

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

Rytis Mičiūnas

**ASMENINĖ ŽVEJO GIDO PROGRAMĖLĖ**

**PERSONAL FISHING GUIDE MOBILE APPLICATION**

Bakalauro baigiamasis darbas

Programų inžinerijos studijų programa, valstybinis kodas 6121BX023 Programų sistemų studijų kryptis

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS INFORMACINIŲ SISTEMŲ KATEDRA

Rytis Mičiūnas

ASMENINĖ ŽVEJO GIDO PROGRAMĖLĖ

**PERSONAL FISHING GUIDE MOBILE APPLICATION**

Baigiamasis bakalauro darbas

Programų inžinerijos studijų programa, valstybinis kodas 6121BX023 Programų sistemų studijų kryptis

**Vadovas**  dr. Arūnas Ribikauskas

(Pedag. vardas, vardas, pavardė)

#### Konsultantas

(Pedag. vardas, vardas, pavardė)

#### Konsultantas

(Pedag. vardas, vardas, pavardė)

Turinys

[Įvadas 1](#_Toc198055463)

[1. Mėgėjiška žvejyba Lietuvoje bei sistemų apžvalga ir palyginimas. 3](#_Toc198055464)

[1.1. Žvejybos svarba Lietuvių kultūroje ir gamtosaugoje 3](#_Toc198055465)

[1.2. Su žvejyba susijusios programėlės 6](#_Toc198055466)

[1.2.1. Internetinė svetainė „Žvejo gidas“ 6](#_Toc198055467)

[1.2.2. Mobilioji aplikacija „Fish Deeper“ 8](#_Toc198055468)

[1.2.3. Mobilioji programėlė „FishAIDetect“ 10](#_Toc198055469)

[1.3. Kuriamos programėlės apžvalga 12](#_Toc198055470)

[1.4. Žvejybinių sistemų palygynimas 15](#_Toc198055471)

[2. Technologijų analizė ir palyginimas 17](#_Toc198055472)

[2.1. Vaizdų atpažinimo modeliai 17](#_Toc198055473)

[2.2. Programavimo kalbų apžvalga ir palyginimas 18](#_Toc198055474)

[2.3. Duomenų bazių apžvalga ir palyginimas 19](#_Toc198055475)

[2.4 Dalinės išvados 21](#_Toc198055476)

[3. Sistemos poreikių ir reikalavimų specifikavimas 22](#_Toc198055477)

[3.1 Pagrindinės sistemos užduotys 22](#_Toc198055478)

[3.2 Funkciniai reikalavimai 31](#_Toc198055479)

[3.3 Nefunkciniai reikalavimai 32](#_Toc198055480)

[3.4 Skyriaus išvados 32](#_Toc198055481)

[4. Mobiliosios programėlės projektavimas ir realizavimas 33](#_Toc198055482)

[4.1 Sistemos architektūra 33](#_Toc198055483)

[4.2 Sekų diagramos 35](#_Toc198055484)

[4.3 Sistemos struktūrinis modelis 42](#_Toc198055485)

[4.4 Galutinio produkto dizainas 44](#_Toc198055486)

[4.5 Skyriaus išvados 48](#_Toc198055487)

[5. Testavimas 49](#_Toc198055488)

[5.1 Testavimo aprašymas 49](#_Toc198055489)

[4.2 Skyriaus išvados 59](#_Toc198055490)

[Išvados 60](#_Toc198055491)

[LITERATŪROS ŠALTINIAI 61](#_Toc198055492)

[1 PRIEDAS (išrtrauka iš aplinkos apsaugos departamento prie aplinkos ministerijos gyvosios gamtos apsaugos departamento atsiųstų duomenų) 64](#_Toc198055493)

PAVEIKSLŲ SĄRAŠAS

[1 pav. Kauno Marių visų žuvų santykinis gausumas ir biomasė 1997–2017 m. 3](#_Toc198055494)

[2 pav. Kauno Marių karšių santykinis gausumas ir biomasė 1997–2017 m. 4](#_Toc198055495)

[3 pav. 291 straipsnio 1 ir 2 dalies 2019-2024 metų nusižengimai 5](#_Toc198055496)

[4 pav. „Žvejo gidas“ pagrindinis pulapis 7](#_Toc198055497)

[5 pav. „Žvejo gidas“ Gėlužės ežero puslapis 8](#_Toc198055498)

[6 pav. „Fish Deeper“ programėlės funkcionalumas 9](#_Toc198055499)

[7 pav. „FishAIDetect“ dirptinis intelektas atpažysta žuvį iš nuotraukos 10](#_Toc198055500)

[8 pav. „FishAIDetect“ socialinis tinklas 11](#_Toc198055501)

[9 pav. Dabartį atspindintis vaizdusis paveiksliukas 12](#_Toc198055502)

[10 pav. Siekiamos būsenos vaizdusis paveikslėlis 14](#_Toc198055503)

[11 pav. Sistemos užduočių diagrama 23](#_Toc198055504)

[12 pav. Autentifikacijos užduotis 24](#_Toc198055505)

[13 pav. Atpažinimo užduotis 25](#_Toc198055506)

[14 pav. Žemėlapio užduotis 27](#_Toc198055507)

[15 pav. Statistiko užduotis 28](#_Toc198055508)

[16 pav. Žvejo bilieto užduotis 29](#_Toc198055509)

[17 pav. Architektūros diagrama 33](#_Toc198055510)

[18 pav. Loginė architektūros diagrama 35](#_Toc198055511)

[19 pav. Prisijungimo sekų diagrama 36](#_Toc198055512)

[20 pav. Registracijos sekų diagrama 37](#_Toc198055513)

[21 pav. Žuvies atpažinimo ir informacijos grąžinimo sekos diagrama 38](#_Toc198055514)

[22 pav. Žemėlapio atidarymo sekos diagrama 39](#_Toc198055515)

[23 pav. Žymeklio kūrimo sekos diagrama 39](#_Toc198055516)

[24 pav. Statistikos formavimo sekos diagrama 40](#_Toc198055517)

[25 pav. Žvejo bilieto atvaizdavimo sekos diagrama 41](#_Toc198055518)

[26 pav. Žvejo bilieto atnaujinimo sekos diagrama 42](#_Toc198055519)

[27 pav. Duomenų bazės modulio diagrama 43](#_Toc198055520)

[28 pav. Paleidimo ir prisijungimo ekranų dizainai 45](#_Toc198055521)

[29 pav. Žymeklių ir žemėlapio ekranų dizainas 46](#_Toc198055522)

[30 pav. Žvejo bilieto ekrano dizainas 47](#_Toc198055523)

[31 pav. Statistikos ekrano dizainas 48](#_Toc198055524)

LENTELIŲ SĄRAŠAS

[1 lentelė Programėlių funkcionalumų palyginimas 15](#_Toc198055525)

[2 lentelė Prisijungimo scenarijus 24](#_Toc198055526)

[3 lentelė Registracijos scenarijus 24](#_Toc198055527)

[4 lentelė Atsijungimo scenarijus 25](#_Toc198055528)

[5 lentelė Atpažinimo darant nuotrauką scenarijus 26](#_Toc198055529)

[6 lentelė Atpažinimo įkeliant nuotrauką scenarijus 26](#_Toc198055530)

[7 lentelė Matyti senus žymeklius scenarijus 27](#_Toc198055531)

[8 lentelė Sukurti naują žymeklį scenarijus 28](#_Toc198055532)

[9 lentelė Statistikos užduotis 29](#_Toc198055533)

[10 lentelė Įkelti žvejo bilietą PDF formatu užduotis 30](#_Toc198055534)

[11 lentelė Įrašyti žvejo bilietą ranka užduotis 30](#_Toc198055535)

[12 lentelė Pažiūrėti žvejo bilietą užduotis 30](#_Toc198055536)

[13 lentelė Funkciniai reikalavimai 31](#_Toc198055537)

[14 lentelė Nefunkciniai reikalavimai 32](#_Toc198055538)

[15 lentelė Testų secenarijai ir atvėjai 49](#_Toc198055539)

[16 lentelė Prisijungimas teisingi duomenys 50](#_Toc198055540)

[17 lentelė Prisijungimas neteisingi duomenys 51](#_Toc198055541)

[18 lentelė Prisijungimas neįvedami duomenys 51](#_Toc198055542)

[19 lentelė Registracija teisingi duomenys 52](#_Toc198055543)

[20 lentelė Registracija nesutampa slaptažodis 52](#_Toc198055544)

[21 lentelė Registracija tušti duomenys 53](#_Toc198055545)

[22 lentelė Atpažynimas fotografuojama žuvis 53](#_Toc198055546)

[23 lentelė Atpažynimas fotografuojama ne žuvis 53](#_Toc198055547)

[24 lentelė Atpažynimas įkeliama žuvies nuotrauka 54](#_Toc198055548)

[25 lentelė Atpažynimas įkeliama ne žuvies nuotrauka 54](#_Toc198055549)

[26 lentelė Atpažynimas įkeliama ne nuotrauka 55](#_Toc198055550)

[27 lentelė Žemėlapio žymeklis užpildomi visi duomenys 55](#_Toc198055551)

[28 lentelė Žemėlapio žymeklis užpildomi būtini duomenys 56](#_Toc198055552)

[29 lentelė Žemėlapio žymeklis nepildomi duomenys 56](#_Toc198055553)

[30 lentelė Statistika, skirtingi laikotarpiai 57](#_Toc198055554)

[31 lentelė Žvejo bilietas teisingas PDF 57](#_Toc198055555)

[32 lentelė Žvejo bilietas ne PDF 58](#_Toc198055556)

[33 lentelė Žvejo bilietas ranka įrašoma teisinga informacija 58](#_Toc198055557)

[34 lentelė Žvejo bilietas neįrašomi duomenys 59](#_Toc198055558)

[35 lentelė Atsijungimas 59](#_Toc198055559)

# Įvadas

Žvejyba yra vienas populiariausių laisvalaikio užsiėmimų Lietuvoje, kuris suteikia galimybę ne tik atsipalaiduoti gamtoje, bet ir pasimėgauti kulinarinėmis savybėmis pasižyminčiomis Lietuvos žuvimis. Vis dėlto, žvejams mėgėjams tenka laikytis tam tikrų teisinių reikalavimų, susijusių su žvejybos taisyklėmis, kurios dažnai priklauso nuo žuvies rūšies, dydžio, sugavimo laikotarpio ar vietos. Šių taisyklių pažeidimai gali užtraukti administracinę atsakomybę bei nuosavybės konfiskavimą, tačiau, ypač pradedantiesiems žvejams, gali būti sudėtinga atsiminti visas taisykles ar atpažinti tam tikras žuvų rūšis. Be to, pagavus žuvį, dažnai nėra pakankamai laiko ją identifikuoti, surasti atitinkamas taisykles ir jas tinkamai interpretuoti, nes kai kurios žuvų rūšys jautriai reaguoja į temperatūros pokyčius, deguonies trūkumą ar kitus aplinkos veiksnius po ištraukimo iš vandens (Ferguson ir Tufts, 1992; Schreer ir kt., 2005).

Technologijų pažanga sudaro galimybes palengvinti žvejų mėgėjų užduotis. Dirbtinio intelekto ir mobiliųjų technologijų plėtra leidžia automatizuoti žuvų atpažinimą pagal nuotraukas bei realiu laiku pateikti reikiamą informaciją apie žvejybos taisykles. Be to, išmaniosios programėlės gali sujungti kelis funkcionalumus, kurie žvejams yra itin naudingi, pavyzdžiui, sugavimo vietų žymėjimą, statistikos generavimą pagal konkrečius vandens telkinius, licencijų valdymą ir galiojimo priminimus.

**Darbo problema** – nors žvejybos taisyklės yra viešai prieinamos, jų sudėtingumas ir neorganizuotas pateikimas apsunkina žvejų mėgėjų galimybes lengvai ir efektyviai laikytis nustatytų reikalavimų.

**Darbo objektas** – žvejo gido mobiliosios programėlės prototipas.

**Darbo tikslas** – palengvinti Lietuvos žvejų mėgėjų pareigą laikytis žvejybos taisyklių, naudojant dirbtinio intelekto technologijas žuvų atpažinimui, informacijai apie reglamentus, sugavimo vietų žymėjimui, statistikos generavimui ir licencijų valdymui.

#### Tikslui pasiekti numatyti uždaviniai:

1. Atlikti žvejybinių sistemų apžvalgą ir palyginimą.
2. Aprašyti galimų technologijų analizę ir palyginimą.
3. Suformuluoti funkcinius ir nefunkcinius reikalavimus, bei suprojektuoti programėlės architektūrą.
4. Sukurti ir ištestuoti žvejo gido programėlės žuvų atpažinimą bei kitus programos funkcionalumus.

#### Darbo naujumas ir praktinė jo vertė

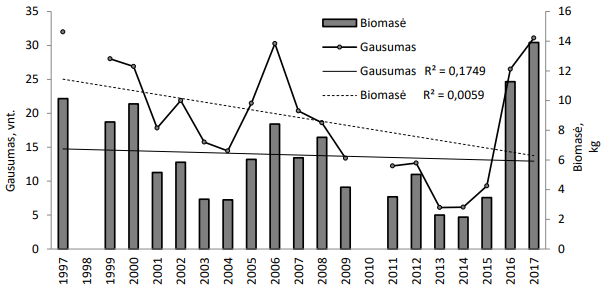
Planuojamos programinės sistemos komponentai daugeliu atvejų nėra visiškai nauji Lietuvos rinkai pritaikytose sistemose. Tačiau rinkoje vis dar nėra integruotos platformos, kuri vienytų visas reikalingas funkcijas vienoje vietoje. Tokios programos sukūrimas eliminuotų būtinybę vartotojams naudotis skirtingomis programomis pagrindiniams žvejybos poreikiams patenkinti, o tai leistų sutaupyti išmaniojo telefono atmintį bei pagerintų vartotojo patirtį. Nors pasaulyje dirbtinis intelektas jau naudojamas žuvims atpažinti, esamos programos nepateikia Lietuvoje galiojančių žvejybos taisyklių, susijusių su identifikuota žuvimi. Sukurta sistema, gebanti tiesiogiai susieti žuvų atpažinimą su Lietuvos žvejybos taisyklėmis, padėtų žvejams išvengti administracinių nusižengimų bei prisidėtų prie žuvų išteklių tausojimo ir gamtos apsaugos. Be to, kai kurios esamos programos leidžia pažymėti vietas, kuriose buvo sugautas laimikis, tačiau rinkoje nėra sprendimų, kurie automatizuotai generuotų išsamią statistinę informaciją apie konkretų vandens telkinį. Tokia statistika galėtų apimti vidutinį pagaunamų žuvų dydį, vyraujančias žuvų rūšis, žvejybos sėkmingumo rodiklius (pavyzdžiui, kaip dažnai pagaunama žuvis konkrečioje vietoje) ir kitas svarbias charakteristikas. Tokia programa ne tik palengvintų žvejų mėgėjų veiklą, bet ir sudarytų prielaidas tikslesnei duomenų analizei, prisidėtų prie tvarios mėgėjiškos žvejybos bei būtų naudinga moksliniams tyrimams apie Lietuvos žuvų populiaciją ir jų pasiskirstymą įvairiuose vandens telkiniuose.

## Mėgėjiška žvejyba Lietuvoje bei sistemų apžvalga ir palyginimas.

Šiame skyriuje bus nagrinėjama mėgėjiškos žvejybos reikšmė Lietuvoje, jos įtaka gamtos ištekliams ir svarba aplinkosaugos kontekste. Taip pat bus analizuojamos esamos su žvejyba susijusios sistemos Lietuvoje, įvertinant jų funkcionalumus, privalumus ir trūkumus. Remiantis šia apžvalga bus išryškinti esami sprendimai ir pagrįstas planuojamos programinės įrangos kūrimo poreikis.

