurować konto do płatności, aby móc wystawiać ogłoszenia
Ścieżka główna	<ol style="list-style-type: none"> 1. Aktor po przejściu do odpowiedniej zakładki naciska przycisk „skonfiguruj płatności” 2. Aktor przechodzi do konfiguracji konta w stripe i wypełnia wszystkie kroki wedle instrukcji
Efekt końcowy	Aktor skonfigurował swoje konto i może wystawiać zlecenia

Tab. 8 Sprawdzanie statystyk zarobków

Nazwa	Sprawdzenie statystyk zarobków
Aktorzy	Użytkownik
Warunki początkowe	Aktor posiada konto skonfigurowane do płatności w Stripe
Opis	Aktor zamierza wejść w statystyki swoich zarobków na przestrzeni czasu
Ścieżka główna	<ol style="list-style-type: none"> 1. Aktor przechodzi do panelu użytkownika 2. Aktor w panelu użytkownika klika ikonę zębatki
Efekt końcowy	Aktor przechodzi do statystyk zarobków swojego konta w Stripe

Tab. 9 Zarezerwowanie ogłoszenia

Nazwa	Zarezerwowanie ogłoszenia
Aktorzy	Użytkownik

Warunki początkowe	Aktor wszedł w szczegóły ogłoszenia
Opis	Aktor zamierza złożyć rezerwację danego ogłoszenia
Ścieżka główna	<ol style="list-style-type: none"> 1. Aktor po wypełnieniu danych naciska przycisk „zarezerwuj” lub enter 2. Po przejściu do płatności aktor opłaca swoje zamówienie
Ścieżka Alternatywna	<ul style="list-style-type: none"> • Aktor wpisał niepoprawne dane przy rezerwacji
Efekt końcowy	Aktor złożył rezerwację

Tab. 10 Dodanie ogłoszenia

Nazwa	Dodanie ogłoszenia
Aktorzy	Użytkownik
Warunki początkowe	Aktor ma skonfigurowane konto do dodawania ogłoszeń
Opis	Aktor zamierza dodać swoje ogłoszenie
Ścieżka główna	<ol style="list-style-type: none"> 1. Aktor naciska przycisk „Dodaj nowe” w zakładce „Twoje ogłoszenia” 2. Aktor wypełnia wszystkie wymagane pola: Tytuł, opis, zdjęcie, cena, typ zwierzęcia 3. Aktor naciska przycisk „dodaj” lub enter
Ścieżka Alternatywna	Aktor nie wprowadził danych w wymaganym polu, program zwróci błąd
Efekt końcowy	Aktor dodał ogłoszenie

Tab. 11 Edytowanie ogłoszenia

Nazwa	Edytowanie ogłoszenia
Aktorzy	Użytkownik
Warunki początkowe	Aktor znajduje się w zakładce twoje ogłoszenia

Opis	Aktor zamierza edytować swoje ogłoszenie
Ścieżka główna	<ol style="list-style-type: none"> 1. Aktor naciska ikonę edycji wybranego ogłoszenia 2. Aktor edytuje wybrane pola 3. Aktor zatwierdza zmiany klikając przycisk „zapisz lub enter”
Ścieżka Alternatywna	Aktor nie wprowadził danych w wymaganym polu, program zwróci błąd
Efekt końcowy	Aktor edytował swoje ogłoszenie

Tab. 12 Usunięcie ogłoszenia

Nazwa	Usunięcie ogłoszenia
Aktorzy	Użytkownik
Warunki początkowe	Aktor znajduje się w zakładce twoje ogłoszenia
Opis	Aktor zamierza usunąć swoje ogłoszenie
Ścieżka główna	<ol style="list-style-type: none"> 1. Aktor naciska ikonę usunięcia wybranego ogłoszenia 2. Aktor zatwierdza zmiany klikając przycisk „zapisz lub enter”
Efekt końcowy	Aktor usunął swoje ogłoszenie

Tab. 13 Przeglądanie poprzednich zamówień

Nazwa	Przeglądanie poprzednich zamówień
Aktorzy	Użytkownik
Warunki początkowe	Aktor znajduje się w panelu użytkownika
Opis	Aktor zamierza przejrzeć swoje poprzednie rezerwacje
Ścieżka główna	Aktor naciska odpowiedni przycisk z napisem „Twoje zamówienia”
Efekt końcowy	Aktor trafia do zakładki z poprzednimi zamówieniami

Tab. 14 Szczegóły ogłoszenia

Nazwa	Szczegóły zamówienia
-------	----------------------

Aktorzy	Użytkownik
Warunki początkowe	Aktor znajduje się w zakładce „Twoje zamówienia”
Opis	Aktor zamierza sprawdzić szczegóły zrealizowanego zamówienia
Ścieżka główna	Aktor po przejrzaniu zrealizowanych zamówień naciska na przycisk „zobacz więcej” przy danym ogłoszeniu
Efekt końcowy	Aktorowi wyświetla się okno pop-up ze szczegółami danej rezerwacji.

Tab. 15 Dodanie opiekuna

Nazwa	Dodanie opiekuna
Aktorzy	Użytkownik
Warunki początkowe	Aktor znajduje się w zakładce „Twoi opiekunowie”
Opis	Aktor zamierza dodać opiekuna do bazy
Ścieżka główna	Aktor naciska przycisk „Dodaj opiekuna”, a następnie wypełnia pole do wypełnienia i zatwierdza
Efekt końcowy	Dodano opiekuna

