



VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
KOMPIUTERINIO IR DUOMENŲ MODELIAVIMO KATEDRA

Bakalauro baigiamasis darbas

Centralizuota gyvūnų globos namų sistema

Atliko:

Rūta Jovaišaitė

parašas

Vadovas:

dr. Joana Katina

Vilnius
2020

Turinys

Santrauka	3
Summary	4
Ivydas	5
1. Susijusių sistemų analizė	6
1.1. „Tautmilės globa“	6
1.2. „Vilniaus gyvūnų globos namai - Grinda“	6
1.3. „Vilniaus prieglauda - LESE“	6
1.4. „Northeast Animal Shelter“	7
1.5. „Toledo animal rescue“	7
1.6. Sistemų palyginimas	8
2. Teorinis sistemos modelis	9
2.1. Sistemos funkciniai reikalavimai	9
2.2. Sistemos naudojimo atvejai	11
2.3. Duomenų bazės modelis	12
2.4. Aplikacijos į gyvūno globą pildymas	14
3. Sistemos praktinis įgyvendinimas	15
3.1. Naudojamos technologijos	15
3.2. Eloquent ORM	15
3.3. Šablono struktūra	16
3.4. Sistemos struktūra	16
3.5. Naujo gyvūno įrašo kūrimas ir valdymas	18
3.6. Įrašų prenumeravimas	19
3.7. Aplikavimas į gyvūno globą	20
3.8. Duomenų filtravimas	20
3.9. Administratoriaus panelė	22
3.10. Saugumo sprendimai	23
Išvados ir rekomendacijos	24
Literatūros šaltiniai	25

Santrauka

Darbo tikslas - sukurti centralizuotą gyvūnų globos namų sistemą, kurioje vartotojai galėtų peržiūrėti globai pasiruošusių gyvūnų sąrašą, o gyvūno globos draugijos valdytų gyvūnų įrašus. Sistemai įgyvendinti buvo analizuotos jau egzistuojančios sistemos Lietuvos ir užsienio rinkose, sukurtas teorinis sistemos modelis ir išsikelti pagrindiniai uždaviniai. Darbo rezultatas - sukurta sistema, kurioje registruoti vartotojai gali matyti globai pasiruošusių gyvūnų sąrašą: jį filtruoti, įtraukti mėgiamus gyvūnus į favoritus, matyti gyvūno vietą žemėlapyje, užsiprenumeruoti įrašus, gauti pranešimus apie naujai įdėtą gyvūną į savo elektroninio pašto adresą bei pildyti anketą gyvūno globai. Autoriaus privilegijas turintys nariai gali valdyti įrašus ir tvirtinti aplikacijas į gyvūno globą, o administratoriai valdyti kategorijas, miestus bei autorius.

Summary

Centralised Animal Shelter System

The goal of this work is to create an animal shelter system that would ease the way of adoption both for the user and for the animal wellness organization. In the first section of this work, already existing similar systems were analysed and deductions were made about what functionality would be needed. In the second section, a theoretical model of the system was modeled and afterwards, a website was created. Results that were achieved creating an animal shelter system:

- Users can register, see all available animal posts, view detailed information about them and the place they are at. They can also filter the animals by various characteristics and apply for adoption. Furthermore, there is a subscription functionality, where users can subscribe to an animal category to get notifications whenever there's a new record in the database that matches that category.
- Authors can create and manage posts, approve or deny adoption applications.
- Administrators manage see the overall statistics of the system objects and can manage cities, categories, authors.

Of course, the system is not fully polished and it might benefit from some improvements:

- Sending an email notification to the author whenever there's a new application submitted.
- Having a 'rehomed animals' category, where all adopted animals are displayed.

Ivyadas

Pasaulyje vis tobulėjant internetinėms sistemoms pamirštos lieka ne pelno siekiančios organizacijos. Vienos iš jų - gyvūnų globa užsiimančios įmonės. Jų puslapiai kurti senomis technologijomis ir nėra patogūs šiuolaikiniam naudotojui. Kadangi kiekviena gyvūnų globos organizacija turi savo puslapį, naudotojams gali būti sunku naršyti po kiekvieną iš jų, kad rastų sau tinkamą augintinį. Nuo to kenčia visos pusės: gyvūnai, kurie lėčiau suranda naujus namus, globėjai, kurie priversti naudotis nepatogiomis sistemomis, ir žmonės, norintys globoti gyvūną. Jeigu egzistuočiau viena, patogi centralizuota sistema, šių kliūčių būtų galima išvengti. Šio darbo tikslas - sukurti sistemą, kuria galėtų naudotis administratoriai, gyvūnų globos draugijos bei paprasti naudotojai, siekiant pagerinti gyvūnų globavimo procesą. Norint pasiekti tikslą, buvo įgyvendinti šie uždaviniai:

- Išanalizuotos panašios, jau egzistuojančios sistemos
- Sukurtas teorinis sistemos modelis
- Pasirinktos tinkamos technologijos
- Teorinis modelis įgyvendintas praktiškai

Sistemos pagrindinis funkcionalumas - registruoti naudotojai gali patogiai matyti sistemoje esančius įrašus: juos filtruoti, peržiūrėti detalų gyvūno aprašymą, matyti jo vietą žemėlapyje. Taip pat palengvintas aplikavimo procesas - nuėjus į detalų gyvūno puslapį, tereikia paspausti vieną mygtuką, užpildyti reikiamą informaciją ir išsiųsti formą. Norint nepraleisti naujų įrašų, yra galimybė prenumeruoti gyvūnų kategoriją: atsiradus naujam įrašui, vartotojas automatiškai gaus pranešimą savo elektroninio pašto adresu. Sistemos autoriai, gyvūnų globos organizacijos, galės patogiai kurti naujus įrašus, juos valdyti, bei patvirtinti ar ištrinti aplikacijas į gyvūno globą. Sistemos administratoriai valdys kategorijas, miestus, autorius.

Šis rašto darbas pradedamas 1-uoju skyriumi, kuriame analizuojamos penkios sistemos: trys iš Lietuvos rinkos, dvi iš užsienio. Skyriaus pabaigoje atliktas šių sistemų funkcionalumo palyginimas. 2-ajame skyriuje kuriamas teorinis puslapio modelis: apibrėžiami funkciniai reikalavimai, sudaryta naudojimo atvejų diagrama, sumodeliuota duomenų bazė bei pateikiama veiklų diagrama. 3-ajame skyriuje aprašomos pasirinktos technologijos bei praktinis sistemos įgyvendinimas ir veikimas.

1. Susijusių sistemų analizė

Prieš pradėdant kurti sistemos teorinį modelį, reikėjo nuspręsti, koks funkcionalumas yra svarbus sistemai. Buvo išanalizuotos 5 sistemos, 3 iš jų kurtos Lietuvos rinkai, o likusios užsienio. Buvo žiūrima tik į gyvūnų atvaizdavimą ir patogumą aplikuoti į jų globą. Kiekvienai prieglauda specifinė informacija (kontaktų puslapis, finansinės ataskaitos, informacija apie dresūrą ir panašaus pobūdžio informacija) nebuvo analizuota. Po analizės buvo nuspręsta, kokios funkcijos yra svarbios ir įtrauktos į teorinį modelį.