## Žvejybos svarba Lietuvių kultūroje ir gamtosaugoje

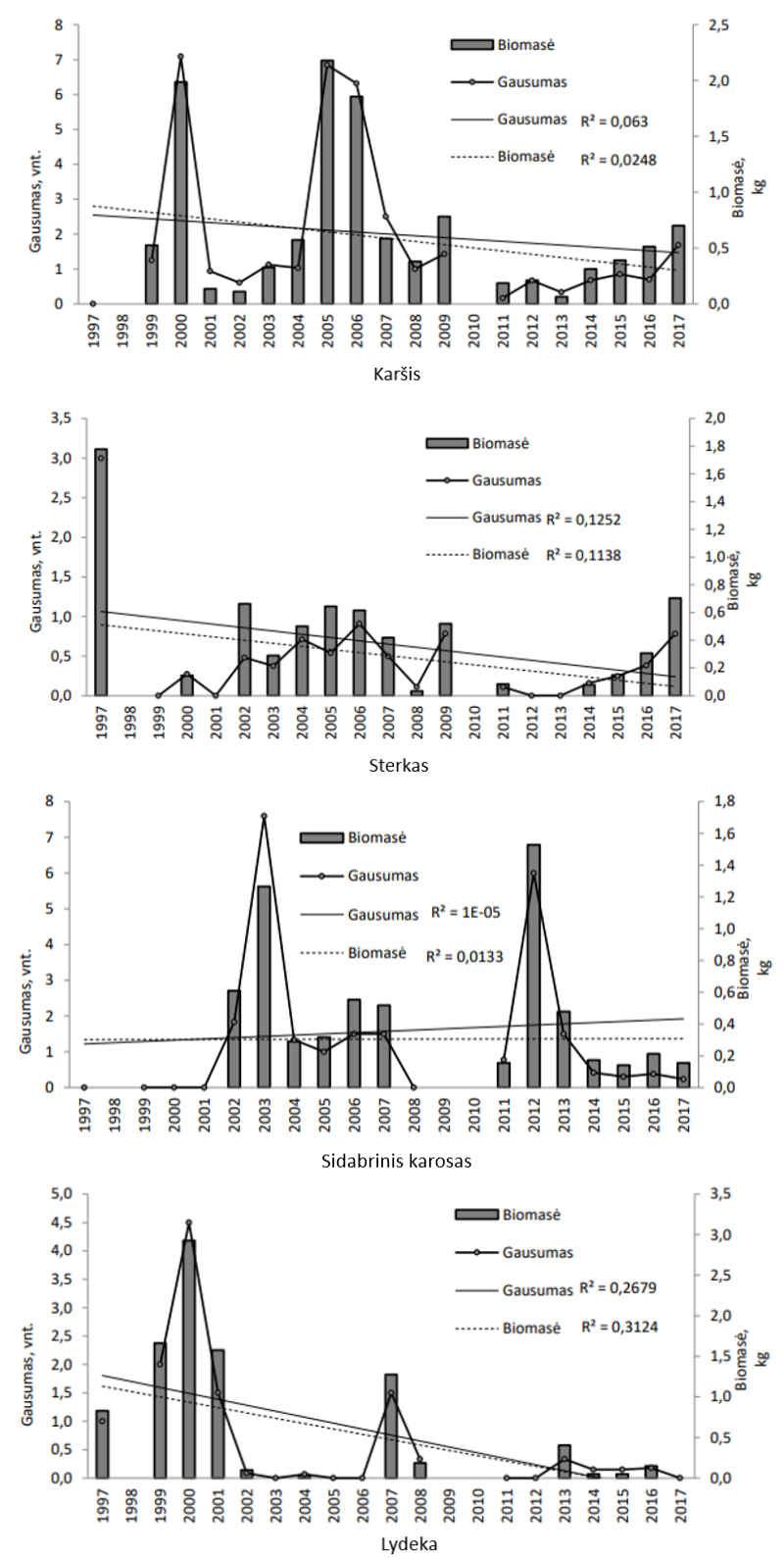
Žvejyba yra viena seniausių žmonijos veiklų, turinti gilias tradicijas tiek pasaulyje, tiek Lietuvoje. Tačiau intensyvi žmogaus veikla, įskaitant žvejybą, gali išeikvoti žuvų išteklius ar net visiškai sunaikinti tam tikras rūšis. Dėl šios priežasties yra būtina nuolat stebėti žuvų populiacijas, prireikus vykdyti jų veisimą (žuvinimą) ir griežtai reglamentuoti tiek mėgėjišką, tiek verslinę žvejybą (Kesminas ir Stakėnas, 2020). Nors mėgėjiška žvejyba dažnai laikoma mažesnį poveikį turinčia veikla, palyginti su versline žvejyba, tyrimai rodo, kad ji taip pat gali reikšmingai paveikti žuvų išteklius. Pavyzdžiui, po 2013 m. įvesto draudimo vykdyti verslinę žvejybą Kauno mariose, žuvų ištekliai stipriai atsigavo. Žemiau pavaizduotas grafikas parodo 1997–2017m. darytus tyrimus naudojant 38–55 mm akies dydžio žiauninius tinklus.



1 pav. Kauno Marių visų žuvų santykinis gausumas ir biomasė 1997–2017 m.

Šaltinis: (Ložys ir kt., 2017)

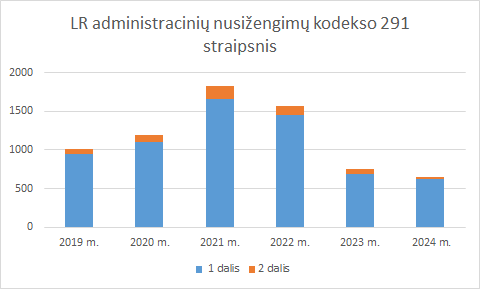
Tyrimo rezultatai aiškiai rodo didelį žuvų pagausėjimą Kauno mariose nuo 2015 m. Tačiau rūšių, kurios buvo aktyviai žvejojamos žvejų mėgėjų, būklė pagerėjo tik nežymiai arba iš vis nepagerėjo (Dainys ir kt., 2023). 1997–2017 m. darytas tyrimas irgi patvirtina tai. Žemiau parodyti karšių, sterkų, sidabrinių karosų ir lydekų tyrimų rezultatai.



2 pav. Kauno Marių karšių santykinis gausumas ir biomasė 1997–2017 m.

Šaltinis: (Ložys ir kt., 2017)

Taigi iš tyrimo rezultatų galime pamatyti, kad lyginant su bendru žuvų kiekiu pagerėjimu, po verslinės žvejybos draudimo, karšių ir sterkų biomasė ir gausumas auga lėčiau, o sidabrinių karosų ir lydekų rezultatai yra net blogesni nei verslinės žvejybos laikotarpiu. Šis pavyzdys puikiai pabrėžia mėgėjiškos žvejybos poveikį gamtos ištekliams. Dėl to yra būtina reguliuoti mėgėjišką žvejybą, tačiau tai nėra paprasta užduotis. Žvejams dažnai būna sudėtinga suprasti ir laikytis visų galiojančių teisės aktų, kartais ir atpažinti skirtingas žuvų rūšis. Taip pat Lietuvos įstatymai periodiškai keičiasi ir gali būt sudėtinga sekti ir identifikuoti visus atnaujinimus. Pavyzdžiui, 2021 m. minimalus lydekos ilgis padidintas iki 50 centimetrų, o maksimalus leistinas kiekis – iki 2 lydekų per vieną žvejybą (*D1-272 Dėl Lietuvos Respublikos Aplinkos Ministro 2013 M. Sausio 4 D. Įsakymo Nr. D1-14 „Dėl Mėgėjų Žvej. . .*, 2021 m.). 2024 m. įsigaliojo dar ir maksimalus imamų lydekų ilgis 80 centimetrų (*D1-136 Dėl Lietuvos Respublikos Aplinkos Ministro 2013 M. Sausio 4 D. Įsakymo Nr. D1-14 „Dėl Mėgėjų Žvej. . .*, 2024 m.). Kadangi Lietuvoje nėra jokių egzaminų žvejams ar kitokių priverstinių priemonių mėgėjiškos žvejybos taisyklių žinių atnaujinimui, žvejai, kurie užsiima šita veikla senai ir iš seno žino visus įstatymus, gali netyčia padaryti žalą Lietuvos gamtai ir užsitraukti administracinę atsakomybę. Žemiau pateikiamas grafikas, vaizduojantis pažeidimų, užfiksuotų pagal Lietuvos Respublikos administracinių nusižengimų kodekso 291 straipsnio 1 ir 2 dalis nuo 2019 iki 2024 metų, skaičių. Šios administracinių nusižengimų kodekso dalys reglamentuoja mėgėjiškos žvejybos taisyklių pažeidimus, išskyrus pažeidimus, susijusius su draudžiamų įrankių naudojimu ar limituotos žvejybos lašišinės giminės žuvų sąlygų nesilaikymu.



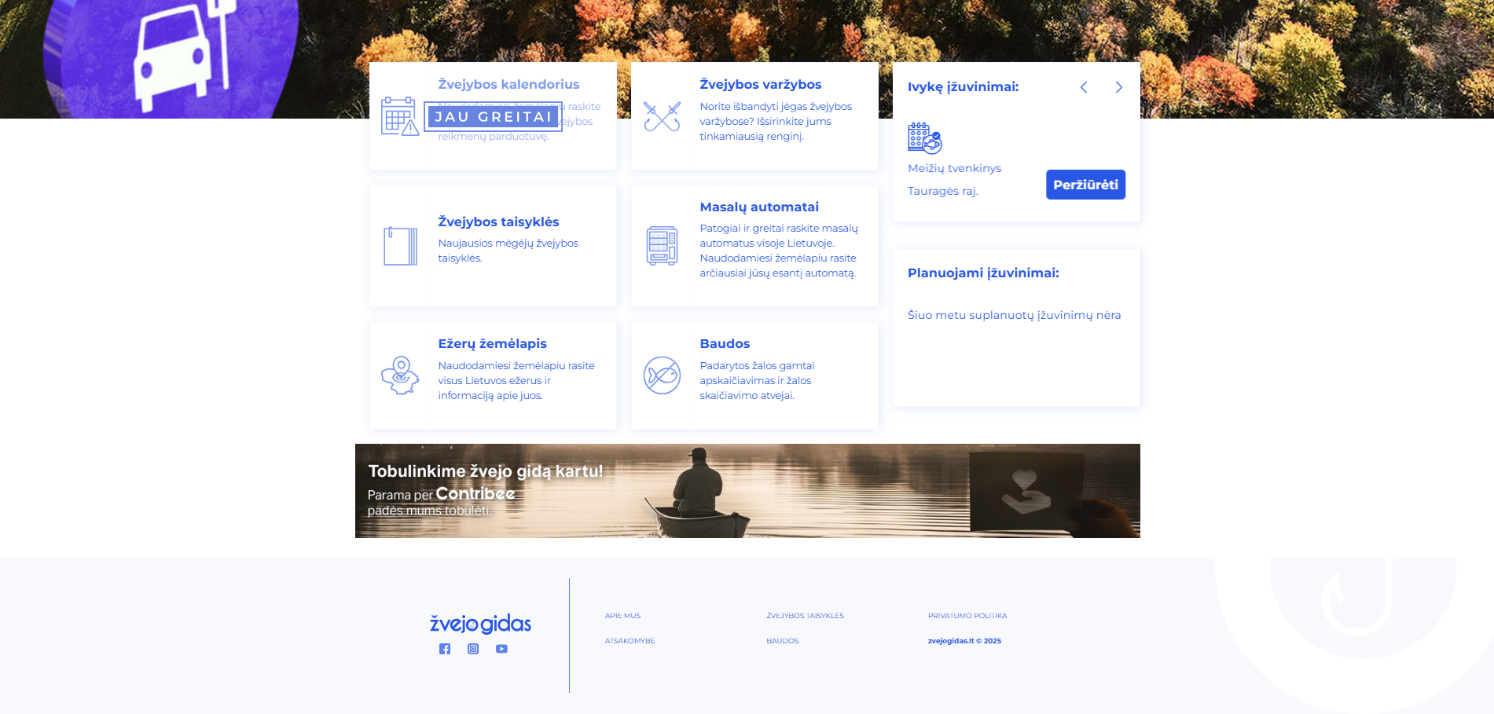
3 pav. 291 straipsnio 1 ir 2 dalies 2019-2024 metų nusižengimai

Šaltinis: (APLINKOS APSAUGOS DEPARTAMENTO PRIE APLINKOS MINISTERIJOS GYVOSIOS GAMTOS APSAUGOS DEPARTAMENTAS, 2025)

Grafikas parodo administracinių nusižengimų, susijusių su mėgėjų žvejybą reglamentuojančių teisės aktų pažeidimais, dinamiką pagal Lietuvos Respublikos Administracinių nusižengimų kodekso 291 straipsnio 1 ir 2 dalis nuo 2019 iki 2024 metų. Nuo 2019 iki 2022 metų buvo stebimas nuoseklus administracinėn atsakomybėn patrauktų asmenų skaičiaus augimas. Šis augimas galėjo būti nulemtas padidėjusio žvejybos intensyvumo, nepakankamo žvejų informuotumo apie galiojančius teisės aktus arba sustiprėjusios pažeidimų fiksavimo kontrolės. Taip pat galėjo prisidėti tam tikrų teisės aktų pakeitimai ar sugriežtinimai, kurie išryškino tam tikras praktikoje dažnai nepastebėtas pažeidimų formas. Nuo 2023 metų stebima mažėjimo tendencija, kuri galimai sietina su geresniu žvejų informuotumu, efektyvesne prevencine veikla arba sumažėjusiu bendru žvejybos aktyvumu. Tačiau svarbu pažymėti, kad grafike pateikiami tik tie pažeidimai, kurie buvo oficialiai užfiksuoti ir už kuriuos asmenys buvo patraukti administracinėn atsakomybėn. Neatmestina, kad dalis pažeidimų liko nepastebėti arba buvo pritaikytas žodinis įspėjimas. Norint užtikrinti tvarią žvejybą Lietuvoje ir sumažinti taisyklių pažeidimų skaičių, būtina diegti modernias technologijas, padedančias žvejams realiuoju laiku suprasti ir laikytis teisės aktų. Specializuota žvejo gido programėlė galėtų tapti svarbiu įrankiu, ne tik padedančiu išvengti pažeidimų, bet ir prisidedančiu prie sąmoningo gamtos išteklių naudojimo. Tuo pačiu, siekiant ilgalaikio žuvų populiacijų išsaugojimo ir vandens telkinių biologinės įvairovės išlaikymo, būtina skirti daugiau dėmesio žuvų išteklių stebėsenai. Reguliarios stebėsenos vykdymas suteiktų vertingų duomenų apie populiacijų būklę, leistų laiku reaguoti į pokyčius ir užtikrintų tvarią gamtos išteklių apsaugą. Šios priemonės neabejotinai būtų naudingos ne tik aplinkosaugai, bet ir moksliniams tyrimams, stiprinant ryšį tarp teorinių žinių ir praktinių sprendimų (Duchovskis ir kt., 2022).

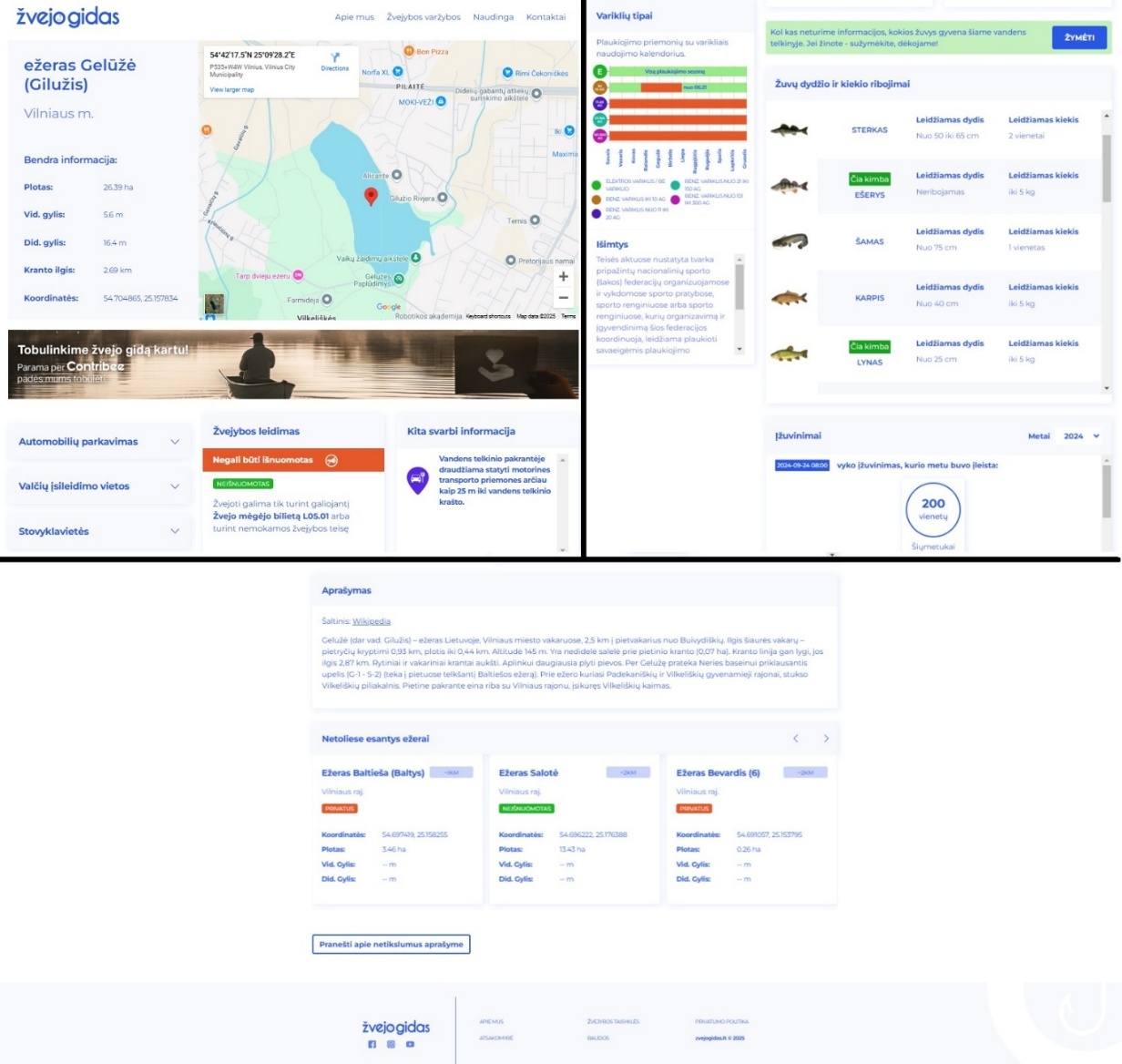
## 1.2. Su žvejyba susijusios programėlės