Tab. 16 Usunięcie opiekuna

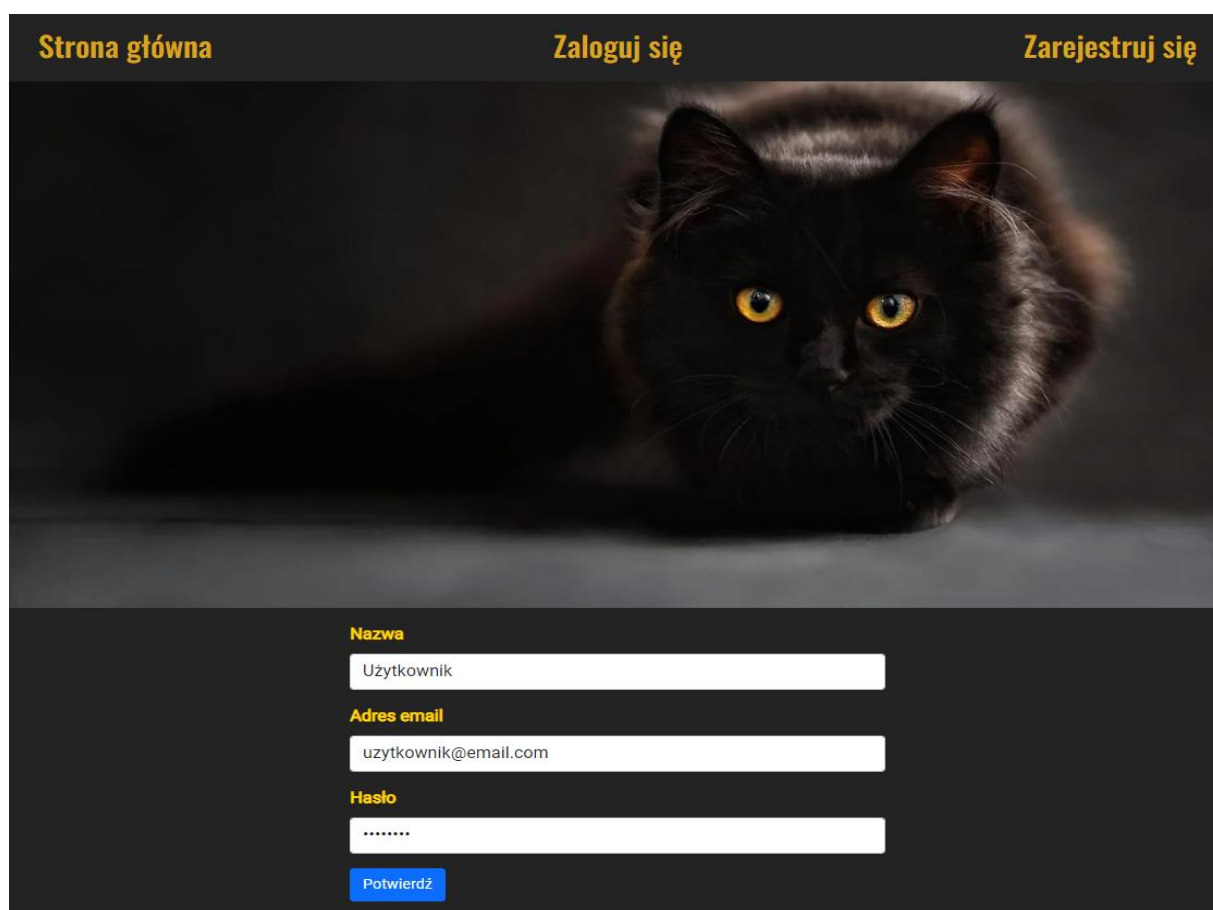
Nazwa	Usunięcie opiekuna
Aktorzy	Użytkownik
Warunki początkowe	Aktor znajduje się w zakładce „Twoi opiekunowie”
Opis	Aktor zamierza usunąć opiekuna z bazy
Ścieżka główna	Aktor naciska ikonę usunięcia przy wybranym opiekunie i zatwierdza
Efekt końcowy	usunięto opiekuna

4. Prezentacja projektu

Rozdział ten krok po kroku z grafikami prezentuje funkcjonowanie systemu z podglądem w celu przedstawienia działania zaimplementowanych funkcjonalności.

4.1. Rejestracja

Rejestracja umożliwia utworzenie konta w aplikacji poprzez wypełnienie wymaganych pól w Formularzu. Jeśli dane nie spełniają danych wymagań lub dany adres email jest już zajęty w bazie, to program pokaże odpowiednią informację. Na Rys. 4. pokazano funkcjonowanie ekranu rejestracji w systemie.

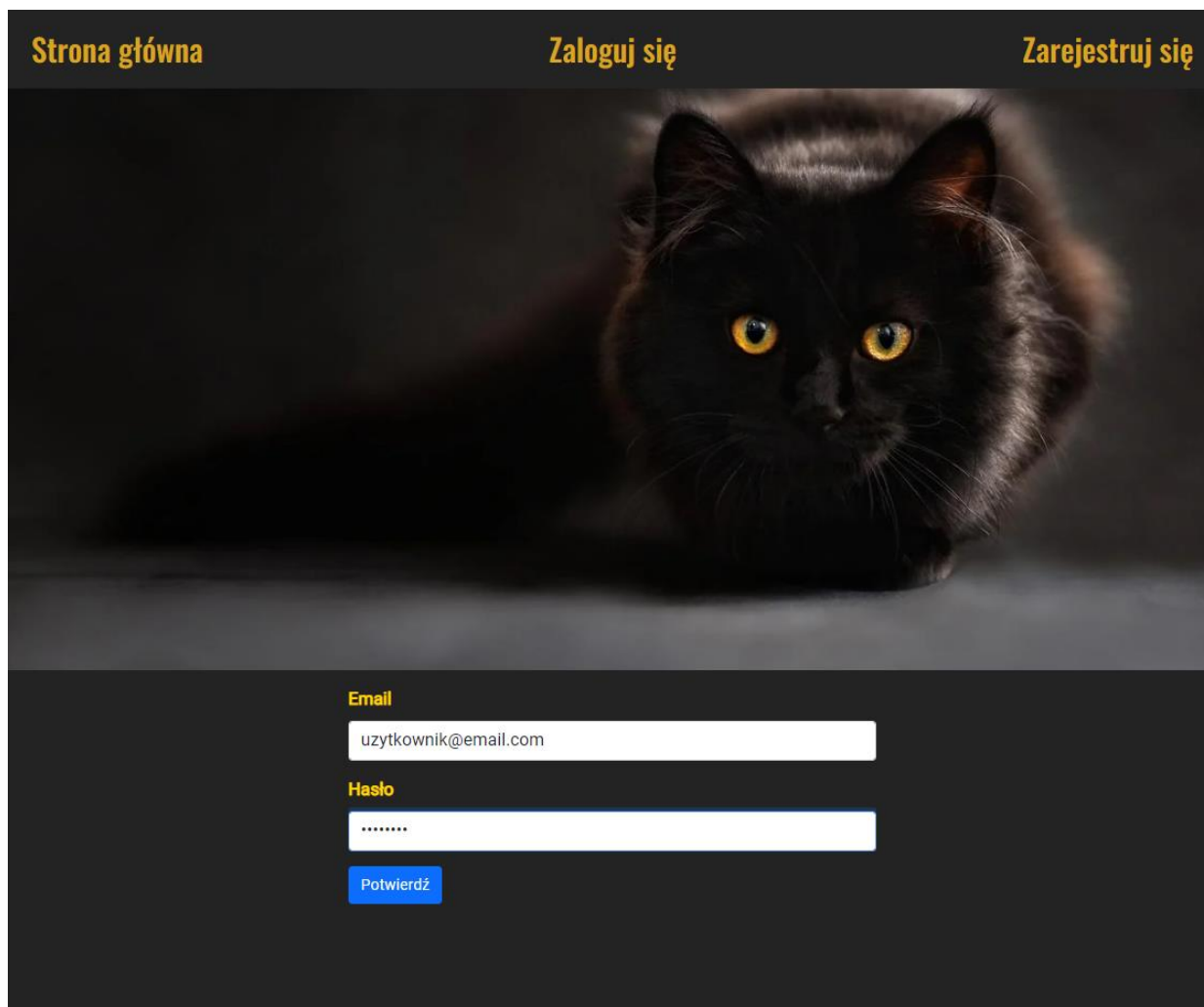


The screenshot displays the registration interface of a web application. At the top, a dark header contains three navigation links in yellow text: "Strona główna", "Zaloguj się", and "Zarejestruj się". The central part of the page features a large, dark image of a black cat with striking yellow eyes. Below this image, the registration form is presented on a dark background. It includes three input fields with yellow labels: "Nazwa" (with the placeholder "Użytkownik"), "Adres email" (with the placeholder "uzytkownik@email.com"), and "Hasło" (with the placeholder "....."). A blue "Potwierdź" button is positioned at the bottom of the form.