1.1. „Tautmilės globa“

VŠĮ „Tautmilės globa“ yra savanoriška beglobių gyvūnų globos organizacija, įsteigta 2012 m. Pagrindinis gyvūnų globos puslapio funkcionalumas:

- Dovanojamo gyvūno tipo pasirinkimas: šuo, katė arba kita.
- Galimas tolimesnis gyvūno tipo filtravimas pagal šiuos parametrus: amžius, ūgio kategorija, vieta, veislė, kailio tipas, lytis, sterilizacija/kastracija, pagrindinė spalva.
- Pasirinkus gyvūną, rodomos jo nuotraukos, informacija pagal visus anksčiau pateiktus parametrus, kontaktiniai duomenys bei gyvūnų globos organizacijos vieta pavaizduota žemėlapyje.
- Dingusių gyvūnų filtravimas pagal šiuos parametrus: amžius, ūgio kategorija, vieta, veislė, kailio tipas, lytis, sterilizacija/kastracija, pagrindinė spalva.
- Rastų gyvūnų filtravimas pagal šiuos parametrus: amžius, ūgio kategorija, vieta, veislė, kailio tipas, lytis, sterilizacija/kastracija, pagrindinė spalva.

1.2. „Vilniaus gyvūnų globos namai - Grinda“

Vilniaus gyvūnų globos namai surenka benamius gyvūnus miesto viešosiose teritorijose, suteikia jiems laikiną prieglaudą ir pirmąją veterinarinę pagalbą. Pagrindinis jų puslapio funkcionalumas:

- Gyvūnų, ieškančių šeiminių, pasirinkimas pagal tris parametrus: šunys, katės, kiti gyvūnai.
- Pasirinkus tipą, vaizduojamos gyvūnų nuotraukos „grid“ sistemoje.
- Paspaudus ant gyvūnų nuotraukos, rodoma daugiau informacijos apie gyvūną.
- Padovanotų gyvūnų sąrašas: nuotraukos pavaizduojamos „grid“ sistemoje, paspaudus ant nuotraukos rodoma daugiau informacijos apie gyvūną.
- „Pagauk - sterilizuok - paleisk“ gyvūnų sąrašas: rodomos jų nuotraukos, paspaudus ant jų - daugiau informacijos apie gyvūną.
- „Grįžę namo“ sąrašas: rodomos jų nuotraukos, paspaudus ant jų - daugiau informacijos apie gyvūną.

1.3. „Vilniaus prieglauda - LESĖ“

Įkurta 2007 m., „Lesė“ užsiima gyvūnų globa ir sterilizacija. Nuo įsikūrimo rado namus 6000 gyvūnų ir finansavo sterilizaciją daugiau nei 2100 gyvūnų. Pagrindinis puslapio funkcionalumas:

- Galimybė pasirinkti miestą, pagal kurį būtų vykdoma tolimesnė gyvūnų paieška.

- Pasirinkus miestą, atvaizduojami visi tame mieste esantys dovanojami gyvūnai.
- Galimas gyvūnų filtravimas pagal jų rūšį bei savivaldybę.
- Paspaudus ant gyvūno nuotraukos, rodoma daugiau informacijos apie patį gyvūną: aprašymas, bruožai, gyvūno vardas, rūšis, skaičius, lytis, lytinis aktyvumas, amžius ir kt.
- Nario registracija: leidžiama palikti komentarus po gyvūno aprašymu, redaguoti savo profilio informaciją.
- Padovanotų gyvūnų sąrašas ir filtravimas pagal rūšį ir savivaldybę.

1.4. „Northeast Animal Shelter“

Išteigtas 1976 m., tai vieni didžiausių Naujosios Anglijos ne pelno siekiantys, nevykdantys eutanazijos, gyvūnų globos namai. Pagrindinis puslapio funkcionalumas:

- Gyvūnų filtravimas pagal rūšį: katės arba šunys.
- Pasirinkus rūšį, vaizduojamos tą rūšį atitinkančios gyvūnų nuotraukos ir gyvūno amžius, svoris, veislė ir lytis.
- Kadangi vaizduojami net ir namus suradę gyvūnai, tad galima pasirinkti, kad būtų rodomi tik laisvi gyvūnai.
- Paspaudus ant gyvūno nuotraukos, rodomas trumpas gyvūno aprašymas ir daugiau nuotraukų bei gyvūno identifikacinis numeris.
- Galima navigacija tarp gyvūnų su pirmyn ir atgal mygtukais.
- Gyvūno profilio spausdinimo funkcija.

1.5. „Toledo animal rescue“

Išteigti 1927 m. JAV, nevykdantys eutanazijos ir suteikiantys gyvūnams namus neribotam laikui, kol tik gyvūnas išlieka fiziškai ir psichiškai sveikas. Pagrindinis puslapio funkcionalumas:

- Paieška pagal gyvūno rūšį.
- Pasirinkus rūšį, vaizduojamos tą rūšį atitinkančios gyvūnų nuotraukos, lytis, amžius ir trumpas aprašymas.
- Galima pildyti anketą, norint pradėti globoti gyvūną.

1.6. Sistemų palyginimas

1 lentelė. Sistemų palyginimas

	„Tautmilės globa“	"Grinda"	"LESĖ"	"Northeast Animal Shelter"	"Toledo animal rescue"
Gyvūno rūšies pasi- rinkimas	+	+	+	+	+
Filtravimas pagal dau- giau nei 1 parametrą	+	-	+	-	-
Platesnė informa- ciją apie gyvūną pa- spaudus ant nuotraukos	+	+	+	+	-
Komentarų funkcija	-	-	+	-	-
Padovanotų gyvūnų vaizdavi- mas	+	+	+	-	-
Aplikacijos į gyvū- no globą pildymas	-	-	-	-	+
Gyvūno vietos atvaiz- davimas žemėlapyje	+	-	-	-	-

Pagal atliktą gyvūnų globos puslapių palyginimą matoma, kad vienintelis bendras dalykas - gyvūno rūšies pasirinkimas (1 lentelė). „Tautmilės globa“, „Grinda“, „LESĖ“ bei „Northeast Animal Shelter“ taip pat turi detalesnį gyvūno peržiūros puslapį. Visos trys nagrinėtos lietuviškos svetainės turi atskirą skiltį padovanotiems gyvūnams. Filtravimą pagal daugiau nei vieną parametą turi tik „Tautmilės globa“ ir „LESĖ“. Palengvinti naudotojų aplikavimą į gyvūno globą nusprendė tik „Toledo animal rescue“: jie turi specialią formą pildyti prašymui, norint globoti gyvūną. Gyvūno vietos atvaizdavimą žemėlapyje turi tik „Tautmilės globa“.

2. Teorinis sistemos modelis

Prieš pradėdant kurti gyvūnų globos namų valdymo sistemą buvo nutarta, kokius funkcinius reikalavimus ji turės. Reikiamo funkcionalumo sprendimui didelę įtaką turėjo panašių, jau sukurtų sistemų analizė. Kadangi sistema naudotųsi daugelio gyvūnų globos namų savininkai, tad ji turi būti pritaikyta ir turėti patogią autoriaus panelę. Teoriniame modelyje yra suprojektuotas sistemos duomenų bazės, naudojimo atvejų ir veiklos diagramos.

2.1. Sistemos funkciniai reikalavimai

Kadangi detali reikalavimų analizė yra svarbus žingsnis prieš kuriant sistemas [8] tad, buvo sudarytas funkcijų planas, kuriame aiškiai nurodomi kiekvieno funkcionalumo reikalavimai bei naudotojų tipai.

Neprisijungę naudotojai turi galėti atlikti šiuos veiksmus:

- Paprasto vartotojo registracija ir prisijungimas. Registracijos anketą sudaro vardo, elektroninio pašto bei slaptažodžio laukai. Prisijungti prie sistemos naudojamas elektroninis paštas bei slaptažodis.
- Gyvūnų sąrašo peržiūra. Rodoma pagrindinė gyvūno nuotrauka, įrašo paskelbimo laikas, autoriaus vardas, „patinka“ paspaudimų skaičius, kategorija, miestas ir trumpas aprašymas.
- Detali gyvūno peržiūra. Rodomos viso gyvūno nuotraukos, įrašo paskelbimo laikas, autoriaus vardas, „patinka“ paspaudimų skaičius, kategorija, miestas, pilnas gyvūno aprašymas bei gyvūnų globos namų vieta žemėlapyje. Neprisijungęs vartotojas negali aplikuoti į gyvūno globą.