### 1.2.1. Internetinė svetainė „Žvejo gidas“

Internetinė svetainė „Žvejo gidas“ yra viena iš svarbiausių platformų, skirtų žvejybos ir turizmo entuziastams Lietuvoje. Ši svetainė buvo sukurta siekiant užpildyti informacijos spragą, kai vartotojai, norintys planuoti žvejybos ar poilsio keliones, neturėjo galimybės rasti centralizuotos, patikimos ir išsamios informacijos apie Lietuvos vandens telkinius. Svetainę kūrę žvejybos ir turizmo entuziastai pastebėjo, kad iki šiol Lietuvoje nebuvo vientisos informacinės erdvės, kuri vienytų svarbiausius duomenis apie žvejybos taisykles, vandens telkinių charakteristikas ir susijusią infrastruktūrą. „Žvejo gidas“ suteikia ne tik naudingą informaciją, bet ir prisideda prie gamtos išteklių tausojimo, skatindamas atsakingą požiūrį į žvejybą ir jos taisyklių laikymąsi. Svetainė taip pat tarnauja kaip svarbus informacinis šaltinis tiek patyrusiems žvejams, tiek pradedantiesiems, padėdama užtikrinti, kad žvejai galėtų lengvai pasiruošti ir atitikti visus reikalavimus dar prieš kelionę. Svetainės funkcionalumas apima daugybę svarbių elementų. Galima rasti sportinių žvejybos varžybų sąrašą su detalia informacija, originalias žvejo mėgėjo taisyklių versijas, baudų atmintinę, žvejybinių masalų automatų lokacijų žemėlapį bei visų Lietuvos ežerų interaktyvų žemėlapį. Žemiau pateikiamas „Žvejo gido“ pagrindinis puslapis.

4 pav. „Žvejo gidas“ pagrindinis pulapis

Šaltinis: (Žvejo gidas, b.d.)

Vartotojai, pasirinkę konkretų ežerą, gali peržiūrėti tokią informaciją kaip vandens telkinio plotas, gylis, kranto linijos ilgis, koordinatės, reikalingi žvejybos leidimai, automobilių parkavimo vietos, valčių įleidimo punktai, stovyklavietės, leidžiami variklių tipai ir jų naudojimo sezoniškumas. Taip pat pateikiama informacija apie telkinyje vyraujančias žuvų rūšis, jų leidžiamus dydžius bei kiekius, dirbtinio žuvinimo darbus, trumpas aprašymas iš „Wikipedia“ bei netoliese esantys kiti ežerai. Apačioje pateikiamas vizualus ežero puslapio pavyzdys.

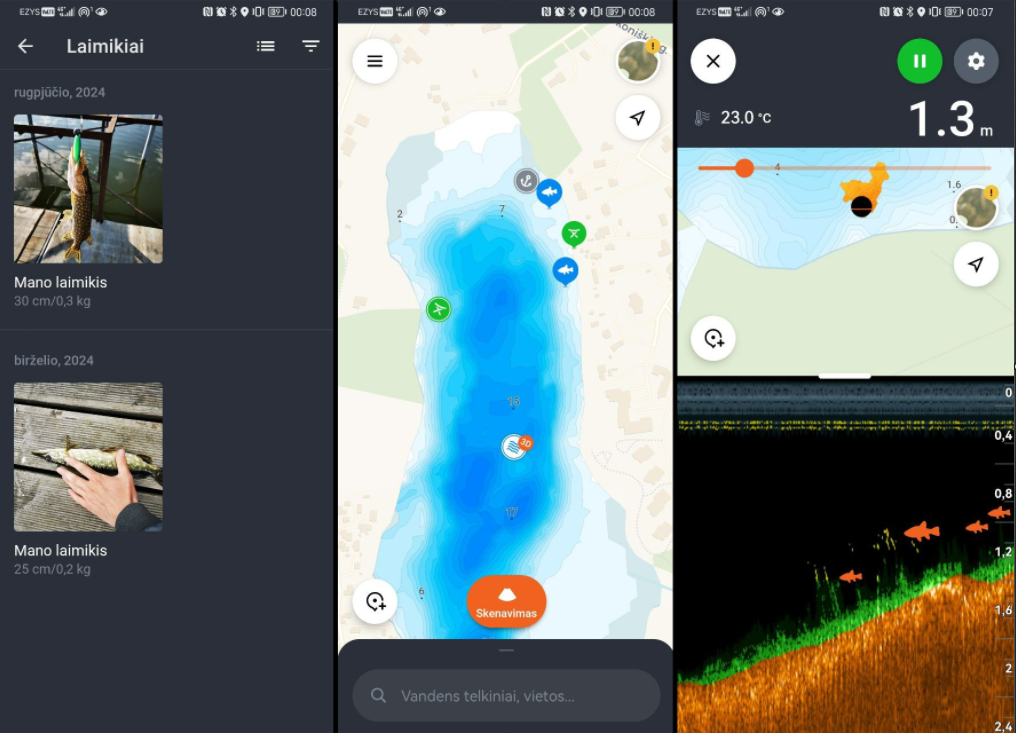
5 pav. „Žvejo gidas“ Gėlužės ežero puslapis

Šaltinis: (Žvejo gidas, b.d.)

Šios funkcijos leidžia vartotojams planuoti keliones bei atsakingai naudoti gamtos išteklius, užtikrinant žvejybos tvarumą ir teisėtumą. (Žvejo gidas, b.d.).

### 1.2.2. Mobilioji aplikacija „Fish Deeper“

Mobilioji programėlė **„**Fish Deeper**“** yra viena populiariausių pasaulyje žvejybai skirtų technologijų, išsiskirianti modernumu ir išmaniųjų technologijų pritaikymu žvejybos praktikoje. Ši aplikacija, sujungta su belaidžiu sonaru, suteikia išsamią informaciją apie vandens telkinio dugno reljefą, žuvų buvimo vietas bei gylį, todėl tampa nepakeičiamu įrankiu tiek mėgėjams, tiek profesionaliems žvejams. Programėlė yra sukurta Lietuvos įmonės „Deeper“, kuri specializuojasi išmaniųjų žvejybos sprendimų kūrime, ir šiuo metu naudojama daugiau nei 50 šalių. Pagrindinė „Fish Deeper“ funkcija – duomenų gavimas iš belaidžio sonaro, kuris siunčia informaciją į išmanųjį telefoną ar planšetinį kompiuterį. Sonaras skenuoja vandens telkinio dugną, nustato jo gylį, reljefą ir parodo žuvų buvimo vietas. Visa informacija realiuoju laiku pateikiama programėlėje, leidžiančioje vartotojui analizuoti aplinką. Be to, sonaras yra kompaktiškas ir pritaikytas naudoti įvairiomis sąlygomis: nuo žvejybos valtyje iki poledinės žūklės. Programėlė taip pat turi integruotą žvejybos žemėlapį, leidžiantį pažymėti svarbias vietas, kuriose buvo sėkmingai sugauta žuvis, bei planuoti būsimas žvejybos išvykas. Naudotojai gali kurti asmeninį žvejybos žurnalą, kuriame įrašomi visi laimikiai, įskaitant jų rūšį, dydį, sugavimo vietą ir datą. Ši funkcija padeda stebėti ir analizuoti sėkmingiausias žvejybos strategijas. Be pagrindinių funkcijų, „Fish Deeper“ programėlėje pateikiama meteorologinė informacija, padedanti numatyti žvejybai palankias sąlygas, tokias kaip vėjo kryptis, oro temperatūra ar atmosferos slėgis. Ji taip pat turi papildomus nustatymus, skirtus ledo žvejybai, kurie leidžia matyti kiekvieno ledo gręžinio vietą ir atskirų žuvų pozicijas po ledu. Viena svarbiausių „Fish Deeper“ išskirtinių savybių yra jos funkcionalumo derinimas su globaliais žemėlapiais. Naudotojai gali naudotis baltimetriniu žemėlapiu, kuris parodo detalų vandens telkinio dugno žemėlapį. Ši funkcija ypač naudinga moksliniams tyrimams, leidžiantiems analizuoti vandens telkinius ir stebėti žuvų populiacijos pokyčius.

Apačioje iliustruojami aprašyti laimikių suvestinės, baltimetriniai žemėlapiai bei vietų žymėjimai ant jų ir pats skanavimo su sonaru procesas.

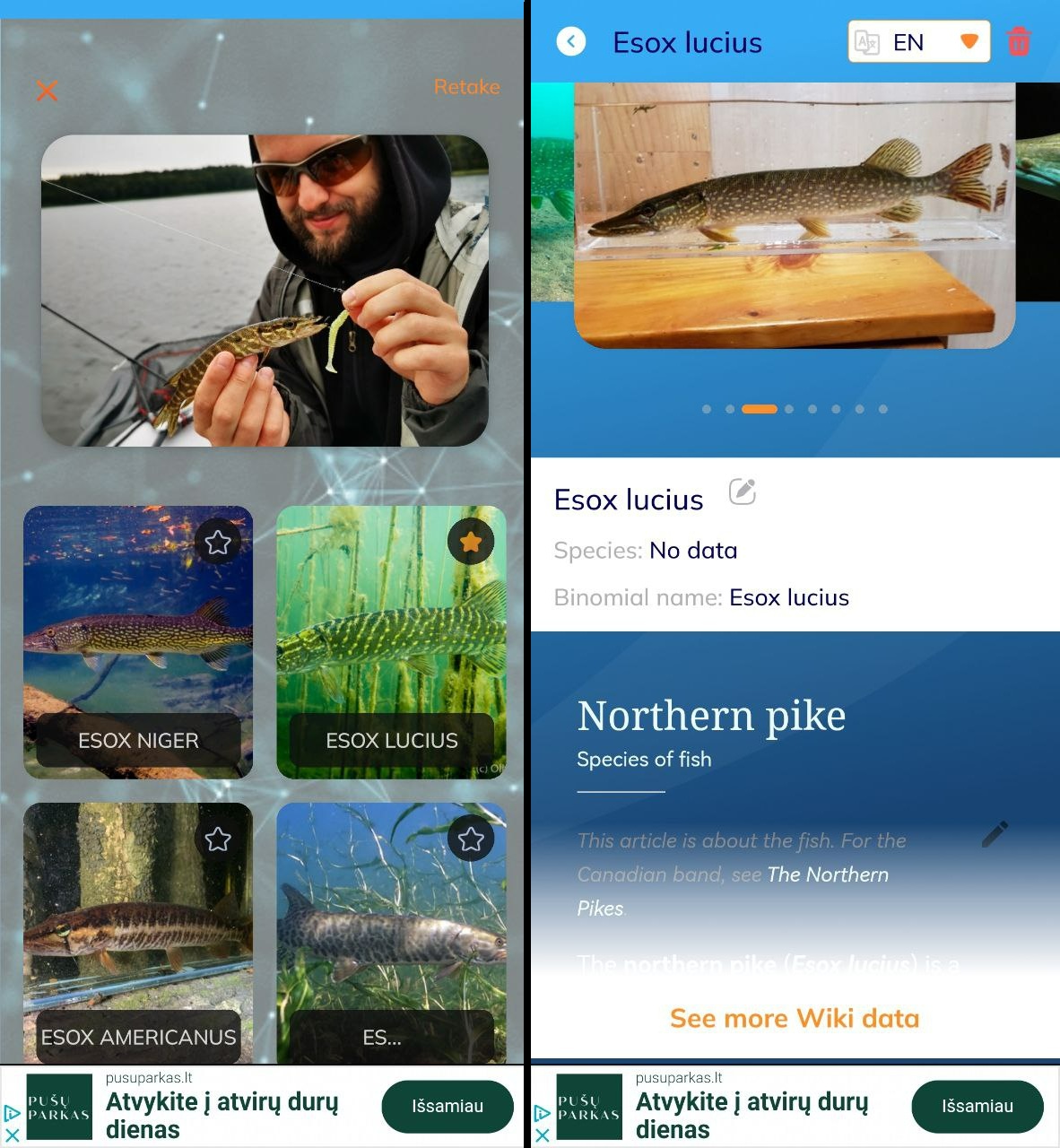
6 pav. „Fish Deeper“ programėlės funkcionalumas

Šaltinis: (Deepersonar, b.d.)

„Fish Deeper“ programėlės ir sonaro sprendimai yra puikus pavyzdys, kaip modernios technologijos gali pagerinti tradicinės žvejybos patirtį. Programėlė suteikia išsamią informaciją apie žvejybos vietas ir sąlygas, leidžia analizuoti bei planuoti žvejybą, taip pat sekti asmeninius rezultatus. Lietuvoje sukurtas produktas yra technologinio inovatyvumo ir praktinės vertės pavyzdys, sulaukiantis vis didesnio pripažinimo tarptautiniu mastu (Deepersonar, b.d.).

### 1.2.3. Mobilioji programėlė „FishAIDetect“

„FishAIDetect“ yra mobilioji programėlė, skirta žuvų rūšims atpažinti pagal nuotraukas. Naudotojai įkelia arba padaro žuvies nuotrauką, o programėlė, naudodama dirbtinio intelekto technologijas, per kelias akimirkas nustato rūšį ir pateikia nuorodą į susijusį „Wikipedia“ straipsnį. Šiame straipsnyje pateikiama pagrindinė informacija apie atpažintą žuvį, įskaitant jos biologines savybes ir kitas aktualias detales. Apačioje pateikiama nuotrauka iliustruojanti šį funkcionalumą.



7 pav. „FishAIDetect“ dirptinis intelektas atpažysta žuvį iš nuotraukos

Šaltinis: (*FishDetect - Fish Identifier – „Google Play“ Programos*, b.d.)

Be žuvų atpažinimo, programėlė turi socialinės bendruomenės funkcionalumą, kuris leidžia naudotojams dalintis savo nuotraukomis, patirtimis ir diskutuoti su kitais žuvų entuziastais. Vartotojai dažnai dalijasi savo akvariumų žuvų nuotraukomis bei diskusijomis apie jas. Apačioje pateikiama šio funkcionalumo iliustracija.

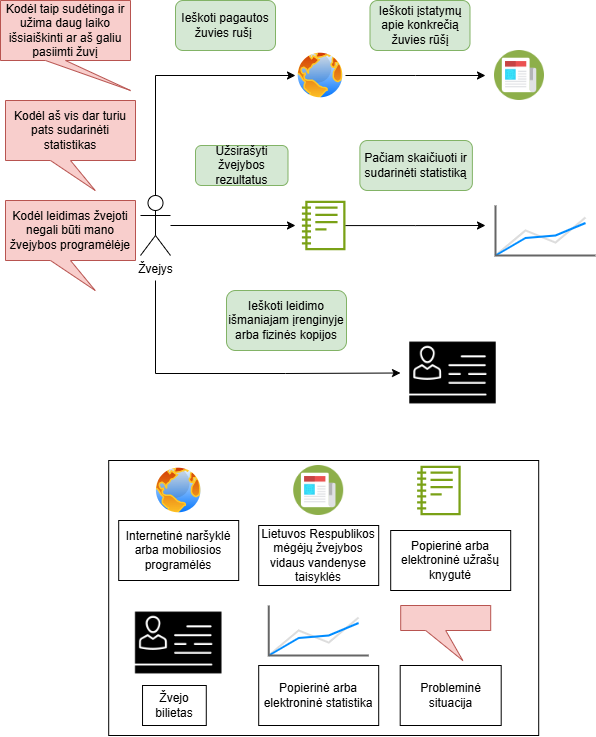


8 pav. „FishAIDetect“ socialinis tinklas

Šaltinis: (*FishDetect - Fish Identifier – „Google Play“ Programos*, b.d.)