Rysunek 4 Ekran rejestracji

4.2. Logowanie

Jeśli użytkownik posiada poprawnie utworzone konto w aplikacji, to może się zalogować, w tym przypadku należy wypełnić dane pola podobnie jak w przypadku rejestracji i po wpisaniu poprawnych danych użytkownik zaloguje się do aplikacji, w innym przypadku program zwróci użytkownikowi błąd. Ekran logowania został przedstawiony na Rys. 5.



Strona główna Zaloguj się Zarejestruj się

Email
uzytkownik@email.com

Hasło

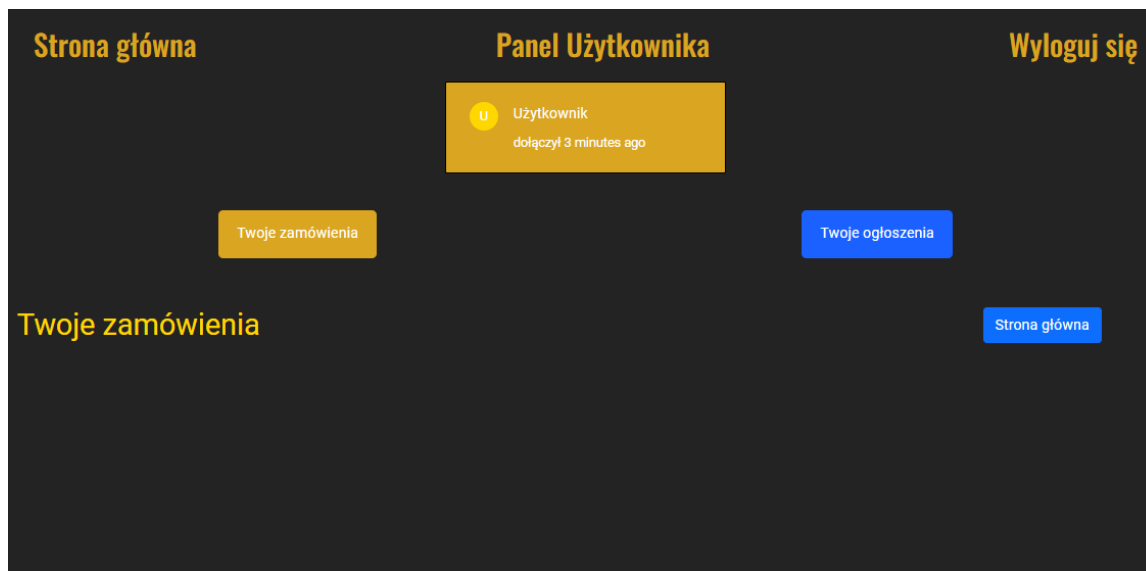
Potwierdź

Rysunek 5 Ekran logowania

4.3. Panel użytkownika

Po zalogowaniu się użytkownik widzi swój panel użytkownika, posiada dwie zakładki „Twoje zamówienia” i „Twoje ogłoszenia”. Domyślnie pokazuje się zakładka „Twoje zamówienia”, gdyż program zakłada, iż nie każdy użytkownik będzie zarówno sprzedawcą i

przekierowuje go od razu do poglądu poprzednich zamówień. Jeśli użytkownik jeszcze nic nie rezerwował, będzie to pole puste. Zostało to przedstawione na Rys. 6.

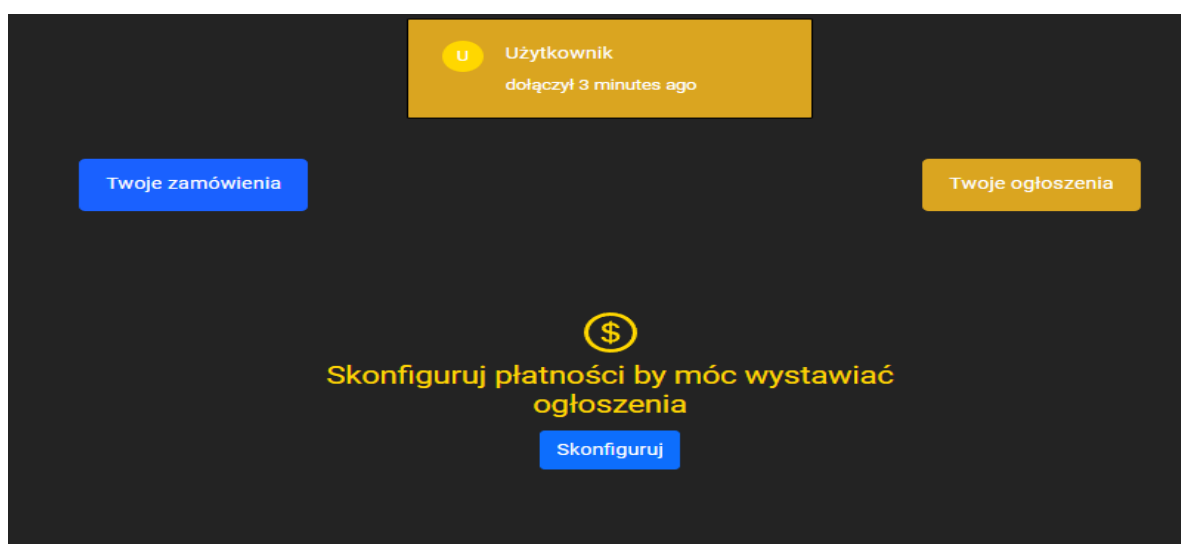


Rysunek 6 Panel użytkownika

4.4. Konfiguracja płatności w panelu użytkownika

Przechodząc do zakładki „Twoje ogłoszenia”, użytkownik, który nie jest sprzedawcą zobaczy komunikat, aby skonfigurować swoje konto do płatności w Stripe.

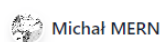
Po kliknięciu przycisku do konfiguracji użytkownik zostanie przekierowany do konfiguracji swojego konta w Stripe Rys. 7.



Rysunek 7 Konfiguracja płatności w panelu użytkownika

4.5. Konfiguracja płatności w Stripe

Jeśli użytkownik zdecyduje się korzystać z aplikacji w celach zarobkowych będzie musiał przejść przez weryfikację w Stripe. Podaje on wtedy swój adres, nazwę firmy, swoje konto bankowe, numer telefonu, adres e-mail. W celu zwiększenia bezpieczeństwa i ochrony konsumentów Stripe weryfikuje autentyczność danej osoby i poprosi o przekazanie numeru swojego dokumentu w formie zdjęcia. Jest to wygodne rozwiązanie, które pozwala bardzo szybko bez zbędnych dokumentów zweryfikować użytkownika. Z podobnych rozwiązań korzystają popularne platformy świadczące usługi bankowe takie jak Revolut czy ZEN. Procedurę przedstawiono na Rys. 8.



**Platforma Michał MERN
współpracuje ze Stripe, aby
zapewnić bezpieczeństwo
płatności i usług finansowych.**

Powered by **stripe** ⓘ

[Kontakt](#)

[Warunki](#)

[Prywatność](#)

Polski ⇅

Tryb testowy

Zweryfikuj dane osobowe

Stripe zbiera te informacje, aby zweryfikować Twoją tożsamość i zapewnić bezpieczeństwo Twojego konta.