Prisijungę naudotojai turi galėti atlikti šiuos veiksmus:

- Gyvūnų sąrašo peržiūra. Rodoma pagrindinė gyvūno nuotrauka, įrašo paskelbimo laikas, autoriaus vardas, „patinka“ paspaudimų skaičius, kategorija, miestas ir trumpas aprašymas.
- Detali gyvūno peržiūra. Rodomos viso gyvūno nuotraukos, įrašo paskelbimo laikas, autoriaus vardas, „patinka“ paspaudimų skaičius, kategorija, miestas, pilnas gyvūno aprašymas bei gyvūnų globos namų vieta žemėlapyje.
- Aplikacijos anketa į gyvūno globą. Prisijungęs vartotojas gali pildyti prašymą globoti pasirinktą gyvūną. Jį sudaro vardo, kontaktinio numerio, dabartinio auginamų gyvūnų skaičiaus, pajamų ir priežasties, kodėl norima globoti gyvūną laukai.
- Vartotojo profilio redagavimas. Kiekvienas vartotojas gali pakeisti savo profilio nuotrauką bei vardą.
- Įrašų filtravimas. Įrašai gali būti filtruojami pagal gyvūno tipą: katės, šunys ir kita, pagal miestą: Vilnius, Kaunas, kita bei pagal įrašus, kuriuos vartotojas įtraukė į savo favoritų skiltį. Filtravimo lange taip pat yra mygtukas, leidžiantis panaikinti visą filtravimo pasirinkimą.
- Gyvūno įtraukimas į favoritus. Užvedus pelę ant gyvūno nuotraukos, galima spustelėti širdelę, kuri įtrauks gyvūną į vartotojo favoritų sąrašą. Dar kartą paspaudus širdelę ant to pačio gyvūno, jis bus pašalintas iš favoritų sąrašo.
- Gyvūnų kategorijos pranešimų prenumeracija. Užvedus pelę ant gyvūno ir paspaudus varpelio ikoną, vartotojas gali užsiprenumeruoti to gyvūno kategorijos automatinius pranešimus į savo elektroninio pašto adresą: atsiradus naujam gyvūno įrašui kategorijoje, vartotojas gaus pranešimą su nuoroda į detalų gyvūno aprašymo puslapį. Paspaudus ant varpelio antrą kartą, vartotojas gali atsisakyti tos kategorijos pranešimų.

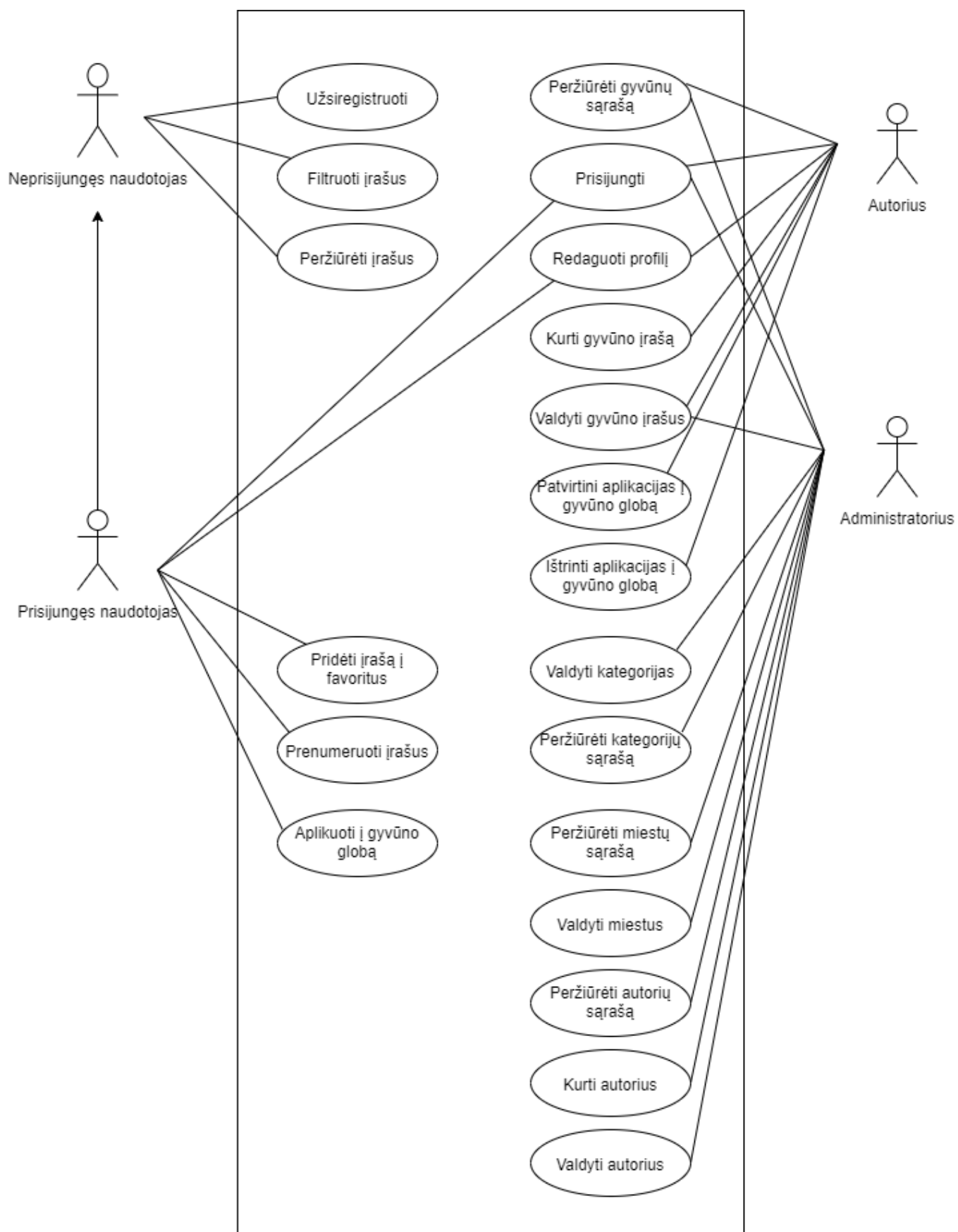
Autoriaus privilegijas turintys naudotojai turi galėti atlikti šiuos veiksmus:

- Autoriaus panelės peržiūra. Rodomas autoriaus sukurtų įrašų skaičius.
- Profilio redagavimas. Nuotraukos, vardo, mobilaus numerio redagavimas. (patobulint šita, mobilų numerį pridėti prie posto info etc)
- Gyvūnų sąrašo peržiūra. Rodomi visi autoriaus sukurti gyvūnai.
- Naujo gyvūno pridėjimas. Kuriant įrašą naujam gyvūnui, reikia pasirinkti kategoriją, miestą, adresą, parašytą aprašymą bei prisegti nuotraukas.
- Gyvūno įrašo redagavimas. Galima redaguoti prieš tai pateiktą informaciją: gyvūno kategoriją, miestą, adresą, aprašymą bei nuotraukas.
- Gyvūno įrašo pašalinimas. Galimybė pašalinti nebeaktualius įrašus iš sistemos.
- Detali gyvūno peržiūra. Rodoma pilna informacija apie gyvūną: kategorija, miestas, aprašymas, aplikantų į gyvūno globą sąrašas, su visais anketoje esančiais punktais.
- Aplikacijos į gyvūną patvirtinimas. Autorius gali patvirtinti pretendento į gyvūno globą anketą, ko pasekoje vartotojui bus išsiųstas pranešimas į jo elektroninį paštą, su platesne informacija apie tolimesnius veiksmus.
- Aplikacijos į gyvūną ištrynimasis. Autorius gali ištrinti neaktualią aplikaciją iš sistemos.