„FishAIDetect“ iš esmės yra daugiau pažintinė ir edukacinė programėlė nei įrankis žvejybai. Ji skirta tiems, kurie nori giliau pažinti žuvų pasaulį, dalintis savo patirtimi su bendraminčiais ir mokytis iš kitų. Dėl savo plačių galimybių identifikuoti žuvų rūšis, ji gali būti laikoma naudingu sprendimu tiek akvariumų entuziastams, tiek gamtos mėgėjams (*FishDetect - Fish Identifier – „Google Play“ Programos*, b.d.).

## 1.3. Kuriamos programėlės apžvalga

Žvejo gido programėlė yra skirta išspręsti dalį problemų, su kuriomis kasdien susiduria Lietuvos žvejai. Ši programėlė orientuota į praktinius sprendimus, kurie palengvins žvejų kasdienybę ir leis efektyviau laikytis galiojančių žvejybos taisyklių bei nuostatų. Apačioje pateiktame paveiksliuke yra vizualiai parodyta, kokie iššūkiai kyla Lietuvos žvejams.

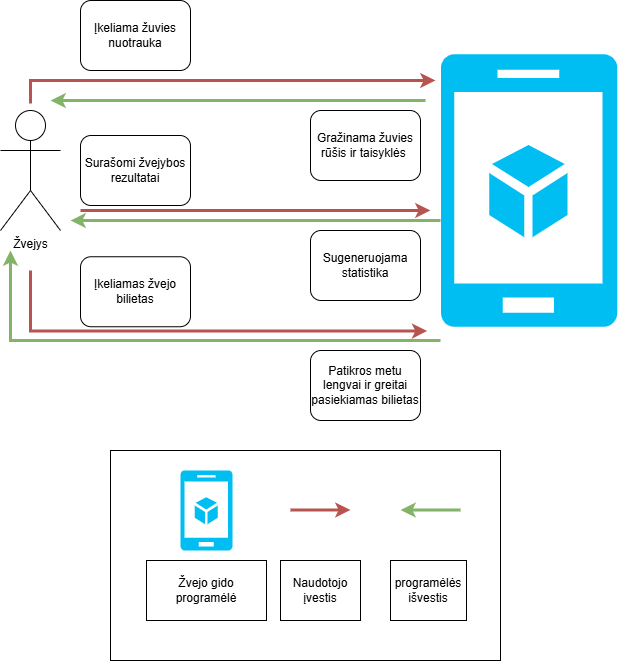
9 pav. Dabartį atspindintis vaizdusis paveiksliukas

Šiame paveiksliuke pateiktos trys pagrindinės problemos, su kuriomis susiduria Lietuvos žvejai.

Pirma problema – žuvų atpažinimas. Pagavęs nežinomą žuvį, žvejys privalo nustatyti jos rūšį, tačiau tai nėra paprasta. Dažniausiai žvejys bando aprašyti žuvies išvaizdą ir, pasinaudodamas interneto paieška, tikisi gauti teisingą rezultatą. Šis metodas reikalauja daug laiko ir gali būti neefektyvus. Alternatyva – naudoti specializuotas žuvų atpažinimo programėles, kurios pagal nuotraukas identifikuoja rūšį. Tačiau šios programėlės dažnai yra pritaikytos kitų šalių faunai, todėl jos gali neteisingai atpažinti žuvį, sumaišyti ją su vandenynų rūšimis arba visai neatpažinti. Net ir nustačius žuvies rūšį, žvejui reikia patikrinti, ar Lietuvos įstatymai leidžia pasiimti šią žuvį. Šiam tikslui naudojamos mėgėjų žvejybos vidaus vandenyse taisyklės, tačiau jų apimtis ir sudėtinga teisinė kalba apsunkina interpretaciją. Be to, laikas tampa kritiniu veiksniu, nes žuvis sausumoje ilgai neišgyvena.

Antra problema – žvejybos duomenų kaupimas ir analizė. Žvejai, siekiantys tobulinti savo įgūdžius, dažnai stengiasi fiksuoti informaciją apie žvejybos laiką, sugautas žuvis ir jų dydžius, taip pat analizuoti, ar tam tikrame vandens telkinyje gyvena didesnės žuvys. Deja, Lietuvoje nėra specializuotų programėlių, kurios būtų pritaikytos tokių duomenų kaupimui ir analizei. Todėl žvejai priversti naudotis sąsiuviniais arba elektroninėmis priemonėmis, o statistiką generuoti savarankiškai. Šis procesas reikalauja daug laiko ir sukelia nepatogumų, nes nėra standartizuotas.

Trečia problema – žvejo bilieto saugojimas. Nusipirkus bilietą internetu, jums atsiunčiamas PDF dokumentas. Šį dokumentą arba dokumento numerį privaloma pateikti kartu su asmens dokumentu patikros metu (Dažniausiai Užduodami Klausimai - Lietuvos Respublikos Aplinkos Ministerija, b.d.). Esminė problema yra tai, kad popierinė kopija gali sušlapti, o atsiųstas PDF dokumentas su laiku gali pasimesti tarp kitų dokumentų. Patogiau būtų, jei žvejo bilietas būtų saugomas nuolat naudojamoje programėlėje.



10 pav. Siekiamos būsenos vaizdusis paveikslėlis

Kaip pavaizduota paveikslėlyje, žvejo gido programėlė siūlo veiksmingus sprendimus pagrindinėms žvejų problemoms spręsti. Žvejys, įkėlęs sugautos žuvies nuotrauką, gautų tikslų žuvies rūšies pavadinimą ir visą aktualią informaciją susijusius su šios žuvies sugavimu, įskaitant leidžiamą dydį ir kitus apribojimus. Tai eliminuoja poreikį ieškoti informacijos internete ar analizuoti sudėtingus teisės aktus. Įkėlus žvejybos rezultatus, sistema automatiškai sugeneruotų detalią statistiką apie laimikius, pavyzdžiui, kurios žuvys dažniausiai kimba tam tikru metų laiku, kokio dydžio laimikiai buvo sugauti, ir kitą svarbią informaciją. Tai leistų žvejams lengviau analizuoti savo veiklos rezultatus ir priimti pagrįstus sprendimus ateityje. Programėlė taip pat leistų patogiai įkelti ir saugoti žvejo bilietą, kurį prireikus būtų galima greitai pasiekti be papildomų pastangų. Tokiu būdu bilietas visuomet būtų po ranka ir paruoštas patikrai. Programėlė supaprastintų žvejų kasdienybę, padėtų sutaupyti laiko, užtikrintų teisės aktų laikymąsi ir pagerintų bendrą žvejybos patirtį.

## 1.4. Žvejybinių sistemų palygynimas

Žemiau pateiktas anksčiau minėtų programinių funkcionalumų palyginimas

1 lentelė Programėlių funkcionalumų palyginimas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nr. | Funkcionalumas | „Žvejo gidas“ | „Deeper“ | „FishAIDetect” | Žvejo gido programėlė |
| 1. | Žvejybos taisyklės | Taip | Ne | Ne | Taip |
| 2. | Specifinės ežerų taisyklės | Taip | Ne | Ne | Taip |
| 3. | Galimybė pažymėti vietas ant žemėlapio | Ne | Taip | Ne | Taip |
| 4. | Baltimetrijos žemėlapis | Ne | Taip | Ne | Ne |
| 5. | Lietuvos ežerų žemėlapis | Taip | Taip | Ne | Taip |
| 6. | Funkcionalumas lietuvių kalba | Taip | Taip | Ne | Taip |
| 7. | Generuoja žvejybų statistikas | Ne | Ne | Ne | Taip |
| 8. | Realiu laiku dugno skenavimas su sonaru | Ne | Taip | Ne | Ne |
| 9. | Žuvų atpažinimas iš nuotraukų | Ne | Ne | Taip | Taip |
| 11. | Įkelti ir gauti pranešimus apie žvejo bilietą | Ne | Ne | Ne | Taip |

Apibendrinant pateiktą lentelę, matyti, kad kiekviena iš nagrinėtų sistemų – „Žvejo gidas“, „Deeper“, „FishAIDetect“ ir žvejo gido programėlė – turi specifinį funkcionalumą, orientuotą į skirtingus naudotojų poreikius, o žvejo gido programėlė centralizuoja daugumą funkcionalumų. „Žvejo gidas“ yra internetinė platforma, orientuota į informaciją apie Lietuvos ežerus ir žvejybos taisykles. Ši svetainė naudinga žvejams, norintiems pasiruošti prieš kelionę į žvejybą – pateikiamos tiek bendros žvejybos taisyklės, tiek specifinės ežerų taisyklės. Platforma siūlo Lietuvos ežerų žemėlapį, o funkcionalumas lietuvių kalba užtikrina patogų naudojimą vietos vartotojams. Nepaisant to, „Žvejo gidas“ nėra mobilioji aplikacija, todėl jai trūksta kai kurių dinamiškų funkcijų, tokių kaip galimybė realiuoju laiku gauti informaciją ar registruoti žvejybos duomenis. „Deeper“ orientuota į technologines inovacijas, skirtas pagerinti žvejybos patirtį. Pagrindinis šios programėlės privalumas yra sonaro technologija, leidžianti gyvai skenuoti vandens telkinių dugną ir gauti batimetrijos žemėlapius. Be to, „Deeper“ suteikia galimybę pažymėti žvejybos vietas ant žemėlapio, o tai ypač naudinga planuojant ateities žvejybos keliones. Tačiau ši programėlė nėra skirta žvejybos taisyklių laikymuisi ar žuvų atpažinimui, todėl jos funkcionalumas yra labiau orientuotas į technologiškai pažangius vartotojus, siekiančius pagerinti savo žvejybos efektyvumą. „FishAIDetect“ specializuojasi žuvų atpažinime pagal nuotraukas, pasitelkdama dirbtinio intelekto technologijas. Ši programėlė suteikia galimybę identifikuoti žuvų rūšis ir gauti apie jas informaciją per „Wikipedia“. Nepaisant savo technologinio pranašumo, „FishAIDetect“ neturi funkcijų, skirtų žvejybos taisyklėms, žemėlapių integracijai ar vietos specifinių reikalavimų laikymuisi. Dėl šios priežasties programėlė yra labiau orientuota į žuvų entuziastus, akvariumų mėgėjus ar tyrėjus, nei į aktyvius žvejus, siekiančius laikytis taisyklių. Žvejo gido programėlė apjungia geriausius kitų programėlių aspektus ir papildo juos specializuotomis funkcijomis. Be bendrų ir specifinių ežerų žvejybos taisyklių pateikimo, ji siūlo galimybę generuoti žvejybų statistikas, identifikuoti žuvis pagal nuotraukas ir valdyti žvejo bilieto informaciją. Programėlė taip pat turi funkcionalumą lietuvių kalba, todėl yra pritaikyta vietos naudotojams. Skirtingai nei „Deeper“, Žvejo gido programėlė neturi sonaro funkcijų, tačiau siūlo galimybę pažymėti žvejybos vietas ir naudotis Lietuvos ežerų žemėlapiu, kas užtikrina išsamų sprendimą žvejams.

Aptartos sistemos pasižymi savo privalumais ir yra specializuotos skirtingose srityse. „Žvejo gidas“ orientuojasi į žvejybos taisykles ir ežerų informaciją, „Deeper“ – į technologinius sprendimus, gerinančius žvejybos patirtį, o „FishAIDetect“ – į žuvų atpažinimą naudojant dirbtinį intelektą. Tuo tarpu Žvejo gido programėlė apjungia daugelį svarbiausių funkcionalumų, taip užtikrindama platesnį paslaugų spektrą žvejams. Dėl šios integracijos Žvejo gido programėlė gali būti laikoma universaliu sprendimu, tinkamu įvairiems žvejų poreikiams.

## Technologijų analizė ir palyginimas

Šiame skyriuje bus analizuojami vaizdų atpažinimo modeliai, skirti žuvų atpažinimui, programavimo kalbos Kotlin, Java ir C#, bei duomenų bazių valdymo sistemos, tokios kaip PostgreSQL, MySQL ir MongoDB. Analizė padės įvertinti šių technologijų tinkamumą, privalumus ir trūkumus, siekiant pasirinkti optimaliausius sprendimus programinės įrangos kūrimui.

## Vaizdų atpažinimo modeliai

Prieš pradedant vaizdų atpažinimo modelių apžvalgą ir palyginimą, derėtų atkreipti dėmesį į keletą svarbių aspektų, kurie yra svarbūs renkantis ir kuriant modelį. Visų pirma, reikalingas didelis kiekis žymėtų vaizdų, naudojamų modelio apmokymui. Šie duomenys turi būti įvairūs, kad atspindėtų visas galimas situacijas, su kuriomis modelis susidurs realiame gyvenime. Surinkti vaizdai dažnai apdorojami – normalizuojami, keičiami jų dydžiai, pašalinamos nereikalingos detalės, kad jie būtų tinkami naudoti neuroniniuose tinkluose (*Admin, 2022)*. Kuriant vaizdų atpažinimo modelius, dažniausiai naudojami giluminiai neuroniniai tinklai, tokie kaip konvoliuciniai neuroniniai tinklai (CNN), kurie yra specialiai sukurti vaizdų apdorojimui. Modelio apmokymas atliekamas naudojant žymėtus duomenis, siekiant išmokyti modelį atpažinti įvairius vaizdo elementus. Šio proceso metu modelis nuolat tikslinamas, siekiant sumažinti klaidų skaičių. Svarbus modelio kūrimo etapas yra testavimas ir validavimas, kurio metu modelis tikrinamas su atskiru duomenų rinkiniu, kuris nebuvo naudojamas mokymui. Taip įvertinamas modelio tikslumas ir gebėjimas generalizuoti. Apmokymo duomenų kokybė ir kiekis yra esminiai veiksniai, lemiantys modelio sėkmę. Aukštos kokybės duomenys tiesiogiai veikia modelio tikslumą, o netinkamai žymėti ar neaiškūs vaizdai gali lemti netikslius rezultatus. Didesnis duomenų kiekis leidžia modeliams geriau generalizuoti ir atpažinti daugiau skirtingų scenarijų, o įvairovė padeda modeliams tapti lankstesniais ir tikslesniais realiame pasaulyje. Be kokybiškų ir įvairių mokymosi duomenų, net geriausi modeliai gali būti neefektyvūs arba netikslūs (Wu ir kt., 2019).

Fishial yra organizacija, siekianti sukurti išsamiausią pasaulyje žymėtų žuvų nuotraukų duomenų bazę ir išvystyti itin tikslų žuvų atpažinimo modelį. Šiuo metu Fishial duomenų bazėje yra sukaupta daugiau nei 850 000 žymėtų žuvų vaizdų, o jų modelis geba atpažinti beveik 300 žuvų rūšių iš skirtingų pasaulio regionų. Siekdama tapti prieinama plačiai auditorijai, įskaitant studentus ir mažas įmones, Fishial platforma yra atvirojo kodo. Tai leidžia vartotojams nemokamai naudotis tiek duomenų rinkiniu, tiek pačiu žuvų atpažinimo modeliu, skatindama mokslinius tyrimus ir praktinius taikymus žuvininkystės bei aplinkosaugos srityse. „FishDetector“ yra algoritmas, skirtas žuvų atpažinimui ir klasifikavimui nevaldomoje aplinkoje. Jis naudoja hibridinį metodą, apimantį Gauss'ų mišinių modelius (GMM), optinį srautą ir giluminius konvoliucinius neuroninius tinklus (CNN), pagrįstus YOLO architektūra. Šis algoritmas teikia pirmenybę YOLO, kai jo ir kitų metodų rezultatai sutampa, užtikrindamas didesnį tikslumą. „FishDetector“ leidžia aptikti ir klasifikuoti žuvis skirtingose aplinkose, todėl jis yra vertingas įrankis ekologiniams ir biologiniams tyrimams. Algoritmas buvo sukurtas naudojant įvairius povandeninių vaizdų rinkinius, siekiant užtikrinti jo veiksmingumą įvairiose situacijose. Jis yra atviro kodo, todėl gali būti naudojamas mokslininkų ir tyrėjų žuvų stebėjimui natūralioje aplinkoje. „Fishial“ ir „FishDetector“ abu yra skirti žuvų atpažinimui, tačiau jie remiasi skirtingais metodais ir turi savitas pranašumo sritis. „Fishial“ modelis koncentruojasi į duomenų bazės plėtrą, siekiant sukurti platų žuvų atpažinimo spektrą, kurį galima lengvai pritaikyti tiek tyrimams, tiek praktiniams taikymams. Tai suteikia galimybę pasiekti didelį žuvų rūšių įvairovės apimtį ir užtikrinti modelio prieinamumą visiems naudotojams. Tuo tarpu „FishDetector“, nors ir naudojantis pažangią hibridinę technologiją su YOLO architektūra, yra labiau orientuotas į žuvų aptikimą natūralioje aplinkoje, kur svarbus yra tikslumas sudėtingomis sąlygomis. Šis modelis yra orientuotas į ekologinius tyrimus, tačiau jo naudojimas gali būti ribotas dėl specifinių aplinkos sąlygų. „Fishial“, be didelio duomenų kiekio ir atvirumo, taip pat apima daugumą Lietuvoje gyvenančių žuvų rūšių, kas yra esminis aspektas mano darbe. Dėl šių priežasčių darbe naudosiu „Fishial“ apmokytą modelį.