Nazwa prawna

Imię

Nazwisko

Adres e-mail

ty@firma.pl

Data urodzenia

Dzień ⇅

Miesiąc ⇅

Rok ⇅

Adres domowy

Polska ⇅

Wiersz adresu 1

Wiersz adresu 2

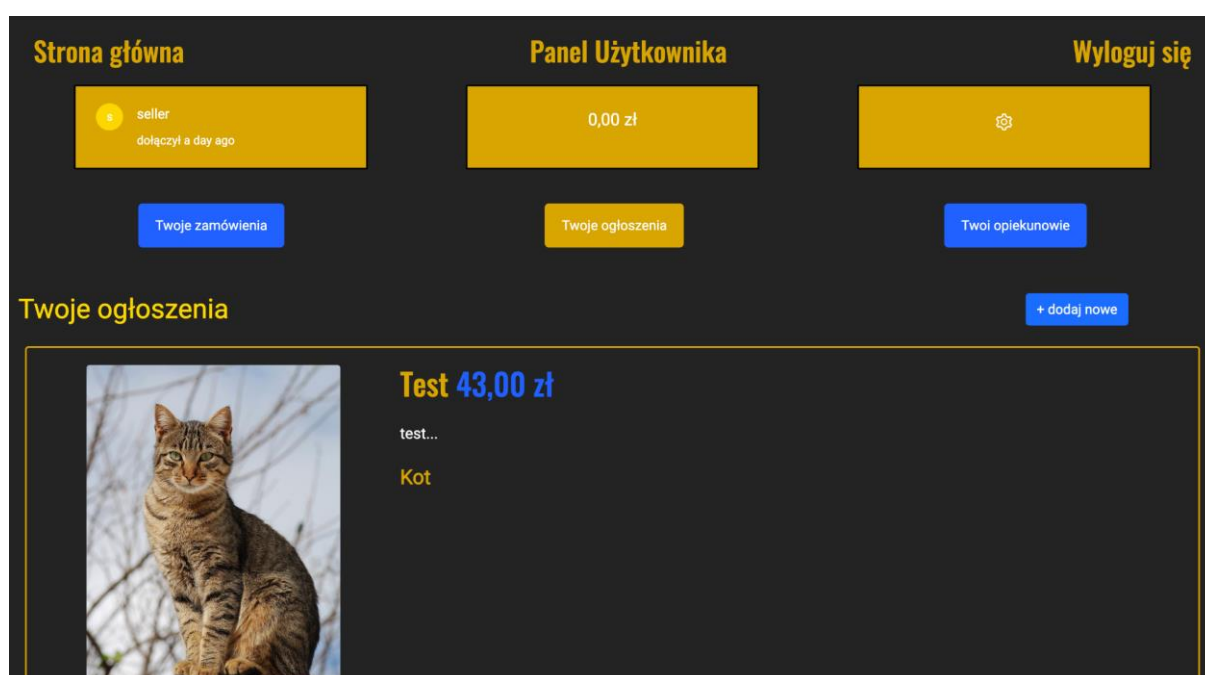
Kod pocztowy

Miasto

Rysunek 8 Konfiguracja płatności w STRIPE

4.6. Panel użytkownika po skonfigurowaniu do wystawiania ogłoszeń

Po poprawnej weryfikacji i konfiguracji konta poprzez Stripe użytkownik może zacząć dodawać własne ogłoszenia. Dodatkowo w panelu użytkownika pojawiają mu się dwa kolejne panele: jeden z saldem konta użytkownika, drugi z ikoną ustawień, gdzie po kliknięciu go użytkownik zostanie przekierowany do ustawień i statystyk swojego konta w Stripe. W tym momencie użytkownik może dodawać do bazy swoich opiekunów, którzy będą się zajmować podopiecznymi oraz zacząć wystawiać ogłoszenia. Dla użytkownika, który skonfigurował swoje konto do wystawiania ogłoszeń pojawia się kolejna zakładka mianowicie "Twoi opiekunowie", która pozwala zarządzać opiekunami Rys. 9.

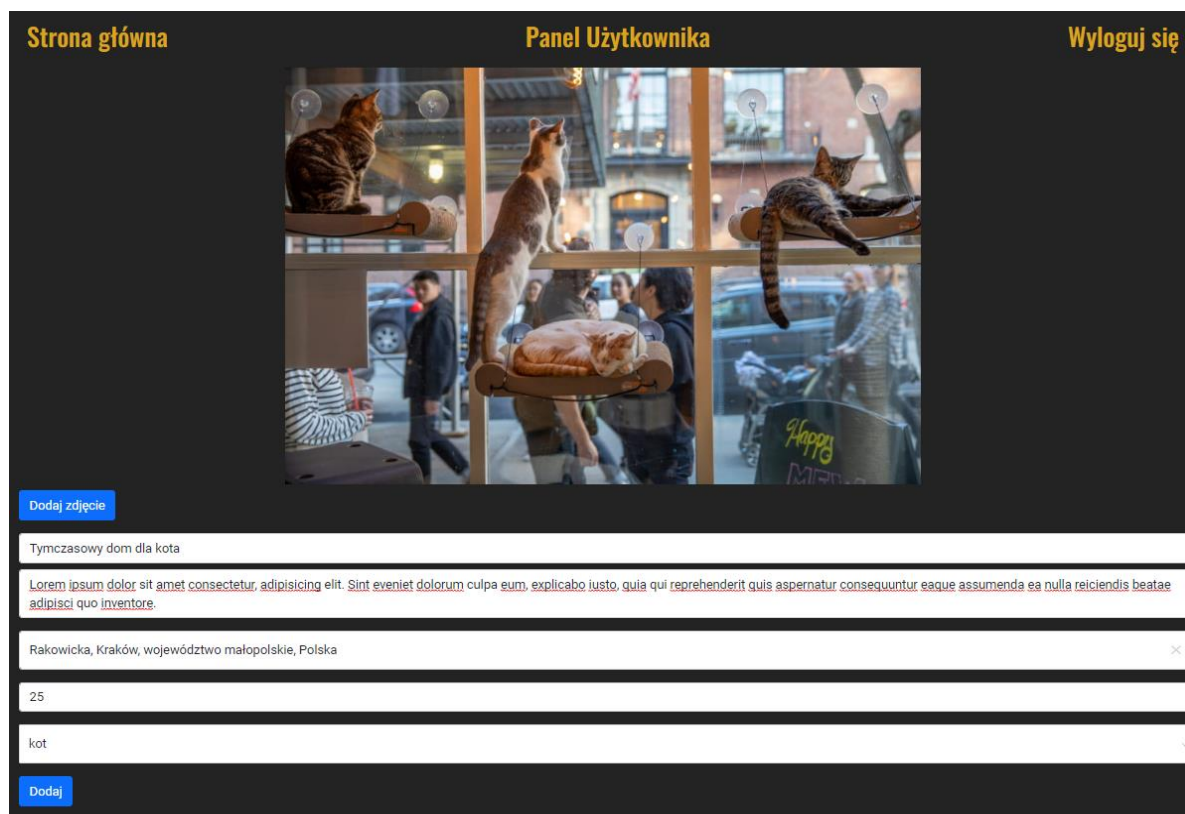


Rysunek 9 Panel użytkownika po skonfigurowaniu do wystawiania ogłoszeń

4.7. Dodawanie/Edycja ogłoszeń

Dodawanie ogłoszenia lub jego edycja to standardowa procedura, gdzie użytkownik wypełnia pola odpowiednimi danymi. Poprzez wybranie przycisku dodaj zdjęcie użytkownik ma możliwość dodania zdjęcia z lokalnego urządzenia. Aplikacja posiada również zaimplementowaną bibliotekę wyszukiwania Algolia [13], która umożliwia użytkownikowi

znalezienie odpowiedniej lokalizacji z mapy i podłączenie jej pod swoje ogłoszenie. Po wszystkim należy wszystko zatwierdzić przyciskiem na dole ekranu Rys. 10.