Administratoriaus privilegijas turintys naudotojai turi galėti atlikti šiuos veiksmus:

- Administratoriaus panelės peržiūra. Rodomi įrašų, kategorijų, autorių ir vartotojų skaičiai.
- Kategorijų sąrašo peržiūra. Rodomos jau sukurtos kategorijos.
- Naujų kategorijų kūrimas. Norint sukurti naują kategoriją, būtina įvesti jos pavadinimą.
- Kategorijų redagavimas. Galima redaguoti jau sukurtas kategorijas.
- Kategorijų ištrynimasis. Pašalinamos nereikalingos kategorijos.
- Miestų sąrašo peržiūra. Rodomi jau sukurti miestai.
- Naujų miestų kūrimas. Pridedami nauji miestai, būtinas miesto pavadinimo laukas.
- Miestų redagavimas. Būtina įvesti naują miesto pavadinimą.
- Miestų trynimasis. Galima pašalinti nereikalingus miestus.
- Autorių sąrašo peržiūra. Rodomas jau egzistuojančių autorių sąrašas.
- Autorių kūrimas. Galimybė sukurti naują autoriaus teises turinti vartotoją.
- Autorių redagavimas. Galima redaguoti autorių vardą, elektroninio pašto adresą.
- Autorių ištrynimasis. Galima trinti egzistuojančius autorius. Ištrynus autorių, ištrinami ir jo visi įrašai.
- Įrašų peržiūra. Rodomi visų autorių sukurti įrašai.
- Įrašų redagavimas. Galima redaguoti visų autorių sukurtus įrašus.
- Įrašų ištrynimasis. Galimybė ištrinti visų autorių sukurtus įrašus.

2.2. Sistemos naudojimo atvejai



1 pav. Sistemos panaudos atvejų diagrama

2.3. Duomenų bazės modelis

```
erDiagram
    users ||--o{ user_subscriptions : "has"
    users ||--o{ likes : "has"
    users ||--o{ animal_requests : "has"
    admins ||--o{ animal_requests : "has"
    categories ||--o{ posts : "has"
    categories ||--o{ likes : "has"
    categories ||--o{ animal_requests : "has"
    posts ||--o{ images : "has"
    posts ||--o{ cities : "has"
    posts ||--o{ authors : "has"
    user_subscriptions ||--o{ posts : "has"
    likes ||--o{ posts : "has"
    animal_requests ||--o{ posts : "has"
    animal_requests ||--o{ cities : "has"
    animal_requests ||--o{ authors : "has"

    users {
        string id PK
        string name
        string email
        string password
        string avatar
        string created_at
        string updated_at
    }
    admins {
        string id PK
        string name
        string email
        string password
        string created_at
        string updated_at
    }
    categories {
        string id PK
        string name
        string created_at
        string updated_at
    }
    posts {
        string id PK
        string author_id FK
        string category_id FK
        string city_id FK
        string description
        string address
        string latitude
        string longitude
        string created_at
        string updated_at
    }
    images {
        string id PK
        string post_id FK
        string path
    }
    cities {
        string id PK
        string name
        string created_at
        string updated_at
    }
    authors {
        string id PK
        string name
        string email
        string password
        string mobile
        string avatar
        string created_at
        string updated_at
    }
    user_subscriptions {
        string user_id FK
        string category_id FK
    }
    likes {
        string post_id FK
        string user_id FK
    }
    animal_requests {
        string post_id FK
        string user_id FK
        string name
        string income
        string pets
        string mobile
        string reason
        string status
        string created_at
        string updated_at
    }
```

- Users lentelėje yra saugomi naudotojų duomenys. Registruojantis įvedamas vardas, elektroninio pašto adresas ir slaptažodis, o vėliau, prisijungus, galima redaguoti savo profilio nuotrauką.

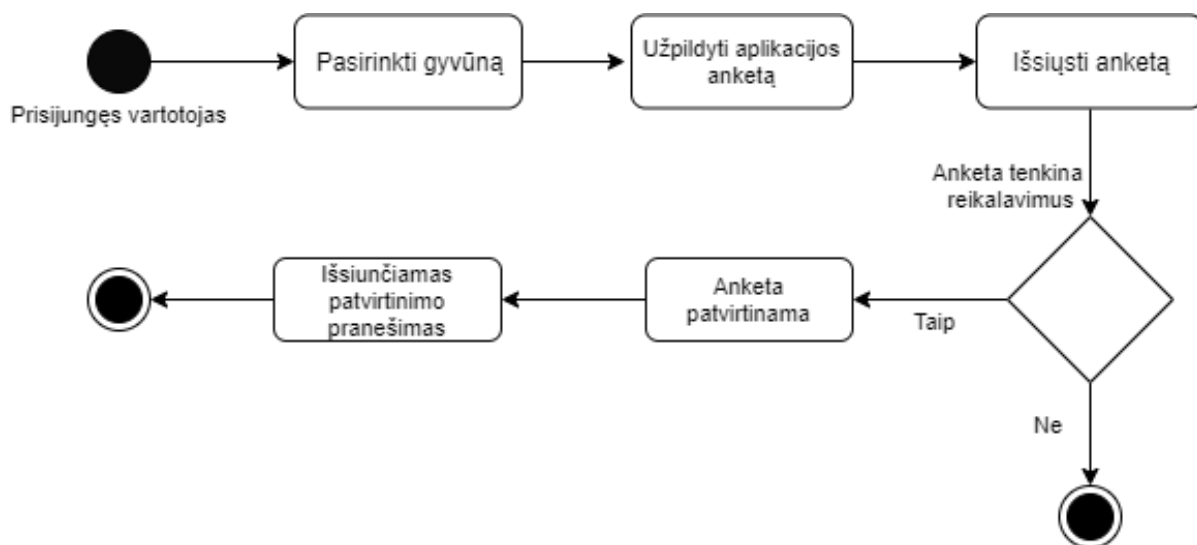
- Admins lentelėje saugoma informacija apie sistemos administratorius. Saugomas jų vardas, elektroninio pašto adresas bei slaptažodis.
- Authors lentelėje yra saugoma informacija apie įrašų kūrimo privilegiją turinčius vartotojus. Saugomas vardas, elektroninio pašto adresas, slaptažodis, mobilusis numeris ir profilio nuotrauka.
- Posts lentelėje saugomi duomenys apie įrašus. Duomenys į šią lentelę įrašomi autoriui užpildžius naujo gyvūno skelbimo sukūrimo anketą. Lentelė turi tris išorinius raktus: autoriaus identifikatorių, gaunamą iš authors lentelės, kategorijos identifikatorių, gaunamą iš kategorijų lentelės, ir miesto identifikatorių, gaunamą iš cities lentelės. Šie raktai leidžia įrašą priskirti įrašo autoriui, pasirinktai kategorijai ir miestui. Taip pat posts lentelė turi aprašymo, adreso, platumos ir ilgumos laukus, kurie naudojami atvaizduoti gyvūno vietą žemėlapyje.
- Categories lentelėje saugomas gyvūno kategorijos pavadinimas. Duomenis į šią lentelę gali įrašyti tik sistemos administratorius, naudojantis su administracine panele.
- Cities lentelėje saugomas miesto pavadinimas, kuriame saugomas gyvūnas. Duomenis į šią lentelę gali įrašyti tik administratorius tipo narys.
- Images lentelėje saugomi prie įrašo pridėti paveikslėliai. Ji turi vieną išorinį raktą: įrašo identifikatorių, gaunamą iš posts lentelės. Taip pat lentelėje yra kelio iki paveikslėlio laukas.
- Likes lentelėje yra saugomi duomenys apie naudotojo pamėgtus įrašus. Lentelė neturi pirminio rakto. Ji susideda iš dviejų išorinių raktų: įrašo identifikatoriaus, gaunamo iš įrašų lentelės ir naudotojo identifikatoriaus, gaunamo iš naudotojų lentelės.
- Naudotojo prenumeratos lentelėje saugomi duomenys apie tai, kokias kategorijas vartotojas yra užsiprenumeravęs, kad gautų informaciją apie naujus įrašus. Lentelę sudaro du išoriniai raktai: naudotojo identifikatorius, gaunamas iš naudotojų lentelės ir kategorijos identifikatorius, gaunamas iš kategorijos lentelės.
- Gyvūnų užklausų lentelėje saugomi duomenys apie naudotojų prašymus globoti gyvūną. Duomenys į lentelę yra įrašomi, kai naudotojas užpildo prašymą globoti gyvūną ir jį išsiunčia. Lentelę sudaro du išoriniai raktai: įrašo identifikatorius, gaunamas iš įrašų lentelės ir naudotojo identifikatorius, gaunamas iš naudotojų lentelės. Taip gaunama informacija apie tai, kuris vartotojas nori kurio gyvūno. Taip pat lentelę sudaro tokie laukai, kaip vardas, uždarbis, gyvūnų skaičius, mobilaus telefono numeris, priežastis, dėl kurios norima globoti gyvūną ir statusas, kuris rodo, ar prašymas yra patenkintas, ar ne.