## 2.2. Programavimo kalbų apžvalga ir palyginimas

Java yra viena iš plačiausiai naudojamų programavimo kalbų, tame tarpe ir „Android“ programų kūrime. Dėl savo ilgos istorijos ir plačios naudojimo srities Java turi didžiulę bendruomenę, išsamią dokumentaciją ir gausybę bibliotekų, kurios palengvina programų kūrimą. Jos objektinis pobūdis leidžia struktūruoti kodą moduliškai, o tai padeda palaikyti ir plėsti programas. Vienas iš pagrindinių Java privalumų yra jos suderinamumas su senesnėmis „Android“ versijomis, todėl tai yra patikimas pasirinkimas norint užtikrinti programos veikimą įvairiuose įrenginiuose (McKenzie, 2021). Tačiau, dėl savo senesnio dizaino, Java kodas gali būti ilgesnis ir mažiau išraiškingas nei modernesnėse kalbose, o tai gali lemti sudėtingesnį ir ilgiau trunkantį kūrimo procesą.

Kita - Kotlin kalba - yra oficialiai palaikoma „Google“ kalba „Android“ kūrimui, siūlo modernią sintaksę ir didesnį išraiškingumą. Kotlin leidžia rašyti trumpesnį, aiškesnį kodą su mažesniu klaidų kiekiu, nes ji palaiko tokias savybes kaip nulinės vertės saugumas, lambda funkcijos ir išplėtimo funkcijos. Kotlin taip pat yra visiškai suderinama su Java, todėl kūrėjai gali lengvai naudoti esamą Java kodą ir palaipsniui pereiti prie Kotlin, neprarandant ankstesnio darbo. Tačiau, nors jos populiarumas auga, Kotlin bendruomenė ir dokumentacija vis dar yra mažesnės nei Java, o mokymosi kreivė naujiems vartotojams gali būti šiek tiek staigesnė (Bose, 2018).

Paskutinė lyginimų kalbų sąraše yra C#. Tai yra universali ir galinga programavimo kalba, kuri naudojama su „Xamarin“ platforma, leidžiančia kurti programas, kurios veikia tiek „Android“, tiek „iOS“ įrenginiuose. C# pasižymi galingomis funkcijomis, tokiomis kaip asinchroninis programavimas, LINQ užklausos, išreikštos kalbos sintaksė ir aukštas našumas, kuris leidžia efektyviai kurti sudėtingas ir našias programas. Be to, C# integracija su „Microsoft“ ekosistema yra puikus privalumas tiems, kurie dirba su „.NET“ karkasu (Grishina, 2024). Tačiau „Xamarin“ naudojimas gali padidinti pradinį kūrimo laiką dėl papildomų kompilacijos žingsnių ir poreikio optimizuoti programos veikimą skirtingose platformose, kas gali būti sudėtingiau nei naudojant natyvias „Android“ kūrimo priemones.

Java pasirinkimas yra pagrįstas dėl jos stabilumo, plačios dokumentacijos, didelės bendruomenės, geros integracijos su „Android“ SDK ir efektyvaus programų veikimo. Todėl šiame projekte bus naudojama Java, nes ji užtikrins sklandų vystymo procesą ir patikimą galutinį produktą.

## 2.3. Duomenų bazių apžvalga ir palyginimas

MySQL, PostgreSQL ir Microsoft SQL yra trys populiarios duomenų bazių valdymo sistemos, kurios turi savitų savybių ir naudojimo atvejų. Šios sistemos skiriasi pagal struktūrizuotų duomenų apdorojimo būdą, skalabilumą ir lankstumą, todėl kiekviena iš jų turi privalumų ir trūkumų.

MySQL yra viena iš plačiausiai naudojamų reliacinių duomenų bazių valdymo sistemų. Ji remiasi SQL (Structured Query Language) kalba, kuri leidžia efektyviai valdyti duomenis struktūrizuotuose formatuose, pavyzdžiui, lentelėse. MySQL yra greita, lengvai naudojama ir puikiai veikia su įvairaus dydžio duomenų rinkiniais. Jos privalumai apima didelį našumą, gerą integraciją su įvairiomis programavimo kalbomis ir technologijomis, bei plačią bendruomenę, siūlančią paramą ir išteklius. (Erickson, 2024)

Kita duombazė kurią apžvelgsiu yra PostgreSQL. Tai yra dar viena labai populiari reliacinė duomenų bazė, tačiau ji turi tam tikrų pranašumų lygininat su MySQL. PostgreSQL remiasi tuo pačiu SQL standartu, tačiau taip pat palaiko daugybę pažangių savybių, tokių kaip JSON ir JSONB duomenų tipai, kurie leidžia saugoti nestruktūrizuotus duomenis. Be to, PostgreSQL palaiko sudėtingesnes duomenų struktūras ir funkcijas, įskaitant papildomus duomenų tipus, funkcijas, indeksus ir kt. Tai suteikia didesnį lankstumą ir galimybę dirbti su įvairaus tipo duomenimis, kas ypač naudinga didesnėms ir sudėtingesnėms programoms, kurios gali turėti tiek struktūrizuotus, tiek nestruktūrizuotus duomenis. Taip pat PostreSQL daug greičiau susitvarko su dideliais duomenų srautais. (Salunke ir Ouda, 2024). PostgreSQL yra žinoma dėl savo duomenų nuoseklumo ir patikimumo. Ši duombazė palaiko ACID principus (atomarumas, nuoseklumas, izoliacija ir patvarumas), todėl idealiai tinka programoms, kuriose svarbu užtikrinti duomenų tikslumą ir saugumą. (*PostgreSQL: About*, b.d.) Tačiau PostgreSQL, dėl savo saugumo ir aukšo lygio funkcionalumo, gali būti sudėtingesnė naudoti nei MySQL.

Paskutinė duomenų bazių valdymo sistema, kurią apžvelgsiu, yra Microsoft SQL Server (MS SQL) – tai viena iš žinomiausių ir plačiausiai naudojamų komercinių reliacinių duomenų bazių valdymo sistemų, sukurta „Microsoft“ korporacijos. MS SQL Server remiasi SQL kalbos standartu ir siūlo pažangius sprendimus duomenų apdorojimui, ypač įmonių aplinkoje. Ji puikiai tinka tiek OLTP (operaciniams), tiek OLAP (analitiniams) darbo krūviams, todėl dažnai naudojama verslo žvalgybos (BI) sprendimuose, kur reikalingas didelis našumas, duomenų saugumas ir mastelio didinimas.

Viena iš pagrindinių MS SQL savybių – glaudi integracija su Microsoft ekosistema, įskaitant Azure, .NET ir Excel, kas leidžia kurti pažangius sprendimus vienoje aplinkoje. MS SQL Server Management Studio (SSMS) yra galingas įrankis, leidžiantis valdyti serverį, kurti užklausas, atlikti analizę ir administruoti duomenis. Skirtingos versijos, tokios kaip Express, Developer, Standard ar Enterprise, leidžia pasirinkti pagal konkrečius projekto poreikius.

MS SQL Server pasižymi stipriomis saugumo savybėmis – ji palaiko autentifikaciją, duomenų šifravimą, prieigos valdymą ir įvykių audito funkcijas. Taip pat ši releacinė duomenų bazė palaiko ACID principus, taip užtikrindama duomenų vientisumą ir patikimumą. Vienas iš MS SQL trūkumų – licencijavimo modelis, kuris gali būti sudėtingas ir brangus, ypač didesnėms įmonėms. Tačiau išsamios dokumentacijos, aktyvios bendruomenės ir profesionalaus palaikymo dėka MS SQL išlieka populiariu pasirinkimu verslo ir akademinėje aplinkoje. (Ilić et al., 2021)

Kiekviena iš šių duomenų bazių turi savo privalumus ir trūkumus: MySQL yra greita, patikima ir plačiai naudojama sistema, ypač tinkama struktūrizuotiems duomenims valdyti. PostgreSQL siūlo daugiau pažangių funkcijų ir didesnį lankstumą, tačiau dėl savo sudėtingumo gali būti perteklinė mažesnėms ar vidutinio dydžio sistemoms. MS SQL užtikrina glaudžią integraciją su Microsoft aplinka, bet dažnai reikalauja didesnių resursų ir licencijavimo. Šiame darbe pasirinkta naudoti MySQL, nes ši sistema pasižymi paprastu valdymu, aukštu efektyvumu bei lengva integracija su kitomis technologijomis – tai leidžia optimaliai įgyvendinti šios programos poreikius.

## 2.4 Dalinės išvados

Šaltinių analizė atskleidė, jog Lietuvoje būtina reguliuoti žvejų mėgėjų veiklą. Tuo tarpu rinkoje esančių sistemų apžvalga parodė, kad nei viena esamų programų nesuteikia centralizuoto, greito ir patogaus būdo žvejyboje pasitikrinti žvejo mėgėjo taisykles ir kitus reikalingus funkcionalumus. Fishail.ai, Java ir MySQL technologijos leidžia sukurti efektyvią ir lanksčią žvejybos sistemą, kuri suteiks galimybę Lietuvos žvejui greitai ir patogiai pasinaudoti tokiais funkcionalumais, kaip žuvų atpažinimas ir taisyklių pateikimas, žvejybų statistikų generavimas ir žvejo mėgėjo bilieto pasiekimas.

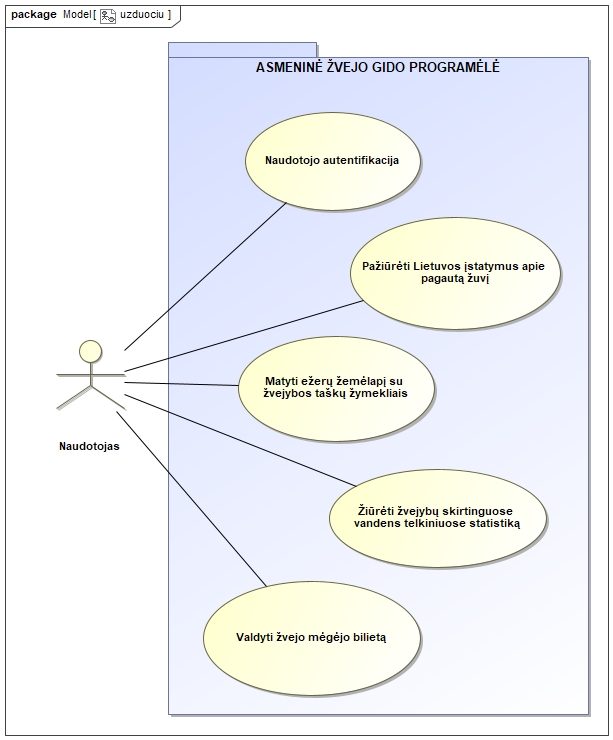
## Sistemos poreikių ir reikalavimų specifikavimas

Šiame skyriuje bus apibrėžti programinės sistemos kūrimui būtini poreikiai ir reikalavimai. Bus išskirti funkciniai reikalavimai, nusakantys, kokią veikseną turi užtikrinti sistema, bei nefunkciniai reikalavimai, apimantys bendruosius sistemos kokybės, veikimo aplinkos ar naudojimo aspektus. Reikalavimų formulavimas padės sukurti nuoseklų projektavimo pagrindą ir užtikrins, kad sistema atitiktų numatytus tikslus bei techninius kriterijus.

## 3.1 Pagrindinės sistemos užduotys

Sistemoje naudotojas identifikuojamas autentifikacijos būdu – prisijungiant prie savo asmeninės paskyros. Didžioji dalis sistemos funkcionalumų yra suasmeninti, t. y. jų veikimas tiesiogiai susijęs su konkrečiu naudotoju ir jo paskyros duomenimis. Prisijungęs naudotojas turi galimybę atverti interaktyvų žemėlapį, kuriame atvaizduojamos anksčiau sukurtos žvejybos vietų žymos, bei sukurti naujas. Kiekviena žyma yra siejama su konkrečia vieta bei gali būti papildyta kita informacija. Be to, naudotojui suteikiama galimybė peržiūrėti žvejybos istorijos statistiką – grafinėje formoje pateikiami duomenys apie anksčiau lankytus žvejybos vandens telkinius, sugautų žuvų skaičių, jų rūšis bei kitus su žvejyba susijusius parametrus. Sistema taip pat leidžia naudotojui įkelti ir administruoti savo mėgėjo žvejo bilietą – jis gali būti saugomas skaitmeniniu formatu, peržiūrimas bei prireikus atnaujinamas. Vienas iš svarbiausių funkcionalumų – galimybė įkelti arba tiesiogiai nufotografuoti sugautą žuvį. Sistema atpažinus žuvies rūšį, naudotojui pateikiama aktuali informacija apie šios žuvies žvejybos reikalavimus, reglamentuojamus Lietuvos Respublikos teisės aktuose – leidžiamą dydį, gaudymo sezoną, limitus ir kt.

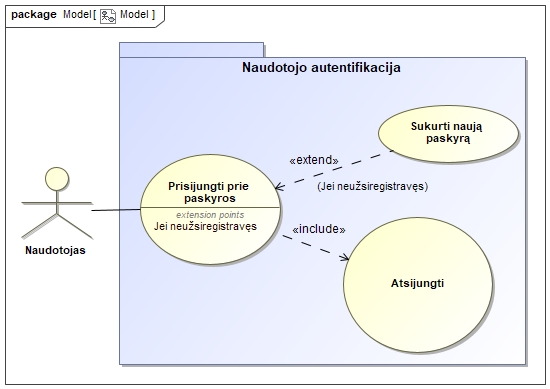
Žemiau pateiktoje veiklų diagramoje (žr. 11 pav.) vizualizuoti pagrindiniai sistemos aktoriai ir jų atliekamos užduotys. Prireikus, diagramos veiklos gali būti detalizuojamos, siekiant tiksliau atvaizduoti sistemos logiką bei jos naudojimo scenarijus.



11 pav. Sistemos užduočių diagrama

Anksčiau pateiktoje užduočių diagramoje (žr. 11 pav.) pavaizduoti pagrindiniai sistemos funkcionalumai ir juos atliekantis naudotojas. Siekiant išsamiau aprašyti sistemos veikimą, visi šie funkcionalumai toliau detalizuojami užduočių scenarijais. Scenarijai pateikiami lentelių forma, kuriose apibrėžiami pagrindiniai aspektai: užduoties pavadinimas, trumpas aprašymas, prieš-sąlygos, pagrindinis scenarijus, alternatyvūs scenarijai, po-sąlygos bei papildoma informacija.

Žemiau pateikiama autentifikacijos užduoties diagrama (žr. 12 Pav.).



12 pav. Autentifikacijos užduotis

Ši diagrama (žr. 12 pav.) vizualizuoja autentifikavimo funkcionalumą. Žemiau pateiktose lentelėse (žr. 2-4 lentelėse) aprašyti pagrindiniai scenarijai.