Strona główna Panel Użytkownika Wyloguj się

Dodaj zdjęcie

Tymczasowy dom dla kota

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Sint eveniet dolorum culpa eum, explicabo iusto, quia qui reprehenderit quis aspernatur consequuntur eaque assumenda ea nulla reiciendis beatae adipisci quo inventore.

Rakowicka, Kraków, województwo małopolskie, Polska

25

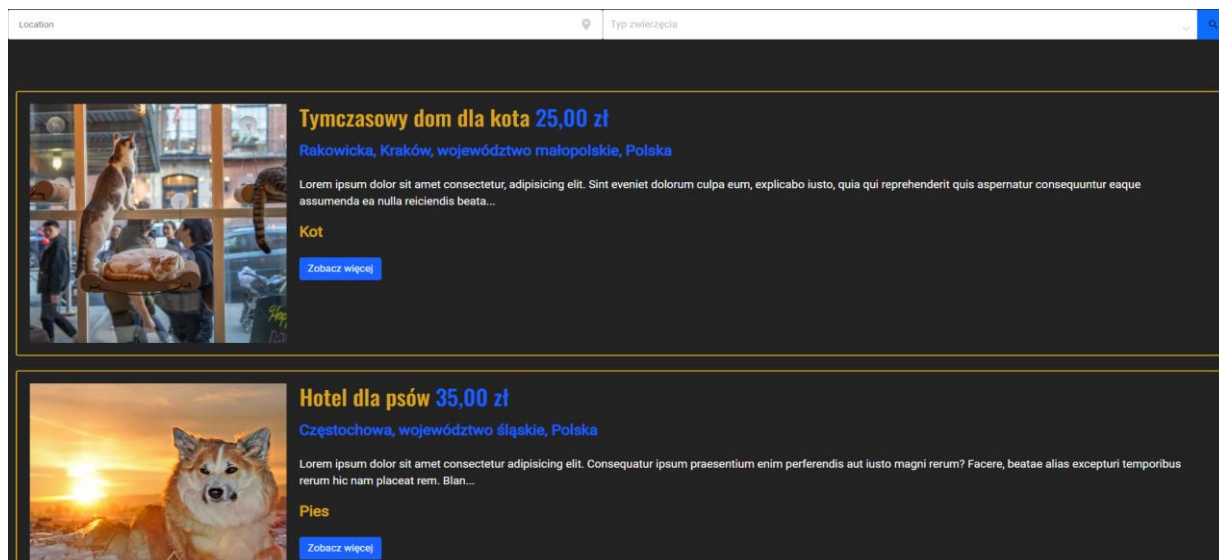
kot

Dodaj

Rysunek 10 Dodanie/edycja ogłoszenia

4.8. Strona główna

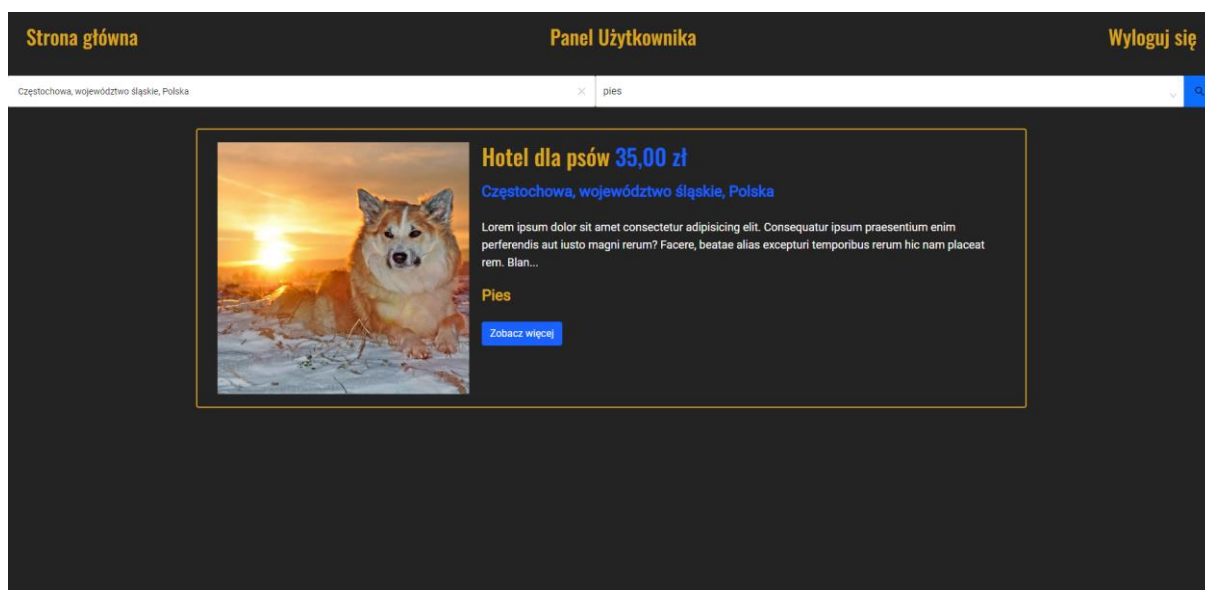
Po wejściu na stronę główną użytkownikowi pojawia się spis aktualnych ogłoszeń oraz na górze panel z filtrami, gdzie może przefiltrować wyniki pod względem interesujących go kategorii i miejsc Rys. 11.