Laravel karkasas palaiko duomenų bazių migracijas. Migracijos yra versijavimo sistema duomenų bazei, leidžianti komandos nariams lengvai modifikuoti ir dalintis aplikacijos duomenų schema.

Papildomos Laravel karkaso sukurtos lentelės:

- Migrations. Saugoma informacija apie atliktas migracijas. Lentelėje yra migracijos pavadinimo ir partijos laukai.

2.4. Aplikacijos į gyvūno globą pildymas



3 pav. Aplikavimo į gyvūno globą veiklos diagrama

3 pav. vaizduojamas aplikacijos į gyvūną anketos pildymo funkcionalumas. Registruotas naudotojas gali pasirinkti gyvūną, užpildyti reikiamą, anketoje prašomą informaciją ir išsiųsti anketą. Autoriai, savo autoriaus panelėje gali matyti visas su jų sukurtais gyvūnais susijusias anketas. Jeigu anketa tenkina autorių, ji gali būti patvirtinama ir į vartotojo elektroninio pašto adresą bus išsiunčiamas pranešimas, kad aplikacija yra sėkminga, su tolimesniais duomenimis. Jei anketa netenkina autoriaus išsikeltų reikalavimų, ji gali būti ištrinta.

3. Sistemos praktinis įgyvendinimas

3.1. Naudojamos technologijos

Prieš pradėdant kurti gyvūnų globos sistemą, buvo pasirinktos technologijos, kurios padėtų sukurti internetinį tinklalapį. Pagrindinis kriterijus renkantis technologijas - didelis ir aiškus dokumentacijos kiekis internete, kuris leistų be didelių sunkumų susipažinti su įrankių funkcionalumu ir padėtų, jei iškiltų sunkumų. Pasirinktos šios technologijos:

- PHP programavimo kalba;
- Laravel karkasas;
- MySQL duomenų bazės valdymo sistema;
- jQuery biblioteka;
- Vue.js karkasas;
- Bootstrap karkasas.

Pasirinkus PHP programavimo kalbą, reikėjo nuspręsti dėl karkaso. Visgi buvo pasirinktas Laravel karkasas, kadangi jis yra vienas iš populiariausių PHP programavimo kalbos karkasų [1]. Jame jau yra įdiegtas autentifikacijos funkcionalumas, yra komandinės eilutės interfeiso įrankis Artisan. Be abejo, Laravel turi ir puikią dokumentaciją. Karkasas taip pat turi objekto vaidmenų modeliavimą, kuris vadinasi Eloquent ORM. Jis suteikia galimybę nerašyti ilgų SQL užklausų, o vietoj to naudotis duomenų baze sąveikaujant su modeliais [6].

Duomenų valdymui buvo pasirinkta MySQL duomenų bazės valdymo sistema. Iš duomenų bazės įrašai gaunami ir siunčiami naudojantis aukščiau minėtomis Eloquent ORM užklausomis.

jQuery yra JavaScript biblioteka. Pasirinkta, nes turi metodus, kuriuos galima iškviesti su viena eilute kodo, vietoj to, kad reiktų rašyti daug eilučių JavaScript kodo.

Vue.js yra Javascript programavimo kalbos karkasas. Jis buvo pasirinktas dėl savo detalios dokumentacijos, mažo dydžio ir greičio. Galiausiai buvo pasirinktas Bootstrap karkasas. Tai yra HTML, CSS ir Javascript karkasas, kuris palaiko daugelį naršyklių, turi savaime prisitaikantį dizainą ir stilius, yra lengvai pritaikomas ir turi didžiulę dokumentacijos bazę.

3.2. Eloquent ORM

Eloquent ORM - tai architektūrinis šablonas, kai modelis, sukurtas Model-View-Controller (MVC) struktūroje atitinka lentelę duomenų bazėje. Modeliai leidžia vykdyti duomenų užklausas lentelėse, taip pat ir įterpti naujus įrašus į lenteles. Panaudoti "Eloquent" nėra sunku: iš pradžių reikia sukurti modelį. Tada „gyvatės atvejo“ (angl. „snake case“) būdu, daugiskaitinis modelio klasės pavadinimas bus naudojamas kaip modelio lentelė, nebent naudotojas pats nurodo kitaip. [3]

Autoriaus modelį susiejame su įrašų modeliu one to many tipo ryšiu, nes vienas autorius gali turėti daug įrašų.

1 išėities kodas. Autoriaus modelis.

```
1 public function posts()  
2 {  
3     return $this->hasMany(Post::class);  
4 }
```

Priskyrus tokį ryšį, galima lengvai gauti visus autoriaus įrašus vienos eilutės pagalba. Kadangi naudojamas Eloquent, nereikia vykdyti papildomo tikrinimo duomenų bazėje. Pavyzdyje parodyta, kaip randami visi autoriaus įrašai, kai autoriaus identifikatorius yra 1. Įrašų lentelėje surandami įrašai, kuriuose svetimo rakto `author_id` reikšmė yra 1.

2 išeities kodas. Įrašų modelis.

```
1 $posts = Author::find(1)->posts;
```

3.3. Šablono struktūra

Puslapio šablono struktūrai buvo naudojamas Blade šablono variklis, kuris jau yra integruotas Laravel karkase. Du pagrindiniai Blade pranašumai yra šablonų paveldėjimas ir sekcijos. Kadangi daugelis internetinių tinklalapių skirtinguose savo puslapiuose naudoja tokį patį ar panašų dizainą, tad norint sumažinti kodo pasikartojimo riziką, naudojamas bazinis šablono failas, kurį paskui paveldi vaikiniai puslapiai [2].

3 išeities kodas. Bazinio šablono struktūra.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <html>
4   <head>
5     <title>App Name – @yield('title')</title>
6   </head>
7   <body>
8     @section('sidebar')
9     This is the master sidebar.
10  @show
11  <div class="container">
12    @yield('content')
13  </div>
14 </body>
15 </html>
```

Šis bazinis šablonas vėliau yra praplėčiamas vaikiniuose šablonuose, naudojant komandą pavadinimu `@extends()`. Su `@section()` komanda, į atitinkamas šablono vietas yra įterpiamas sekcijose esantis kodas.

4 išeities kodas. Šablonas.