2 lentelė Prisijungimo scenarijus

|  |  |
| --- | --- |
| Pavadinimas | Prisijungti |
| Trumpas aprašymas | Prisijungimas prie jau sukurtos paskyros |
| Prieš - sąlygos | Turi egzistuoti paskyra prie kurios bus bandoma prisijungti |
| Pagrindinis scenarijus | 1. Atidaroma programėlė 2. Įvedami prisijungimo duomenys 3. Spaudžiamas *Prisijungti* mygtukas |
| Alternatyvūs scenarijai | 1. Įvedami neteisingi duomenys    1. Naudotojas informuojamas, prisijungimo duomenys neteisingi 2. Įvedami prisijungimo duomenys neegzistuojančiai paskyrai    1. Naudotojas informuojamas, prisijungimo duomenys neteisingi |
| Po - sąlygos | Naudotojas sėkmingai prisijungia prie savo paskyros |
| Papildoma informacija | Prisijungimui reikalingi duomenys: prisijungimo vardas ir slaptažodis |

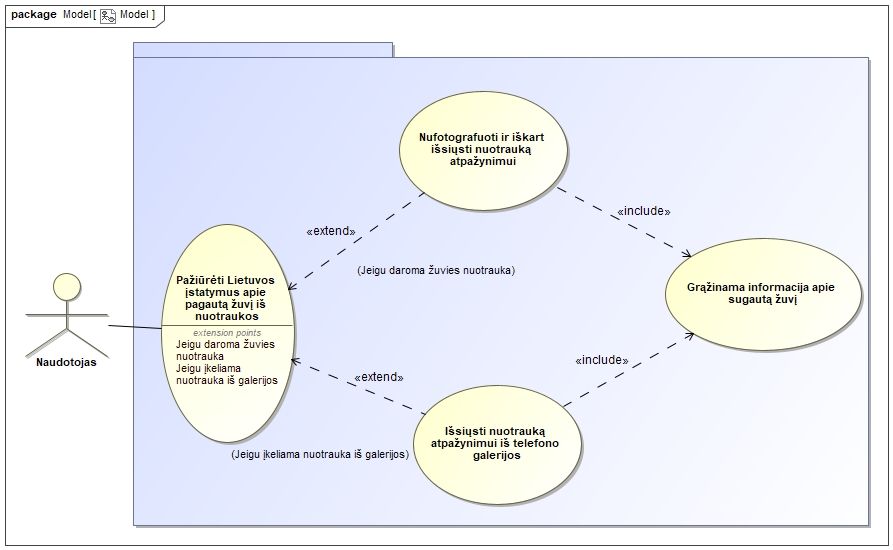
3 lentelė Registracijos scenarijus

|  |  |
| --- | --- |
| Pavadinimas | Registruotis |
| Trumpas aprašymas | Sukuriama nauja paskyra |
| Prieš - sąlygos | Nėra egzistuojančios paskyros su norimu prisijungimo vardu |
| Pagrindinis scenarijus | 1. Atsidaroma programėlė 2. Paspaudžiamas mygtukas *Registruotis* 3. Suvedami duomenys 4. Paspaudžiamas mygtukas *Registruotis* |
| Alternatyvūs scenarijai | 1. Įvedamas jau egzistuojantis prisijungimo vardas    1. Naudotojas informuojamas, kad toks prisijungimo vardas jau egzistuoja 2. Slaptažodis neatitinka pakartotinai įvesto slaptažodžio    1. Naudotojas informuojamas, kad slaptažodžiai nesutampa |
| Po - sąlygos | Naudotojas sėkmingai priregistruojamas ir prijungiamas prie sistemos |
| Papildoma informacija | Registruojantis suvedamas unikalus prisijungimo vardas ir atitinkantis reikalavimus slaptažodis |

4 lentelė Atsijungimo scenarijus

|  |  |
| --- | --- |
| Pavadinimas | Atsijungti |
| Trumpas aprašymas | Atsijungiama nuo naudojamos paskyros |
| Prieš - sąlygos | Naudotojas yra prisijungęs prie paskyros |
| Pagrindinis scenarijus | 1. Atsidaromas pagrindinis puslapis 2. Paspaudžiamas *Atsijungti* mygtukas |
| Po - sąlygos | Naudotojas atjungiamas nuo paskyros ir atsiranda prisijungimo puslapyje |

Toliau pateikiama žuvies atpažinimo ir įstatymų pateikimo užduoties diagrama (žr. 13 pav.).



13 pav. Atpažinimo užduotis

Ši diagrama (žr. 13 pav.) parodo žuvies atpažinimo iš nuotraukos procesą ir svarbiausios informacijos, paremtos Lietuvos reglamentais, grąžinimą. Toliau lentelėse (žr. 5-6 lentelėse) pateikiami šios užduoties scenarijai

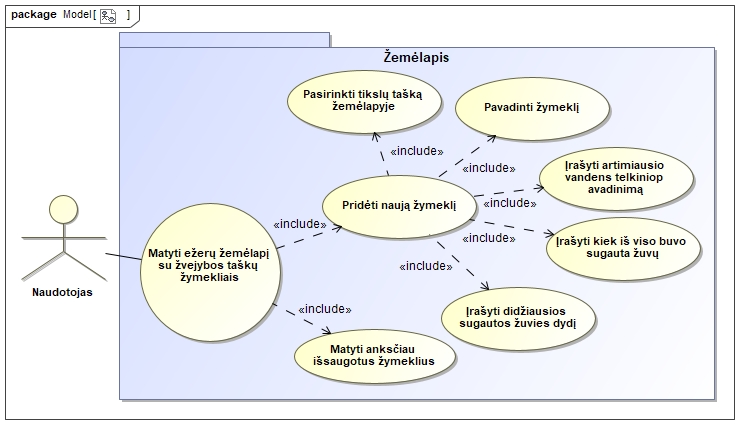
5 lentelė Atpažinimo darant nuotrauką scenarijus

|  |  |
| --- | --- |
| Pavadinimas | Išsiųsti nuotrauką fotografuojant žuvį |
| Trumpas aprašymas | Naudotojas nufotografuoja žuvį, programėlė apdoroja nuotrauką ir grąžina svarbiausią informaciją susijusią su pagauta žuvies rūšimi |
| Prieš - sąlygos | Reikia būti prisijungusiam prie egzistuojančios paskyros ir reikia turėti žuvį kurią galima nufotografuoti |
| Pagrindinis scenarijus | 1. Pagrindiniame ekrane paspaudžiamas mygtukas *Atpažinti žuvį* 2. Nukreipiama kamera į žuvį ir paspaudžiamas mygtukas *Fotografuoti* 3. Stebima programėlės grąžinta informacija apie sugautą žuvį |
| Alternatyvūs scenarijai | 1. Nufotografuojama ne žuvis    1. Programėlė praneša, kad žuvis nebuvo atpažinta 2. Nufotografuojama žuvis kuriai nėra jokių apribojimų    1. Programėlė grąžina bendrus įstatymus visoms Lietuvos žuvims   3. Nufotografuojama žuvis negyvenanti Lietuvos vandenyse arba neįtraukta į duombazę  3.1. Programėlė grąžina tik žuvies pavadinimą |
| Po - sąlygos | Naudotojas mato padarytą žuvies nuotrauką ir žino ar gali ją pasiimti ar ne |
| Papildoma informacija | Nuotrauka išsaugoma į galeriją, tad naudotojui nebereikia papildomai fotografuoti žuvies |

6 lentelė Atpažinimo įkeliant nuotrauką scenarijus

|  |  |
| --- | --- |
| Pavadinimas | Išsiųsti nuotrauką įkeliant iš galerijos |
| Trumpas aprašymas | Naudotojas įkelia žuvies nuotrauką iš galerijos, programėlė apdoroja nuotrauką ir gražina svarbiausią informaciją susijusia su pagauta žuvies rūšimi |
| Prieš - sąlygos | Reikia būti prisijungusiam prie egzistuojančios paskyros ir reikia turėti žuvies nuotrauką galerijoje |
| Pagrindinis scenarijus | 1. Pagrindiniame ekrane paspaudžiamas mygtukas *Atpažinti žuvį* 2. Paspaudžiamas mygtukas *Įkelti iš galerijos* 3. Išsirenkama nuotrauka ir paspaudžiama ant jos 4. Stebima programėlės gražinta informacija apie sugautą žuvį |
| Alternatyvūs scenarijai | 1. Įkeliama ne žuvies nuotrauka    1. Programėlė praneša, kad žuvis nebuvo atpažinta 2. Įkeliama žuvies nuotrauka kuriai nėra jokių apribojimų    1. Programėlė grąžina bendrus įstatymus visoms Lietuvos žuvims   3. Įkeliama negyvenančios Lietuvos vandenyse arba neįtraukta į duombazę žuvies nuotrauka  3.1. Programėlė grąžina tik žuvies pavadinimą |
| Po - sąlygos | Naudotojas mato padarytą žuvies nuotrauką ir žino ar gali ją pasiimt ar ne |
| Papildoma informacija | Nuotrauka gali būti išsaugota ne būtinai galerijoje. Svarbu, kad nuotrauka būtų pasiekiama telefono atmintyje. |

Kita užduoties diagrama pavaizduoja vandens telkinių žemėlapį su išsaugotais žymekliais (žr. 14 Pav.).



14 pav. Žemėlapio užduotis

Šioje diagramoje (žr. 14 pav.) aprašoma žemėlapio užduoties diagrama. Plačiau išnagrinėjami scenarijai toliau pavaizduotose lentelėse (žr. 7-8 lentelėse).

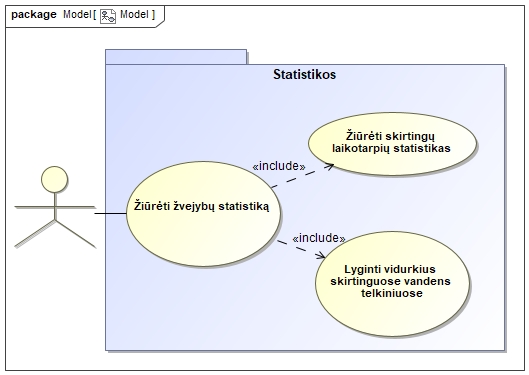
7 lentelė Matyti senus žymeklius scenarijus

|  |  |
| --- | --- |
| Pavadinimas | Matyti anksčiau išsaugotus žymeklius |
| Trumpas aprašymas | Naudotojas atsidaręs žemėlapį matys anksčiau išsaugotus žymeklius |
| Prieš - sąlygos | Naudotojas prisijungęs prie egzistuojančios paskyros ir turi iš anksčiau išsisaugojęs bent vieną žymeklį |
| Pagrindinis scenarijus | 1. Pagrindiniame ekrane paspaudžiamas mygtukas *Žemėlapis* 2. Atsidaro žemėlapis su išsaugotais žymekliais |
| Alternatyvūs scenarijai | 1. Iš anksčiau neišsaugotas nei vienas žymeklis    1. Atsidarys tusčias žemėlapis be žymeklių |
| Po - sąlygos | Naudotojas mato visus iš anksčiau išsaugotus žymeklius |
| Papildoma informacija | Detalesnę žymeklio informaciją galima pamatyti paspaudus ant jo |

8 lentelė Sukurti naują žymeklį scenarijus

|  |  |
| --- | --- |
| Pavadinimas | Sukurti naują žymeklį |
| Trumpas aprašymas | Naudotojas gali sukurti naują žymeklį, kurį galės paskui peržiūrėti |
| Prieš - sąlygos | Naudotojas turi būti prisijungęs prie egzistuojančios paskyros |
| Pagrindinis scenarijus | 1. Pagrindiniame ekrane paspaudžiamas mygtukas *Žemėlapis* 2. Judindamas žemėlapį išsirenka taško koordinates 3. Paspaudžia mygtuką *Pridėti žymeklį* 4. Įrašomas žymeklio pavadinimas 5. Įrašomas vandens telkinio pavadinimas 6. Įrašomas pagautų žuvų kiekis 7. Įrašomas didžiausios pagautos žuvies dydis 8. Paspaudžiamas mygtukas *Išsaugoti* |
| Alternatyvūs scenarijai | 1. Neįvedamas žymeklio arba vandens telkinio pavadinimas    1. Programėlė praneša, kad šie laukai privalomi 2. Neįvedamas pagautų žuvų kiekis arba didžiausios pagautos žuvies dydis    1. Programėlė išsaugo žymeklį be tų parametrų. |
| Po - sąlygos | Sukuriamas naujas žymeklis ir papildomi statistikos duomenys duombazėje |
| Papildoma informacija | Išsaugoti duomenys apie sugautas žuvis ir didžiausią žuvį naudojami sudarant statistikas |

Apačioje pateiktoje diagramoje (žr. 15 pav.) pavaizduojama žvejybų statistikos užduoties diagrama



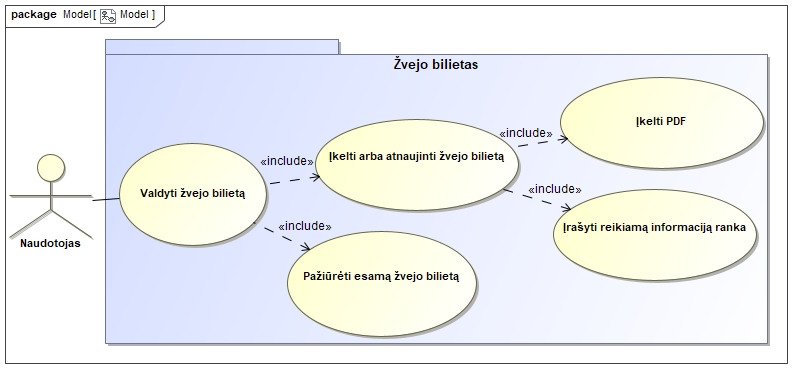
15 pav. Statistiko užduotis

Pateiktoje diagramoje (žr. 15 pav.) atvaizduojamas statistikos užduoties diagrama. Plačiau aprašomas scenarijus lentelėje apačioje (žr. 9 lentelėje).

9 lentelė Statistikos užduotis

|  |  |
| --- | --- |
| Pavadinimas | Statistikos užduotis |
| Trumpas aprašymas | Naudotojas gali pamatyti savo sėkmingų arba nesėkmingų žvejybų statistiką skirtinguose vandens telkiniuose |
| Prieš - sąlygos | Naudotojas turi būti prisijungęs prie egzistuojančios paskyros ir turėti bent vieną įrašą iš praeitų žvejybų |
| Pagrindinis scenarijus | 1. Pagrindiniame ekrane paspaudžiamas mygtukas *Statistika* 2. Paspaudžiama ant norimo laikotarpio |
| Alternatyvūs scenarijai | 1. Nėra įrašų    1. Tuščia lentelė 2. Duombazėje išsaugoti tik vandens telkinių pavadinimai    1. Lentelėje pavaizduoti vandens telkinių pavadinimai, bet duomenų nėra |
| Po - sąlygos |  |
| Papildoma informacija | Statistikos puslapyje galima pridėti įrašų apie žvejybas, tačiau be žymeklio. |

Paskutinė užduoties diagrama (žr. 16 pav.) vaizduoja žvejo mėgėjo bilieto valdymą.



16 pav. Žvejo bilieto užduotis

Ši diagrama (žr. 16 pav.) pavaizduoja žvejo bilieto valdymą. Toliau lentelėse (žr. 10-12 lentelėse) aprašomi žvejo bilieto scenarijai.

10 lentelė Įkelti žvejo bilietą PDF formatu užduotis

|  |  |
| --- | --- |
| Pavadinimas | Įkelti bilietą PDF formatu |
| Trumpas aprašymas | Naudotojas gali įkelti arba atnaujinti savo turimą žvejo mėgėjo bilietą, įkeldamas PDF rinkmeną nusipirktą oficialioje [www.alisas.lt](http://www.alisas.lt) svetainėje. |
| Prieš - sąlygos | Naudotojas turi būti prisijungęs prie egzistuojančios paskyros |
| Pagrindinis scenarijus | 1. Pagrindiniame ekrane paspaudžiamas mygtukas *Žvejo bilietas* 2. Paspaudžiamas mygtukas *Atnaujinti* 3. Paspaudžiamas mygtukas *Įkelti PDF* 4. Pasirenkama norima PDF rinkmena telefono atmintyje |
| Alternatyvūs scenarijai | 1. Įkeliama ne PDF rinkmena    1. Programėlė informuoja, kad pasirinkta netinkama rinkmena 2. Įkeliama netinkamo formato rinkmena    1. Programėlė informuoja, kad duomenų nuskaityti neišėjo |
| Po - sąlygos | Įkeliami/atnaujinami duomenys duombazėje |
| Papildoma informacija |  |

11 lentelė Įrašyti žvejo bilietą ranka užduotis

|  |  |
| --- | --- |
| Pavadinimas | Įrašyti bilieto duomenis rankiniu būdu |
| Trumpas aprašymas | Naudotojas gali atnaujinti arba naujai įrašyti duomenis apie savo žvejo mėgėjo bilietą |
| Prieš - sąlygos | Naudotojas turi būti prisijungęs prie egzistuojančios paskyros |
| Pagrindinis scenarijus | 1. Pagrindiniame ekrane paspaudžiamas mygtukas *Žvejo bilietas* 2. Paspaudžiamas mygtukas *Atnaujinti* 3. Paspaudžiamas mygtukas *Įrašyti* rankiniu būdu 4. Įrašoma bilieto numeris, vardas, pavardė, bilieto galiojimo pabaigos data 5. Spaudžiamas mygtukas *Išsaugoti* |
| Alternatyvūs scenarijai | 1. Įrašomi neteisingo formato arba iš vis neįrašomi duomenys    1. Programėlė informuoja, kad duomenys neteisingi |
| Po - sąlygos | Įkeliami/atnaujinami duomenys duombazėje |
| Papildoma informacija |  |

12 lentelė Pažiūrėti žvejo bilietą užduotis

|  |  |
| --- | --- |
| Pavadinimas | Pažiūrėti žvejo bielietą |
| Trumpas aprašymas | Naudotojas gali pažiūrėti savo įkeltą žvejo bilietą |
| Prieš - sąlygos | Naudotojas turi būti prisijungęs prie egzistuojančios paskyros ir įkėlęs žvejo bilietą į sistemą |
| Pagrindinis scenarijus | 1. Pagrindiniame ekrane paspaudžiamas mygtukas *Žvejo bilietas* |
| Alternatyvūs scenarijai | 1. Nėra įkelto žvejo bilieto    1. Nėra pavaizduotų žvejo bilieto duomenų. |
| Po - sąlygos |  |
| Papildoma informacija | Jeigu naudotojas nėra dar nei karto įkėlęs žvejo bilieto, programėlė atvaizduos tik mygtuką kurio pagalba galima įkelti bilietą |

## 3.2 Funkciniai reikalavimai

Šiame poskyryje pateikiami funkciniai reikalavimai, kurie apibrėžia, kokias funkcijas sistema turi atlikti, kad atitiktų keliamus tikslus ir užtikrintų numatytą naudotojo patirtį. Funkciniai reikalavimai apima konkrečius sistemos veiksmus, galimybes ir sąveikas tarp naudotojo bei sistemos. Šių reikalavimų tikslas – užtikrinti aiškią sistemos elgseną ir nustatyti pagrindines jos sąsajas bei operacijas. Toliau pateiktoje lentelėje ( žr. 13 lentelėje) pateikiami patys funkciniai reikalavimai, bei priskiriami identifikacijos numeriai.