Rysunek 11 Ogłoszenia na stronie głównej

4.9. Filtrowanie ogłoszeń

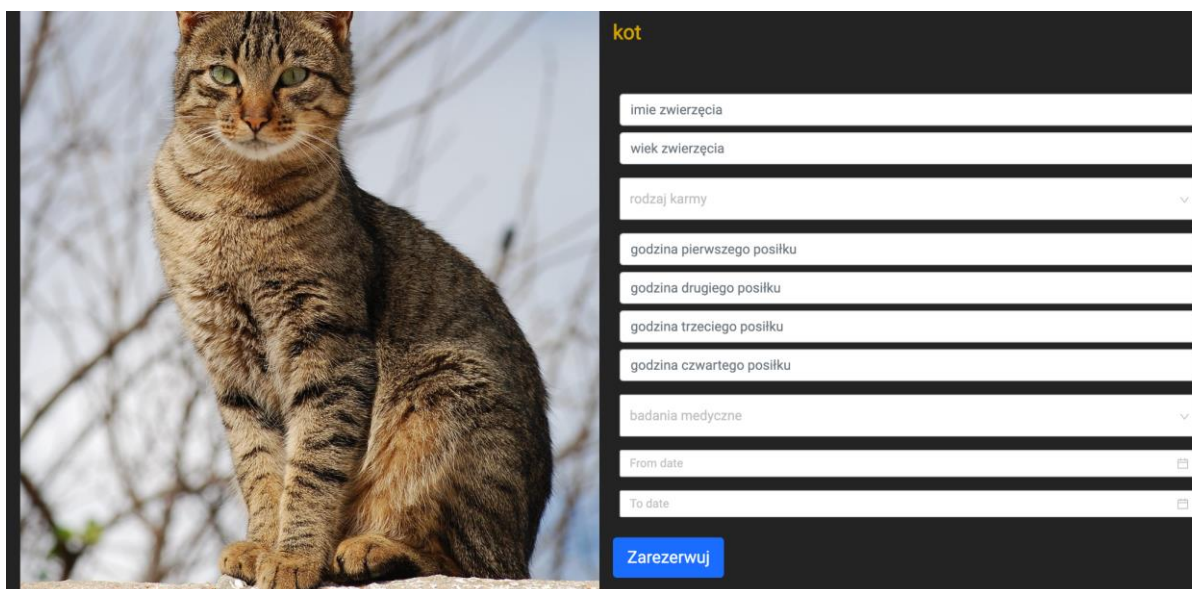
Użytkownik może przefiltrować dane ogłoszenie zarówno za pomocą obu dostępnych filtrów jak i używając tylko jednego z nich, aplikacja nadal przefiltruje wyniki poprawnie biorąc pod uwagę tylko podaną wartość Rys. 12.



Rysunek 12 Filtrowanie ogłoszeń

4.10. Dokonywanie rezerwacji

Gdy użytkownik zdecyduje się zarezerwować dane ogłoszenie, wystarczy, że kliknie w przycisk znajdujący się w nim „Zobacz więcej” jak i w tytuł lub zdjęcie ogłoszenia. Wtedy pojawią mu się pełne szczegóły ogłoszenia. Po wybraniu interesującego go zakresu czasowego po kliknięciu przycisku „zarezerwuj”, zostanie przekierowany do płatności Stripe. Jako że w zleceniu jest podana stawka za dobę, program automatycznie przeliczy należną płatność pod względem ilości dni. Użytkownik ma tutaj do wypełnienia kilka pól takich jak imię zwierzęcia, wiek, rodzaj karmy czy godziny wydawania posiłków. Może również zdecydować się na dodatkowe badania lekarskie. Zostało to przedstawione na Rys. 13.

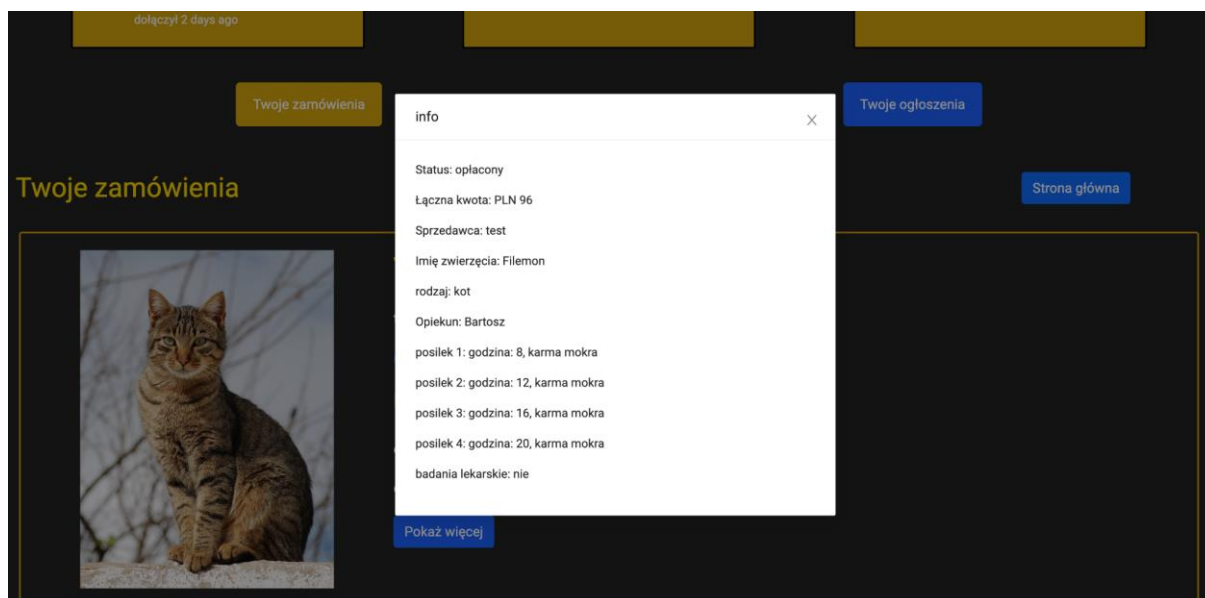


The image shows a reservation form for a cat. On the left is a photo of a brown tabby cat sitting on a rock. To the right is a dark-themed form with the title 'kot' in yellow. The form contains the following fields: 'Imię zwierzęcia', 'wiek zwierzęcia', 'rodzaj karmy' (with a dropdown arrow), 'godzina pierwszego posiłku', 'godzina drugiego posiłku', 'godzina trzeciego posiłku', 'godzina czwartego posiłku', 'badania medyczne' (with a dropdown arrow), 'From date' (with a calendar icon), and 'To date' (with a calendar icon). At the bottom of the form is a blue button labeled 'Zarezerwuj'.