```
1 @extends('layouts.master')
2 @section('content')
3   turinio kodas
4 @endsection
```

Sistemoje, be bazinio šablono dar yra administratoriaus, autoriaus ir autorizacijos šablonai.

3.4. Sistemos struktūra

Kadangi sistema kurta naudojantis Laravel karkasu, tad buvo laikomasi šio karkaso kūrimo gerųjų praktikų. Jis naudoja MVC (model-view-controller) architektūrą, kuri sistemą suskaido į tris

dalis:

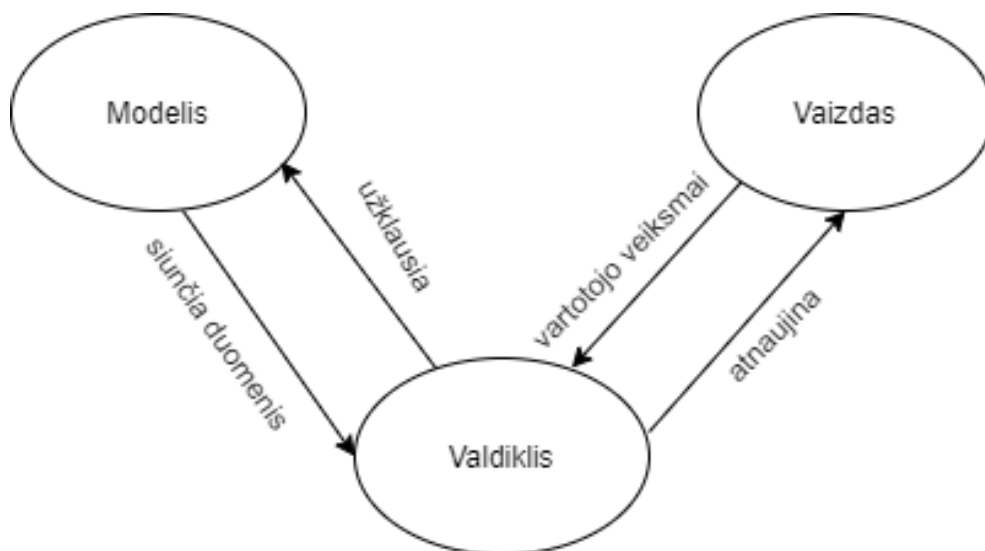
- Modelis (angl. model) yra objektas, kuris sąveikauja su duomenų baze. Modelis valdo aplikacijos duomenis ir elgseną, atsako į užklausas apie savo būseną ir atsako į instrukcijas pakeisti būseną. Aplikacijoje yra šie modeliai: Admin.php, AnimalRequest.php, Author.php, Category.php, City.php, Image.php, Like.php, Post.php, User.php. Šiuose modeliuose yra aprašyti ryšiai su kitais sistemos modeliais. Pavyzdžiui, duomenų bazės lentelė animal_requests turi išorinio rakto stulpelį user_id, kurį gauna iš users lentelės. Norint lengvai pasiekti users reikšmes, AnimalRequest modelyje aprašomas santykis su Users moduliui, kuris susijęs su users duomenų bazės lentele.

5 išeities kodas. AnimalRequest modelis

```
1 public function user()  
2     {  
3         return $this->belongsTo(User::class);  
4     }
```

Tuo tarpu, Users modelyje specifikuojama, koks ryšys sieja jį su AnimalRequest modeliu. Šiuo atveju, tai yra “hasMany” ryšys, kuris reiškia, kad AnimalRequest modelis gali turėti daug naudotojų.

- Vaizdas (angl. view) yra modelio būsenos vizualizacija. Vaizdai valdo grafinę arba tekstinę išeitį. Vaizdai atvaizduoja modelio turinį, gaudami duomenis per modelį ir specifikuodami kaip gauti duomenys turi būti vaizduojami.
- Valdiklis (angl. controller) duoda galimybę keisti modelio būseną. Valdiklis interpretuoja pelės ir klaviatūros įvestis, gautas iš sistemos naudotojo ir duoda komandas modeliui ir/arba vaizdui pasikeisti, pagal tai, kokie duomenys buvo įvesti. Kontroleris išverčia sąveiką su vaizdu į veiksmus, kuriuos atlieka modelis. Naudotojo sąveika su veizdu gali būti mygtukų paspaudimai, pasirinkimai iš meniu. Internetinėje aplikacijoje naudotojo interakcija su vaizdu atvaizduojamos kaip HTTP GET ir POST užklausos. Šioje sistemoje yra šeši paprasto naudotojo kontroleriai: Controller.php, kuris atsakingas už užklausų autorizavimą, darbų (angl. jobs) išsiuntimą ir užklausų validavimą, HomeController.php atsakingas už pagrindiniame puslapyje esantį filtravimo funkcionalumą, PostController.php atsakingas už veiksmus su sistemoje esančiais įrašais: įrašo pamėgimą, įrašo kategorijos užsiprenumeravimą, detalią įrašo peržiūrą ir anketos į gyvūno globą pildymą. ProfileController.php yra atsakingas už naudotojo profilio redagavimą, LoginController.php atsakingas už prisijungimą, o RegisterController.php atsakingas už naujo naudotojo registraciją. Taip pat yra tik administratoriaus teisę turinčio vartotojo kontroleriai: AuthorController.php, CategoryController.php, CityController.php, kurie leidžia valdyti (sukurti, redaguoti, ištrinti) sistemos objektus, PostController.php, kuris leidžia kurti, redaguoti, ištrinti, bet ne kurti sistemos įrašo objektus ir DashboardController.php, kuris atsakingas už administratoriaus panelę. Autoriaus privilegiją turintys vartotojai taip pat turi savo valdiklius: AnimalRequestsController.php, kuris atsakingas už užklausų į gyvūną patvirtinimą arba ištrynimą, PostController.php, atsakingą už įrašų kūrimą, rodymą, redagavimą ir ištrynimą ir ProfileController.php, atsakingą už profilio redagavimą.



4 pav. Model-view-controller veikimo principas.

3.5. Naujo gyvūno įrašo kūrimas ir valdymas

Gyvūno įrašo kūrimas, redagavimas ir ištrynimasis - funkcijos, kurias gali atlikti autoriaus privilegiją turintis naudotojas. Metodai, susiję su įrašais yra PostController valdiklyje. Autorius, norėdamas sukurti naują įrašą, privalo įvesti visus būtinuosius laukus: gyvūno kategoriją, miestą, adresą, aprašymą bei nuotraukas (žiūrėti 5 pav.) Redaguojant jau sukurtą įrašą, užpildyti laukai yra automatiškai užpildomi jau anksčiau įvesta informacija. Gyvūno įrašus redaguoti ir ištrinti taip pat gali administratoriaus privilegiją turintis narys.

ADD NEW POST

Category

City

Address

powered by Google

Images

5 pav. Gyvūno įrašo kūrimo anketa.

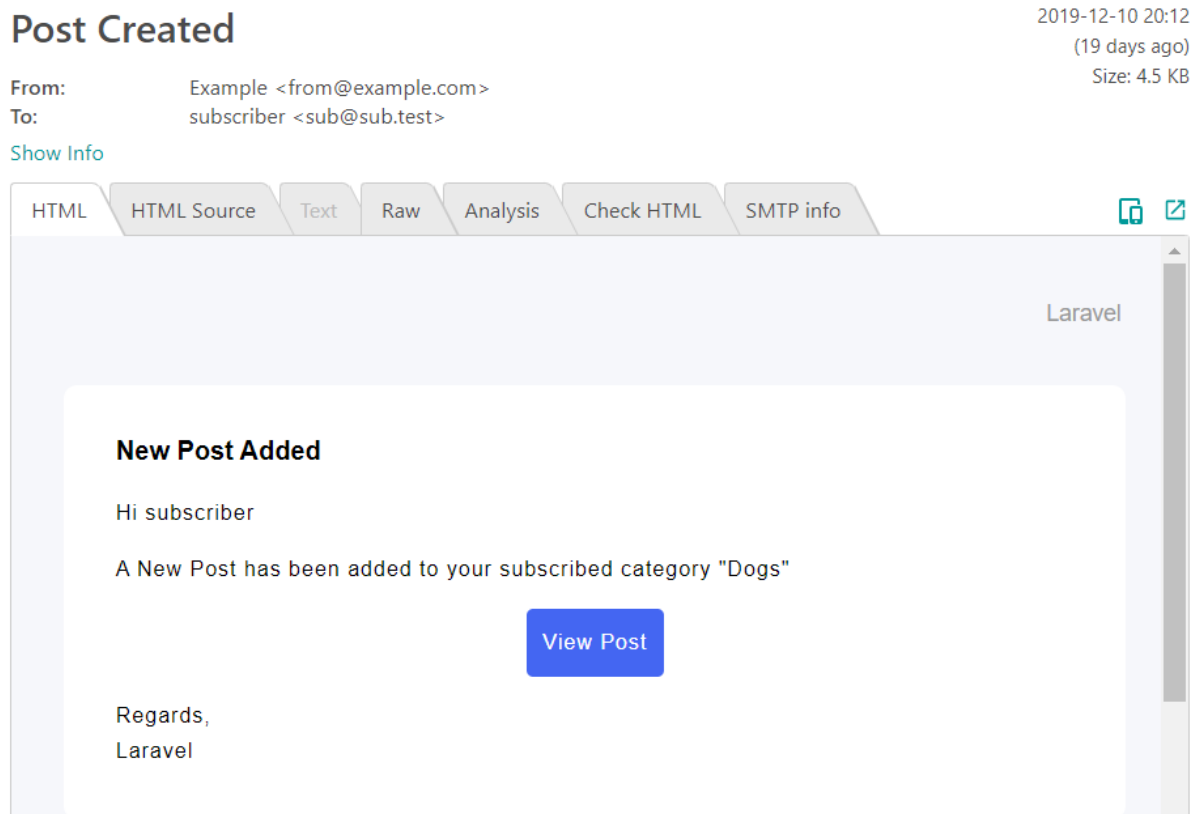
Pagal autoriaus įvestą adresą, sukurtą gyvūno įrašo puslapyje esančiame žemėlapyje yra vaizduojama gyvūno buvimo vieta. Iš pradžių gaunamas įvestas adresas, o po to, iš adreso gaunamos koordinatės: platumos ir ilgumos, kurios naudojamos nustatyti žymeklio vietai žemėlapyje.

6 išeities kodas. Koordinačių gavimas.