13 lentelė Funkciniai reikalavimai

|  |  |  |
| --- | --- | --- |
| **Nr.** | **ID** | **Aprašymas** |
| 1 | FR-1 | Sistema turi leisti naudotojui prisijungti prie sistemos |
| 2 | FR-2 | Sistema turi leisti naudotojui sukurti naują paskyrą programėlėje |
| 3 | FR-3 | Sistema turi leisti naudotojui saugiai atsijungti nuo savo paskyros |
| 4 | FR-4 | Sistema turi leisti naudotojui įkelti nuotrauką iš telefono galerijos |
| 5 | FR-5 | Sistema turi leisti naudotojui nufotografuoti žuvį |
| 6 | FR-6 | Sistema turi atpažinti žuvį iš įkeltos arba padarytos nuotraukos |
| 7 | FR-7 | Sistema turi atvaizduoti įkeltą arba nufotografuotą nuotrauką |
| 8 | FR-8 | Sistema turi pateikti informaciją apie atpažintą žuvies rūšį, kurios dėka naudotojas suprastu ar gali pasiimti žuvį |
| 9 | FR-9 | Sistema turi pateikti pateikti teisingą informaciją apie žuvį remiantis Lietuvos reglamentais |
| 10 | FR-10 | Naudotojas turi galėti atsidaryti ir matyt žemėlapį |
| 11 | FR-11 | Sistema turi pažymėti visus žymeklius žemėlapyje, kuriuos buvo išsaugojęs naudotojas |
| 12 | FR-12 | Sistema turi leisti naudotojui sukurti naują žymeklį |
| 13 | FR-13 | Sistema turi leisti naudotojui pasirinkti tikslią vietą naujam žymekliui |
| 14 | FR-14 | Sistema turi leisti naudotojui kuriant naują žymeklį įrašyti žymeklio pavadinimą, vandens telkinio pavadinimą, pagautų žuvų kiekį bei didžiausios žuvies dydį |
| 15 | FR-15 | Naudotojas turi galėti atsidaryti savo žvejybų statistiką |
| 16 | FR-16 | Sistema turi atvaizduoti visus vandens telkinius statistikos skiltyje, apie kuriuos naudotojas yra užpildęs |
| 17 | FR-17 | Sistema turi pateikti vidurkį tuose vandens telkiniuose pagautų žuvų skaičių ir didžiausios žuvies vidurkį |
| 18 | FR-18 | Sistema turi leisti naudotojui pasirinkti statistikos laikotarpį |
| 19 | FR-19 | Sistema turi leisti pridėti papildomų duomenų statistikos lange |
| 20 | FR-20 | Naudotojas turi galėti pažiūrėti savo žvejo bilietą jei, jei jis buvo įkeltas |
| 21 | FR-21 | Naudotojas turi galėti įkelti žvejo bilietą PDF formatu |
| 22 | FR-22 | Naudotojas turi galėti surašyti žvejo bilieto informaciją rankiniu būdu |
| 23 | FR-23 | Sistema turi leisti atnaujinti žvejo bilieto informaciją |
| 24 | FR-24 | Sistema turi turėti pagrindinį ekraną kuriuo būtų pasiekiamas pagrindinis funkcionalumas |

## Nefunkciniai reikalavimai

Šiame poskyryje pateikiami nefunkciniai reikalavimai, apibrėžiantys kokybinius sistemos aspektus, kurie nėra tiesiogiai susiję su konkrečių funkcijų įgyvendinimu, tačiau daro esminę įtaką sistemos veikimui, patikimumui bei vartotojo patirčiai. Nefunkciniai reikalavimai gali apimti tokius kriterijus kaip našumas, patikimumas, saugumas, suderinamumas su įvairiomis įrenginių platformomis, naudojimo paprastumas ir priežiūros galimybės. Šių reikalavimų laikymasis užtikrina, kad sistema būtų ne tik funkcionali, bet ir kokybiška bei ilgaamžė. Toliau pateiktoje lentelėje ( žr. 14 lentelėje) pateikiami nefunkciniai reikalavimai, bei priskiriami identifikacijos numeriai.

14 lentelė Nefunkciniai reikalavimai

|  |  |  |
| --- | --- | --- |
| **Nr.** | **ID** | **Aprašymas** |
| 1 | **NFR-1** | Sistema turi būti pasiekiama bet kuriuo paros metu |
| 2 | **NFR-2** | Sistema gali naudotis tik registruoti naudotojai |
| 3 | **NFR-3** | Sistema sukurta „Java“ programavimo kalba |
| 4 | **NFR-4** | Duomenų bazė sukurta naudojant „MySQL“ kalba |
| 5 | **NFR-5** | Sistema turėtų būti intuityvi ir naudotojai turėtų per pirmas 10minučių suprasti kaip ja naudotis |
| 6 | **NFR-6** | Sistema turėtų įvykdyti užklausas ne ilgiau kaip per 2 sekundes esant stabiliam ir greitam internetui išskyrus žuvies atpažinimo užklausas. |
| 7 | **NFR-7** | Sistema turėtų atpažinti ir grąžinti informaciją apie žuvį ne ilgiau kaip per 10 sekundžių |
| 8 | **NFR-8** | Sistemos funkcionalumas turėtų būti pateiktas lietuvių kalba |
| 9 | **NFR-9** | Sistema turi būti pasiekiama 95% metų laikotarpyje. |
| 10 | **NFR-10** | Naudotojo duomenys turi būti saugomi debesyje |
| 11 | **NFR-11** | Duomenų bazė turi daryti atsargines kopijas |
| 12 | **NFR-12** | Sistema turi būti prieinama „Android“ naudotojams |
| 13 | **NFR-13** | Vartotojo sąsajos pranešimai turi būti vienareikšmiški ir aiškūs |
| 14 | **NFR-14** | Sistema turi palaikyti bent 10 000 registruotų naudotojų |
| 15 | **NFR-15** | Neprisijungęs naudotojas gali tik užsiregistruoti arba prisijungti prie sistemos |

## 3.4 Skyriaus išvados

Šiame skyriuje buvo nustatyti pagrindiniai programinės sistemos reikalavimai, padedantys tiksliai apibrėžti, kaip turėtų veikti kuriama sistema. Pateikti funkciniai reikalavimai parodė svarbiausias naudotojo atliekamas veiklas, tokias kaip autentifikacija, žvejybos vietų žymėjimas, žuvų atpažinimas ar statistikos peržiūra. Tuo tarpu nefunkciniai reikalavimai užtikrino, kad sistema veiks patikimai, bus saugi, patogi naudotojui ir pritaikyta skirtingoms įrenginių aplinkoms. Šie reikalavimai sudaro tvirtą pagrindą tolimesniam programinės sistemos projektavimui ir kūrimui.

## Mobiliosios programėlės projektavimas ir realizavimas

Šiame skyriuje bus aprašomi mobiliosios programėlės kūrimo etapai – nuo architektūrinių sprendimų ir vartotojo sąsajos projektavimo iki pagrindinių funkcionalumų įgyvendinimo. Bus pateikti techniniai sprendimai, naudojamos technologijos bei paaiškinimai, kaip buvo įgyvendintas žuvų atpažinimas, duomenų apdorojimas, žemėlapio integracija ir sąveika su duomenų baze. Skyrius atskleis, kaip teoriniai sprendimai buvo pritaikyti praktikoje siekiant sukurti funkcionalią, vartotojui patogią ir techniškai efektyvią programėlę.

## Sistemos architektūra

Šiame skyriuje pateikiamas sukurto sprendimo sistemos architektūros aprašymas. Aptariama, kaip skirtingos sistemos dalys sąveikauja tarpusavyje, siekiant užtikrinti sklandų funkcionalumų veikimą, duomenų apdorojimą ir išorinių paslaugų integraciją. Architektūrinis modelis padeda suprasti pagrindinius komponentus, jų tarpusavio ryšius bei informacijos srautą tarp vartotojo įrenginio, vidinės logikos ir išorinių paslaugų. Toks požiūris leidžia įvertinti sistemos efektyvumą, lankstumą bei galimybes tolesniam plėtimui.

Žemiau pateiktoje architektūros diagramoje vizualiai atvaizduojami pagrindiniai sistemos komponentai ir jų tarpusavio sąryšis (žr. 17 pav.)



17 pav. Architektūros diagrama

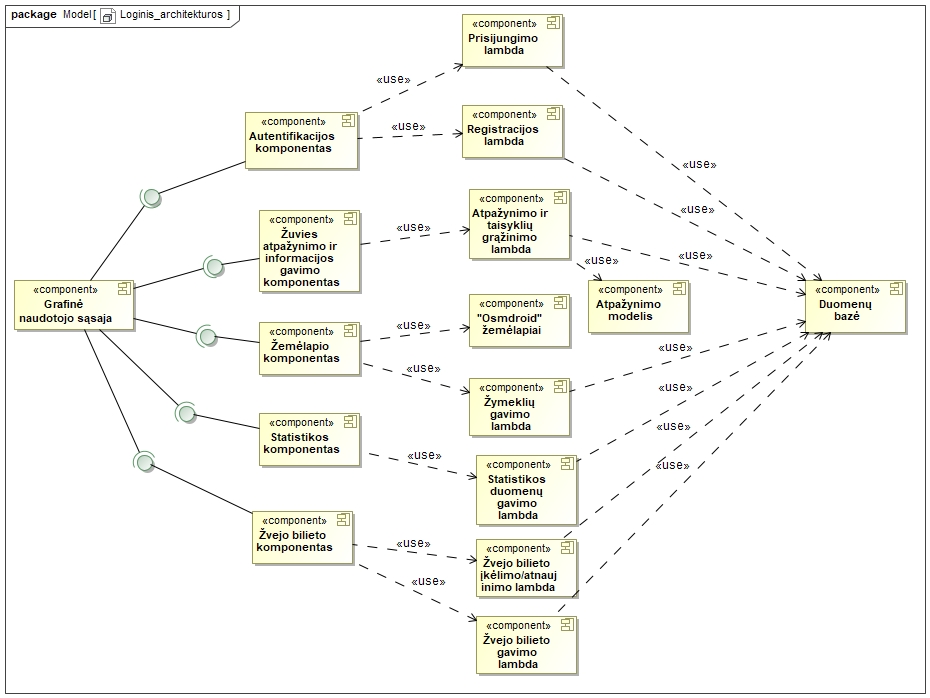
Kuriamai sistemai pasirinktas trijų sluoksnių architektūros modelis, kuris padeda aiškiai atskirti naudotojo sąsają, verslo logiką ir duomenų valdymo komponentus. Toks modelis leidžia ne tik lengviau prižiūrėti bei plėtoti sistemą, bet ir užtikrina jos lankstumą, saugumą bei našumą.

Sistema sudaryta iš šių pagrindinių komponentų:

* Programėlė (kliento pusė). Naudotojas tiesiogiai sąveikauja su mobiliąja programėle, kuri leidžia vykdyti pagrindines funkcijas – fiksuoti žuvies vaizdus, peržiūrėti statistinius duomenis bei naudotis žemėlapių sistema. Dalį funkcijų programėlė įgyvendina lokaliai, tačiau sudėtingesnėms užklausoms, pvz., žuvies atpažinimui ar informacijos iš duomenų bazės gavimui, būtina sąveika su debesijos infrastruktūra.
* AWS Lambda funkcijos (verslo logikos sluoksnis). Kai naudotojo užklausa reikalauja papildomų duomenų arba žuvies atpažinimo, programėlė kreipiasi į AWS debesijos aplinkoje veikiančias Lambda funkcijas. Šios funkcijos atlieka tarpininko vaidmenį – priima užklausą, apdoroja ją pagal logiką ir perduoda tolesniam apdorojimui. Priklausomai nuo užklausos pobūdžio, Lambda gali:
* užmegzti ryšį su duomenų baze (pvz., gauti ar įrašyti žvejybos duomenis).
* kreiptis į trečiųjų šalių API (šiuo atveju – į žuvų atpažinimo modelį „Fishial“).
* Trečiųjų šalių paslauga „Fishial“ (žuvų atpažinimo API). Vaizdo atpažinimo funkcionalumas įgyvendinamas naudojantis išoriniu API „Fishial“, kuris, gavęs žuvies nuotrauką, pateikia rūšies atpažinimo rezultatus. Šis komponentas yra integruotas per AWS Lambda sluoksnį, todėl programėlė tiesiogiai su „Fishial“ API nesąveikauja.
* Duomenų bazė (duomenų valdymo sluoksnis). Sistemos duomenų bazė saugo naudotojų pateiktą informaciją – žuvų atpažinimo rezultatus, naudotojo statistiką, žvejybos lokacijas ir pan. Prieiga prie šios bazės taip pat vyksta tik per Lambda funkcijas, taip užtikrinant saugų duomenų apdorojimą ir sisteminį vientisumą.

Tokios architektūros modelis pasižymi keliais esminiais privalumais. Visų pirma, jis suteikia lankstumo – kiekvienas sistemos komponentas gali būti keičiamas ar atnaujinamas nepaveikiant kitų dalių. Be to, užtikrinamas didesnis saugumas, kadangi programėlė neturi tiesioginės prieigos nei prie duomenų bazės, nei prie išorinių paslaugų. Taip pat svarbus pritaikomumas – naudotojo sąsajos vystymas gali vykti nepriklausomai nuo verslo logikos ir duomenų valdymo sluoksnių. Galiausiai, sistemos architektūra leidžia efektyviai plėsti mastelį: prireikus AWS Lambda funkcijų ar duomenų bazės apkrova gali būti paskirstyta ir automatiškai pritaikoma pagal poreikius.

Plačiau programėlės veikimo principus vizualizuoja loginė diagrama (žr. 18 pav.)

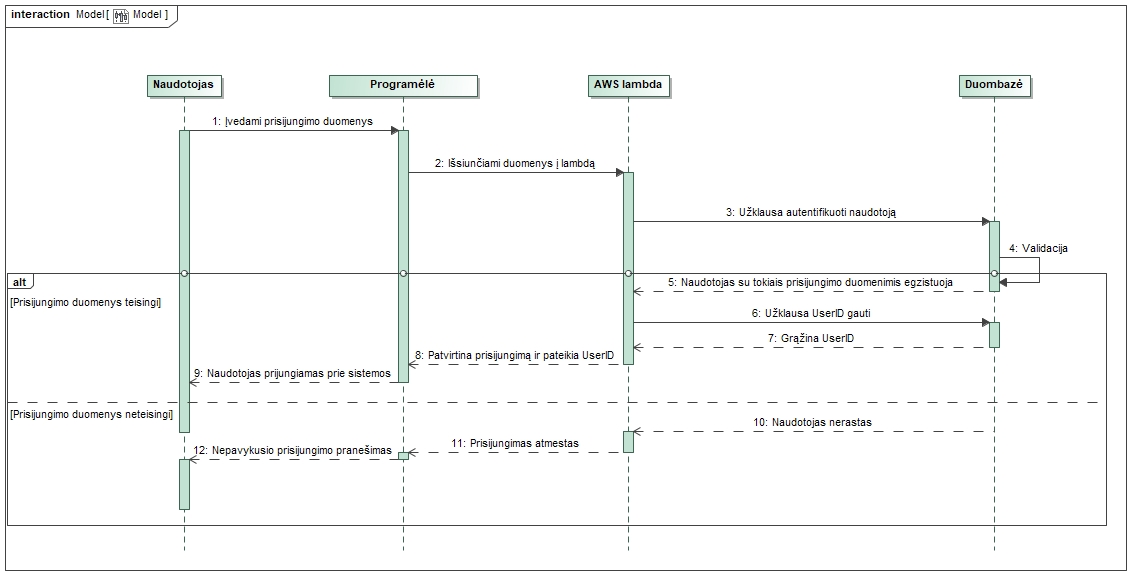


18 pav. Loginė architektūros diagrama

Kaip matyti diagramoje (žr. 18 Pav.), programėlės logika yra aiškiai atskirta nuo duomenų apdorojimo ir saugojimo komponentų. Tokia struktūra užtikrina, kad pagrindinis darbas, susijęs su atpažinimu ir duomenų apdorojimu, būtų atliekamas debesijos infrastruktūroje, taip sumažinant apkrovą įrenginiui ir leidžiant naudotojui gauti atsakymus greitai bei patikimai. Be to, loginė architektūra leidžia lengvai plėsti sistemą integruojant papildomas paslaugas ar keičiant esamus komponentus.