Rysunek 13 Dokonywanie rezerwacji

4.11. Szczegóły rezerwacji

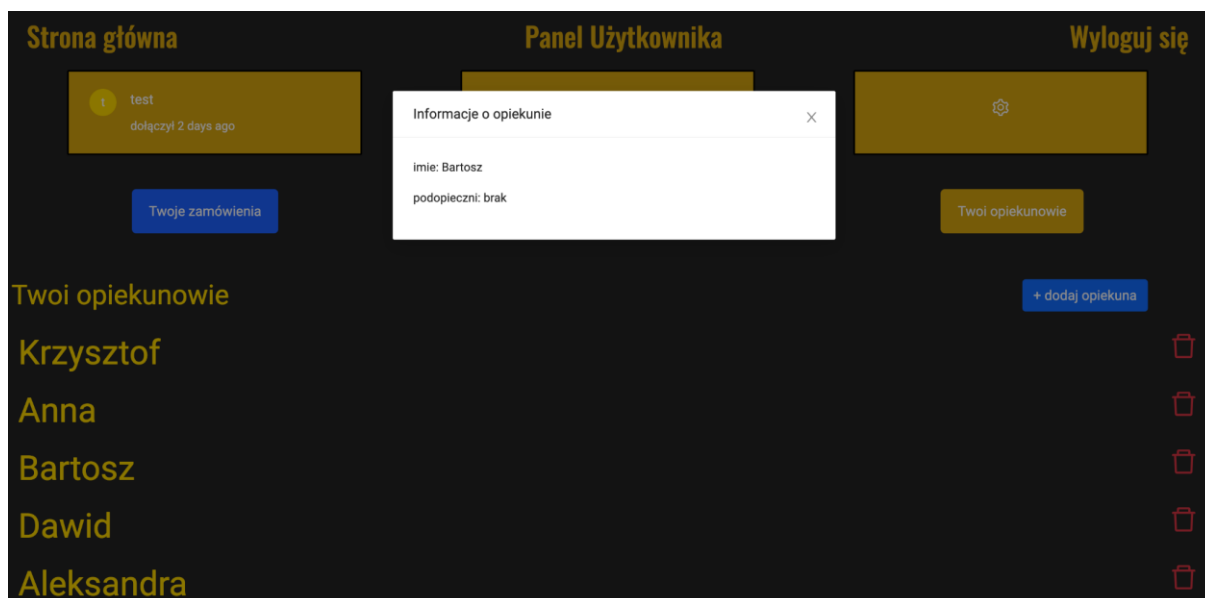
Po zarezerwowaniu danego ogłoszenia przez użytkownika pojawi się ono w zakładce „Twoje zamówienia”. Po wejściu w szczegóły ogłoszenia zobaczy on szczegółowe informacje o swojej rezerwacji Rys. 14.



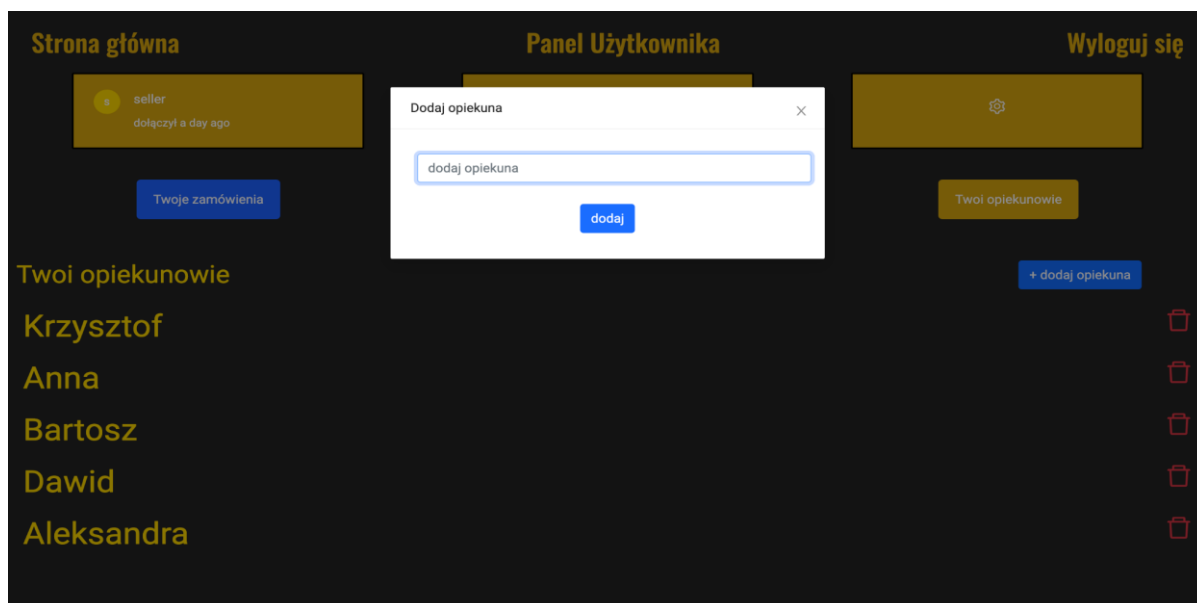
Rysunek 14 Sprawdzenie szczegółów rezerwacji przez użytkownika

4.12. Opiekunowie

Użytkownik, który skonfigurował swoje konto do wystawiania ogłoszeń ma również możliwość dodawania opiekunów do swojej bazy danych, a także zarządzania nimi bądź usuwania ich. W tym celu należy się udać do zakładki “Twoi opiekunowie”, gdzie pojawi się lista z opiekunami. Po naciśnięciu na imię danego opiekuna pojawi się okno ze szczegółowymi informacjami na temat podopiecznych danego opiekuna. Podopieczni zostaną przypisani automatycznie do opiekuna po rezerwacji dokonanej w danym obiekcie, przez klienta danego ogłoszenia, gdy użytkownik wystawiający dane ogłoszenie usunie danego opiekuna z bazy, system automatycznie przypisze zwierzę do innego opiekuna. W przypadku gdy użytkownik chce dodać nowego opiekuna do swojej bazy danych lub usunąć wystarczy nacisnąć odpowiedni przycisk po prawej stronie, w przypadku usunięcia nacisnąć przycisk po prawej stronie nazwy danego opiekuna Rys. 15. i Rys. 16.



Rysunek 15 Panel z opiekunami



Rysunek 16 dodanie nowego opiekuna do bazy danych

5. Podsumowanie

W ramach pracy zaprojektowano i zaimplementowano system informatyczny od samego początku, od badania rynku i istniejących rozwiązań poprzez sformułowanie wymagań dotyczących projektu, aż do tworzenia, implementacji i testowanie całego systemu.

Proces tworzenia projektu pokazał, że dziś praca programisty to nie tylko praca odtwórcza i wykonywanie zleconych zadań, lecz również uczestniczenie w procesie tworzenia projektu aplikacji od strony teoretycznej i sprecyzowania wymagań pod klienta. Projekt ten pozwolił również na dogłębnierze poznanie technologii tworzenia aplikacji webowych wykorzystywanych obecnie na rynku i analizy tych technologii nie tylko pod kątem samego stworzenia aplikacji, lecz potencjalnej rozbudowy aplikacji i rozbudowy całego projektu poprzez zatrudnienie w nim większej ilości osób.

Możliwe scenariusze rozwoju systemu:

- dodawanie komentarzy,
- dodanie możliwości oceny ogłoszeń jak i sprzedających,
- wprowadzenie możliwości pozyskania odznaki zaufanego sprzedawcy,
- rozbudowa kolejnych kategorii,
- dodanie nowych typów użytkowników,
- stworzenia aplikacji mobilnych na systemy operacyjne iOS lub Android,
- logowanie do aplikacji za pomocą portali społecznościowych takich jak Facebook, Google,
- logowanie do aplikacji za pomocą telefonu.

Założone wymagania implementacyjne i funkcjonalne zostały spełnione, a system został odpowiednio przetestowany zarówno pod względem działania na różnych urządzeniach, jak i na przeglądarkach internetowych. Aktualna wersja systemu jest stabilna i system jest gotowy do wdrożenia i użytkowania usprawniając ludziom posiadającym zwierzęta domowe planowanie podróży.

Stworzony system posiada jednak nadal wady:

- brak możliwości dodawania komentarzy,
- brak weryfikacji przez administrację,
- jedynie dwa typy kategorii zwierząt: pies i kot,

- mała ilość dodatkowych usług.