```
1 $address_data = $request->address ;
2 $address = $address_data['address'] ?? '';
3 $latitude = $address_data['latitude'] ?? '';
4 $longitude = $address_data['longitude'] ?? '';
```

3.6. Įrašų prenumeravimas

Siekiant padidinti naujų gyvūnų įrašų matomumą buvo sukurtas automatinis pranešimų siuntimo funkcionalumas. Kiekvienas prisijungęs naudotojas gali užsiprenumeruoti pasirinktos gyvūnų kategorijos pranešimus, kurie bus siunčiami į jo elektroninio pašto adresą. Norint tą padaryti, reikia užvesti pelę ant gyvūno nuotraukos, kuris yra kategorijoje, kurios pranešimus naudotojas nori gauti. Užvedus, reikia spustelėti varpelio ikoną. Jeigu norima atsisakyti pranešimų, reikia dar kartą paspausti ant varpelio ikonos. Apie sėkmingą prenumeravimo įvykį praneša sistemoje integruoti skrudintuvo (angl. toaster) pranešimai. Kiekvieną kartą, kai užprenumeruotoje kategorijoje bus sukurtas naujas įrašas, naudotojas gaus pranešimą į pateiktą elektroninio pašto adresą su nuoroda į naują įrašą.



6 pav. Pranešimas apie naują įrašą.

Siųsti pranešimus panaudota API tvarkyklė „Mailgun.“ Siunčiant laiškus, naudojamas „Mail“ metodas pavadinimu to(). Šis metodas priima el pašto adresą, vartotojo instanciją arba vartotojų kolekciją. Jeigu duodamas objektas arba objektų kolekcija, tvarkyklė automatiškai naudos jų elektroninio pašto ir vardo savybes, tad verta patikrinti, ar šie atributai egzistuoja naudotojo objektuose. Norint išsiųsti elektroninį laišką, reikia perduoti „mailable“ klasę į send() metodą:

7 išeities kodas. Pranešimų siuntimo kodas.

```
1 Mail :: to ($user) -> send (new PostCreated ($post , $user));
```

3.7. Aplikavimas į gyvūno globą

Norint palengvinti aplikavimo į gyvūno globą procesą, buvo sukurtas funkcionalumas, leidžiantis registruotam vartotojui tiesiogiai siųsti prašymus skelbimą įkėlusiai asociacijai. Norint tai padaryti, tereikia nueiti į detalų norimo gyvūno puslapį ir paspausti mygtuką „want it.“ Mygtuko paspaudimas aktyvuos modalinį langą, kuriame naudotojas turės užpildyti tokią informaciją, kaip vardą, kontaktinį numerį, dabartinį turimų gyvūnų skaičių, pajamas bei priežastį, dėl kurios norima pasiimti globoti gyvūną (7 pav.).

Apply for this animal

Your Name	Contact No
Current Number of Pets	Annual Income
Reason	
Apply Cancel	

7 pav. Aplikavimo į gyvūno globą anketa.

Tuo tarpu, įrašo autorius, savo panelėje matys visus prašymus į gyvūno globą ir galės juos atmesti arba patvirtinti. Patvirtinus prašymą, į aplikavusio vartotojo elektroninio pašto bus išsiųstas laiškas, pranešantis apie sėkmingą aplikaciją su tolimesniais skelbimo autoriaus duomenimis.

3.8. Duomenų filtravimas

Esant dideliame duomenų kiekiui ir norint rasti aktualią informaciją, labai svarbu turėti duomenų filtravimą [9]. Sistemoje filtravimą gali atlikti bet kuris naudotojas, tačiau aktualiausias jis - paprastam vartotojui. Prisijungęs vartotojas gali filtruoti įrašus pagal savo favoritus, pagal kategoriją bei pagal miestą. Galima filtruoti pagal vieną arba kelias reikšmes. Norint atsisakyti filtro, galima dar kartą paspausti ant reikšmės arba paspausti „Reset all“. Ši komanda atstato visus filtrus

į pradinę būseną.

8 išeities kodas. Filtravimas pagal favoritus, kategorijas ir miestus.