Projekt systemu informatycznego oraz jego realizacja były związane z rozwiązaniem problemu wobec zwierząt domowych w razie nieobecności właściciela z różnych przyczyn i znalezienia satysfakcjonującego rozwiązania spełniającego wymagania użytkownika. System zakładał wykorzystanie nowoczesnych technologii w celu zapewnienia długoletniego wsparcia oraz w stworzeniu maksymalnie wygodnego rozwiązania w użytkowaniu.

Bibliografia

- [1] S. Hoque, Full-Stack React Projects - Second Edition: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node.js, Packt Publishing, 2020.
- [2] MongoDB, Dokumentacja techniczna, [Online].
<https://www.mongodb.com/docs/>, 2022.
- [3] Express.js, Dokumentacja techniczna, [Online].
<https://devdocs.io/express>, 2022.
- [4] React, Dokumentacja techniczna, [Online].
<https://reactjs.org/docs/getting-started.html>, 2022.
- [5] Node.js, Dokumentacja techniczna, [Online].
<https://devdocs.io/node/>, 2022.
- [6] B. Frain., Responsive Web Design with HTML5 and CSS: Develop future-proof responsive websites using the latest HTML5 and CSS techniques, 3rd Edition, Packt Publishing; 3rd edition, April 30, 2020.
- [7] M. Haverbeke. Eloquent JavaScript 3rd edition, No Starch Press, 2018.
- [8] SASS, Dokumentacja techniczna, [Online].
<https://sass-lang.com/documentation/>, 2022.
- [9] Stripe, Dokumentacja techniczna, [Online].
<https://stripe.com/docs>, 2022.
- [10] G. Amuthan. Spring MVC Przewodnik dla początkujących, Wydawnictwo Helion, Gliwice 2015.
- [11] Font Awesome, dokumentacja techniczna, [Online].
<https://fontawesome.com/docs>, 2022.
- [12] Bootstrap Dokumentacja techniczna, [Online].
<https://getbootstrap.com/docs/4.0/getting-started/introduction/>, 2022.
- [13] Algolia, dokumentacja techniczna, [Online].
<https://www.algolia.com/doc/>, 2022.
- [14] Ant Design, dokumentacja techniczna, [Online].
<https://www.ant.design/docs/react/introduce>, 2022.

Spis tabel

Tab. 1 Przeglądanie ofert	20
Tab. 2 Filtrowanie wyników.....	20
Tab. 3 Przeglądanie szczegółów ogłoszenia	21
Tab. 4 Logowanie.....	21
Tab. 5 Rejestracja	22
Tab. 6 Przeglądanie własnego profilu.....	22
Tab. 7 Konfiguracja konta do płatności	23
Tab. 8 Sprawdzanie statystyk zarobków	23
Tab. 9 Zarezerwowanie ogłoszenia	23
Tab. 10 Dodanie ogłoszenia.....	24
Tab. 11 Edytowanie ogłoszenia	24
Tab. 12 Usunięcie ogłoszenia	25
Tab. 13 Przeglądanie poprzednich zamówień.....	25
Tab. 14 Szczegóły ogłoszenia.....	25
Tab. 15 Dodanie opiekuna.....	26
Tab. 16 Usunięcie opiekuna	26

Spis Ilustracji

Rysunek 1 Wzorzec Model View Controller	15
Rysunek 2 Diagram przypadków użycia.....	18
Rysunek 3 Schemat bazy danych.....	19
Rysunek 4 Ekran rejestracji	27
Rysunek 5 Ekran logowania	28
Rysunek 6 Panel użytkownika.....	29
Rysunek 7 Konfiguracja płatności w panelu użytkownika	29
Rysunek 8 Konfiguracja płatności w STRIPE	30
Rysunek 9 Panel użytkownika wystawiającego ogłoszenia	31
Rysunek 10 Dodanie/edycja ogłoszenia	32
Rysunek 11 Ogłoszenia na stronie głównej.....	33
Rysunek 12 Filtrowanie ogłoszeń	33
Rysunek 13 Dokonywanie rezerwacji.....	34
Rysunek 14 Sprawdzenie szczegółów zarezerwowanego ogłoszenia przez użytkownika	35
Rysunek 15 Panel z opiekunami.....	36
Rysunek 16 dodanie nowego opiekuna do bazy danych.....	36

Streszczenie

W obecnych czasach zwierzę w domu staje się bardzo często nieodzownym członkiem rodziny. Jednakże w przypadku wyjazdów, czy to na wakacje, służbowych czy gdzieś daleko do rodziny, pojawiają się problemy natury logistycznej jak rozwiązać problem z karmieniem, sprzątaniami czy innymi potrzebami dobrostanu zwierzęcia.

Zaprezentowany został projekt systemu informatycznego (aplikacji webowej) umożliwiającego użytkownikowi znalezienie odpowiedniego schroniska bądź hotelu dla zwierząt domowych.

Praca składa się z 5 rozdziałów. Rozdział 1 opisuje cel pracy wraz z prezentacją podobnych, istniejących na rynku rozwiązań, opisując wady i zalety każdego z nich. Wprowadza też teoretycznie w zagadnienia systemu informatycznego i aplikacji internetowej. W 2 rozdziale zostało przedstawione kryterium doboru danych technologii informatycznych zastosowanych w systemie oraz dobrane ostatecznie technologie. Także do każdej z wybranych technologii został dodany opis i uzasadnienie doboru wybranej technologii. W 3 rozdziale przedstawiono wymagania dotyczące budowy systemu. Struktura projektu w oparciu o wzorce projektowe i interfejs użytkownika. Został także przedstawiony diagram przypadków użycia i schemat bazy danych. Przedstawiono scenariusze przypadków użycia, a także problemy implementacyjne napotkane w projekcie. 4 Rozdział zawiera prezentację systemu w formie praktycznej. Opisy danego działania oraz interfejs graficzny. 5 rozdział zawiera podsumowanie całego projektu wraz z możliwymi scenariuszami rozbudowy całego systemu.

Abstract

Nowadays, a pet at home very often becomes an indispensable member of the family. However, in the case of trips, whether for vacation, business or somewhere far away to visit family, there are logistical problems of how to solve the problem of feeding, cleaning or other welfare needs of the animal.

The design of an information system (web application) that allows the user to find a suitable shelter or hotel for pets is presented.

The work consists of 5 chapters. Chapter 1 describes the purpose of the work with a presentation of similar existing solutions on the market, describing the advantages and disadvantages of each of them. It also gives a theoretical introduction to the issues of information system and web application. Chapter 2 presents the criterion for selecting the given information technologies used in the system and the technologies finally selected. Also, a description and justification for the selection of the chosen technology has been added to each of the selected technologies. In the 3rd chapter, the requirements for the construction of the system are presented. Design structure based on design patterns and user interface. A use case diagram and database diagram were also presented. Use case scenarios are presented, as well as implementation problems encountered in the project. 4 The chapter includes a presentation of the system in practical form. Descriptions of the given activity and the graphical interface. 5 Chapter contains a summary of the entire project with possible scenarios for expanding the entire system.