```
1 if($favorites = \request()->favorites){
2     if(auth()->check()){
3         $posts_q->whereIn('id', auth()->user()->likes->pluck('
4             post_id')->toArray());
5     }
6 }
7 if($category_ids = \request()->categories){
8     $posts_q->whereIn('category_id', $category_ids);
9 }
10 if($city_ids = \request()->cities){
11     $posts_q->whereIn('city_id', $city_ids);
12 }
```

8 kodo išeityje matoma, kad filtravimas pagal favoritus veikia tik tada, jei vartotojas yra prisijungęs. Kitiems filtravimo tipams, pagal kategoriją ir miestą, prisijungti nereikia.

Filter Posts 

[Reset All](#)

Personalization
☐ Favorites (1)

Categories
☐ Other (1)
☐ Cats (0)
☐ Dogs (1)

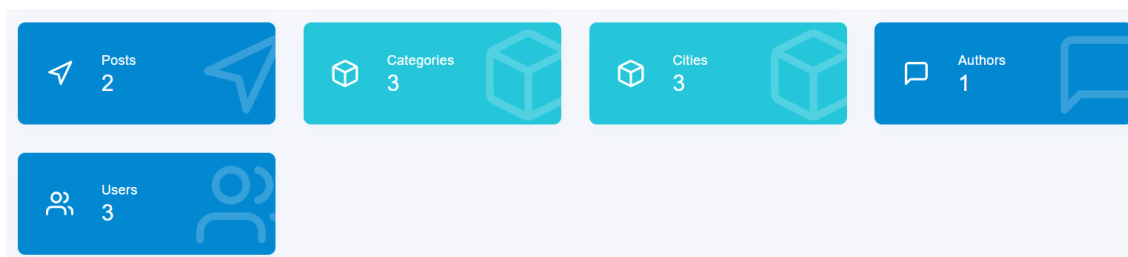
Cities
☐ Vilnius (1)
☐ Kaunas (0)
☐ Other (1)

Filter

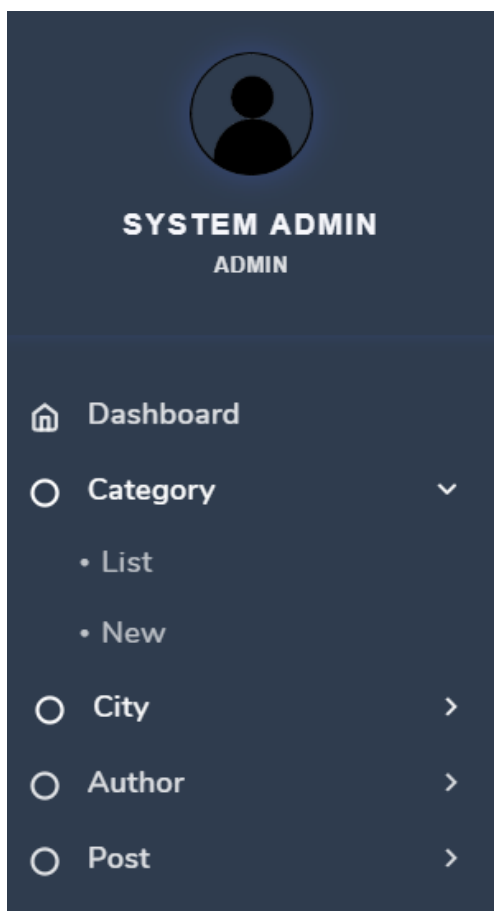
8 pav. Įrašų filtravimo pasirinkimai

3.9. Administratoriaus panelė

Sistemoje daugiausiai galimybių turintis naudotojas yra administratorius. Norint palengvinti sistemos administravimą, sukurta speciali panelė valdymui. Norint ją pasiekti, reikia būtinai prisijungti su administratoriaus tipo naudotoju. Panelės pagrindiniame puslapyje galima matyti puslapio statistiką: bendrus įrašų, kategorijų, miestų, autorių bei vartotojų skaičius (9 pav.) Panelės šoniniame skyriuje (žiūrėti 10 pav.) taip pat yra kategorijų, miestų, autorių ir įrašų valdymo pasirinkimai, siekiant sukurti paprastą ir intuityvią vartotojo patirtį. Norint valdyti kurią nors iš kategorijų, pvz miestus, tereikia paspausti ant pavadinimo. Tada JavaScript pagalba išsiskleidžia tolimesni veiksmai: galima peržiūrėti visus sukurtus miestus, spaudžiant ant „List“ arba sukurti naują miestą, spaudžiant ant „New.“



9 pav. Administratoriaus panelės statistika



10 pav. Administratoriaus panelės šoninis skyrius

3.10. Saugumo sprendimai

Norint užtikrinti sistemos saugumą, pasinaudota Laravel karkaso siūlomomis saugumo funkcijomis:

- Laravel karkasas naudoja CSRF prieigos žymę (angl. token), užtikrinti, kad pašaliniai tretieji asmenys negalėtų siųsti netikrų užklausų. Tam įvykdyti, Laravel į kiekvieną užklausą, kuri ateina iš naudotojo siunčiamos formos, sukuria ir integruoja prieigos žymę. Kai užklausa yra išskviečiama, Laravel palygina užklausos prieigos žymę su žyme, kuri yra išsaugota naudotojo sesijoje. Kad šis funkcionalumas veiktų, į sistemą, prie kiekvienos formos buvo pridėdamas „@csrf“ pagalbininkas [4].
- Naudojantis Laravel, slaptažodžiai duomenų bazėje nėra saugomi paprastu tekstu. Karkasas naudoja „bcrypt“ slaptažodžių šifravimą, kas užtikrina vartotojų prisijungimo saugumą [7].
- Kiekvienoje sistemoje svarbu, kad skirtingas privilegijas turintys nariai galėtų pasiekti tik jų privilegijai būdingas funkcijas, pavyzdžiui paprastas sistemos naudotojas negalėtų pasiekti administratoriaus panelės. Saugumui užtikrinti buvo panaudota tarpinė programinė įranga (angl. middleware).

Išvados ir rekomendacijos

Atlikus panašių sistemų, jau esančių rinkoje, analizę, buvo pastebėta, kad šiuo metu nėra sistemos, kuri atitiktų visus bakalaurnio darbo metu sukurtos sistemos reikalavimus. Praktiškai sukurta sistema, leidžianti vartotojams lengviau ir efektyviau ieškoti gyvūnų, gauti automatinius pranešimus bei pretenduoti į jų globą. Gyvūnų globos namams sukurtas patogus būdas kelti įrašus bei priimti aplikacijas į gyvūno globą: visas anketas jie gali matyti vienoje vietoje, kas palengvina sprendimą, kuriam pretendentui paskirti gyvūno globojimą. Administratoriai taip pat gali lengvai naudotis savo panelėje esančiomis valdymo funkcijomis: matyti bendrą statistiką, kontroliuoti miestus, kategorijas, autorius bei redaguoti ar ištrinti įrašus.

Didžiąją dalį sistemos įgyvendinimo užėmė CRUD operacijos. Kitą nemažą laiko tarpą - Mailgun integracija, norint siųsti automatinius pranešimus naudotojams. Kuriant sistemą gauta daug naudingos techninės informacijos, kaip kad MVC modelio pritaikymas, integracija su išorinėmis sistemomis. Taip pat susipažinta su gyvūnų globos puslapių sistemų dabartine situacija.

Sistemai, žinoma, yra vietos ir tobulėjimui. Sekantys veiksmai galėtų būti tokie:

- Įrašų autoriai galėtų gauti pranešimus, kai kažkas užpildo aplikaciją į gyvūno globą.
- „Padovanotų gyvūnų“ kategorija leistų vartotojams pasidžiaugti, kad gyvūnai surado namus.

Literatūros šaltiniai

- [1] Bitdegree tutorials: Most popular php frameworks: Which should you choose? <https://www.bitdegree.org/tutorials/php-frameworks/>.
- [2] Laravel docs: Blade templates. <https://laravel.com/docs/5.7/blade>.
- [3] Laravel docs: Eloquent orm. <https://laravel.com/docs/5.0/eloquent#introduction>.
- [4] Martin Bean. *Laravel 5 Essentials*. 2015.
- [5] David Lane Hugh E. Williams. *Web Database Applications with PHP and MySQL: Building Effective Database-Driven Web sites*. 2009.
- [6] Hamed Halimi Ishaq Jound. *Comparison of performance between Raw SQL and Eloquent ORM in Laravel*. 2016.
- [7] Swati D. Ghule Kausar I. Bagwan. *A Modern Review on Laravel- PHP Framework*. 2019.
- [8] Nora Koch Maria José Escalona. *Metamodeling the Requirements of Web Systems*. 2007.
- [9] Surendra Goel Thomas E. Donaldson, David E. Kellum. *Filtering search results*.