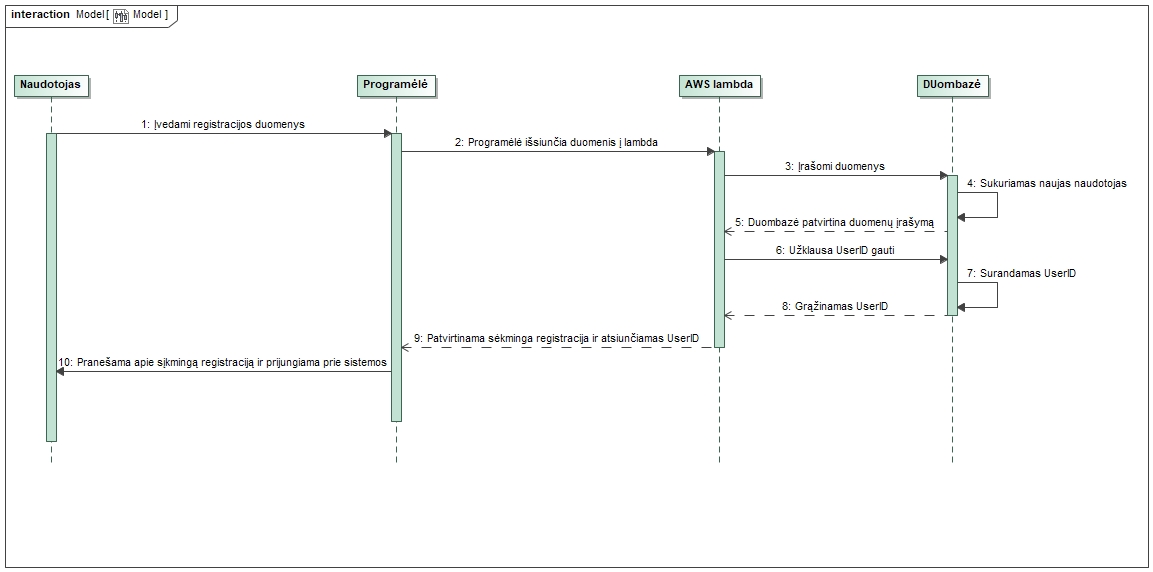
## 4.2 Sekų diagramos

Šiame poskyryje pateikiamos sistemos sekų diagramos, kurios vaizduoja pagrindinių programėlės funkcijų vykdymo eigą bei dalyvaujančių komponentų sąveiką laiko atžvilgiu. Sekų diagramos padeda geriau suprasti, kaip programėlė apdoroja naudotojo užklausas, kaip keičiamasi informacija tarp vidinių sistemų dalių (pvz., AWS Lambda funkcijų, duomenų bazės, atpažinimo modelio) ir kokia tvarka vykdomi veiksmai, siekiant pateikti atsakymą naudotojui.



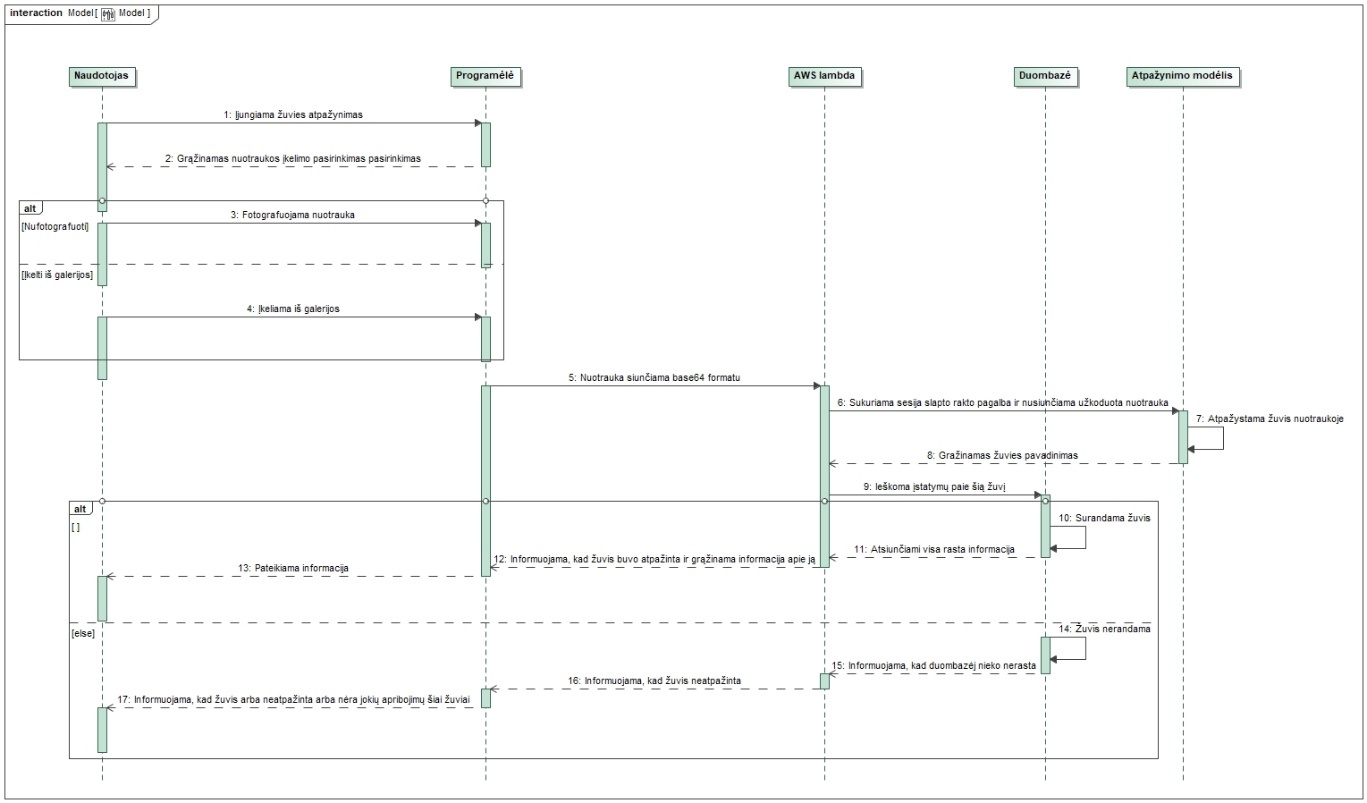
19 pav. Prisijungimo sekų diagrama

Pavaizduotoje sekos diagramoje (žr. 19 pav.) iliustruojama prisijungimo prie programėlės eiga. Naudotojui atidarius programėlę ir įvedus prisijungimo duomenis, šie duomenys yra perduodami į AWS Lambda funkciją. Lambda perduoda užklausą duomenų bazei, siekdama patikrinti naudotojo įvestus duomenis. Jei prisijungimo duomenys yra teisingi, duomenų bazė tai patvirtina ir perduoda naudotojo identifikatorių (UserID). Lambda grąžina programėlei gautą UserID ir patvirtina sėkmingą prisijungimą. Tuomet naudotojui atveriamas pagrindinis programėlės ekranas. Jei duomenų bazė neranda naudotojo su pateiktais prisijungimo duomenimis, ji apie tai informuoja Lambda funkciją, kuri atmeta prisijungimo bandymą. Programėlė, savo ruožtu, pateikia naudotojui pranešimą apie neteisingus prisijungimo duomenis.



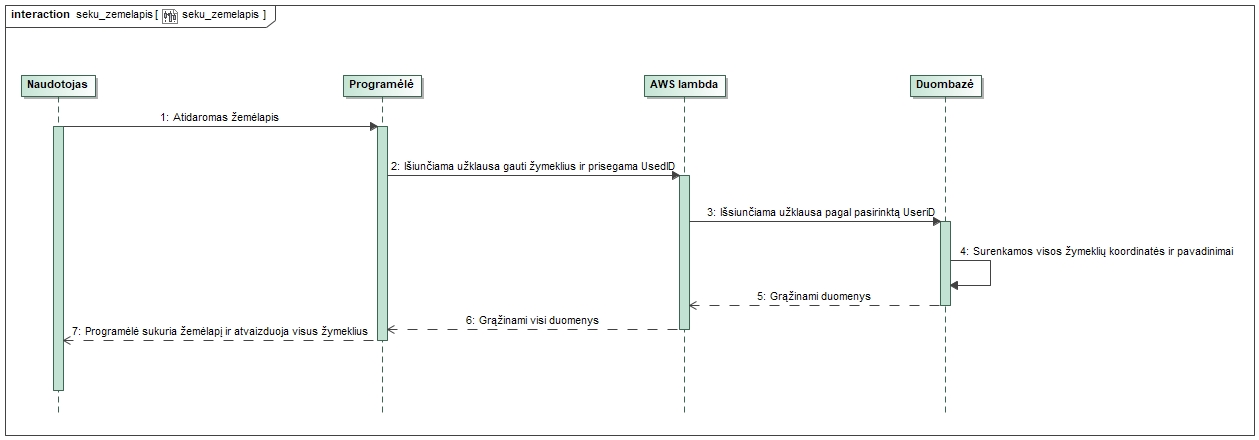
20 pav. Registracijos sekų diagrama

Diagramoje (žr. 20 pav.) pavaizduota naudotojo registracijos seka. Naudotojas programėlėje užpildo registracijos formą, pateikdamas visus reikalingus duomenis. Šie duomenys siunčiami į AWS Lambda funkciją, kuri suformuoja užklausą duomenų bazei dėl naujo naudotojo sukūrimo. Duomenų bazė sukuria naują naudotoją ir patvirtina registracijos sėkmę. Tuomet Lambda funkcija paprašo naudotojo identifikatoriaus (UserID), kurį duomenų bazė suranda ir grąžina. Gavusi UserID, Lambda perduoda jį programėlei kartu su patvirtinimu apie sėkmingą registraciją. Galiausiai programėlė informuoja naudotoją apie sėkmingą registraciją ir prijungia jį prie sistemos.



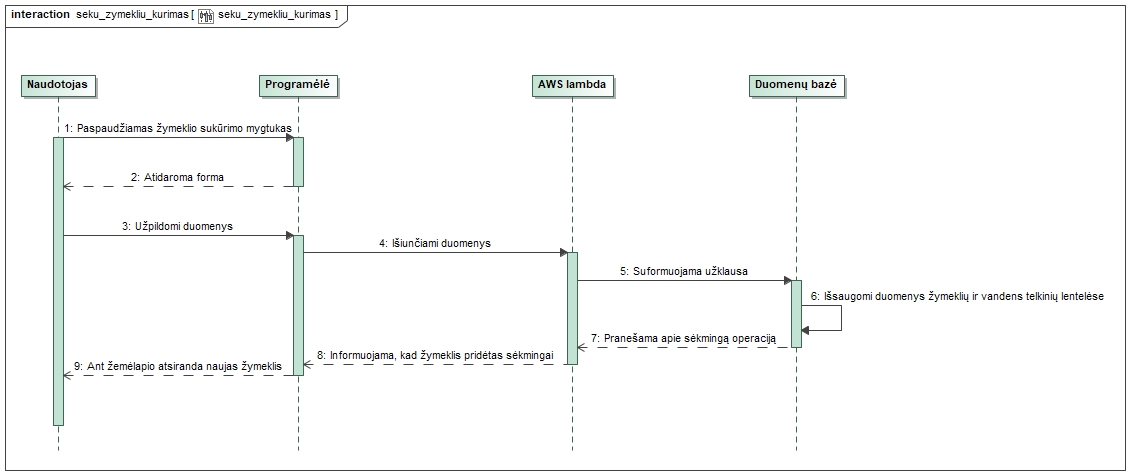
21 pav. Žuvies atpažinimo ir informacijos grąžinimo sekos diagrama

Ši diagrama (žr. 21 pav.) vaizduoja žuvies atpažinimo iš nuotraukos procesą bei svarbiausios informacijos, paremtos Lietuvos reglamentais, grąžinimą naudotojui. Naudotojas paspaudžia žuvies atpažinimo mygtuką, tuomet gali pasirinkti, ar fotografuoti žuvį tiesiogiai, ar įkelti nuotrauką iš galerijos. Gavusi nuotrauką, programėlė ją užkoduoja base64 formatu ir perduoda AWS Lambda funkcijai. Lambda inicijuoja sesiją su „Fishial“ atpažinimo modeliu ir persiunčia užkoduotą vaizdą. Modelis atpažįsta žuvį ir grąžina jos pavadinimą. Tuomet Lambda kreipiasi į duomenų bazę siekdama gauti informaciją apie šią žuvies rūšį. Jei duomenų bazėje tokia informacija egzistuoja, ji grąžinama Lambda funkcijai, kuri perduoda ją programėlei ir informuoja, kad žuvis sėkmingai atpažinta. Programėlė tuomet pateikia naudotojui informaciją aiškiai ir vizualiai. Jei duomenų bazėje tokios žuvies nėra, sistema informuoja Lambda funkciją, jog duomenys nerasti. Lambda praneša programėlei, kad žuvis neatpažinta, o programėlė atvaizduoja žinutę, jog rūšiai nėra taikomų apribojimų arba žuvis neatpažinta.



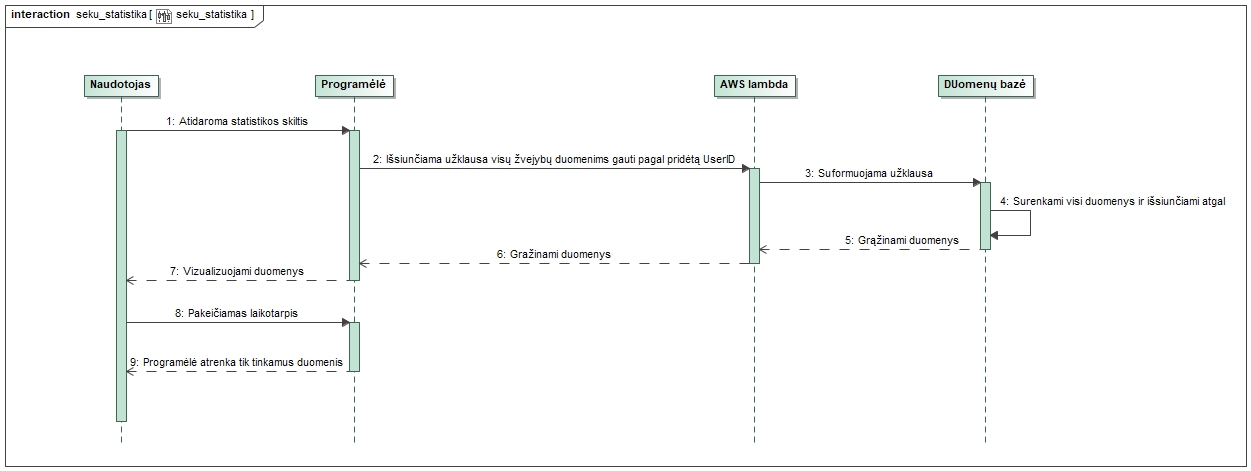
22 pav. Žemėlapio atidarymo sekos diagrama

Šioje diagramoje (žr. 22 pav.) pavaizduota žemėlapio atidarymo ir žymeklių atvaizdavimo seka. Naudotojas pagrindiniame ekrane pasirenka žemėlapio peržiūrą. Programėlė išsiunčia užklausą į AWS Lambda funkciją, pridėdama naudotojo identifikatorių (UserID). Lambda pagal gautą UserID suformuoja užklausą į duomenų bazę. Duomenų bazė surenka visų su naudotoju susijusių žymeklių pavadinimus ir jų koordinates, tuomet grąžina šiuos duomenis Lambda funkcijai. Lambda perduoda gautą informaciją programėlei, kuri sukuria žemėlapio vaizdą ir atvaizduoja visus gautus žymeklius naudotojo ekrane.



23 pav. Žymeklio kūrimo sekos diagrama

Šioje diagramoje (žr. 23 pav.) pavaizduota žymeklio sukūrimo seka. Naudotojas paspaudžia naujos žymos pridėjimo mygtuką, ir programėlė atveria duomenų įvedimo formą. Naudotojui užpildžius formą ir išsaugojus formą, programėlė išsiunčia įvestus duomenis kartu su naudotojo identifikatoriumi (UserID) į AWS Lambda funkciją. Lambda suformuoja užklausą ir išsiunčia ją į duomenų bazę. Duomenų bazė išsaugo žymeklio duomenis ir praneša Lambda funkcijai apie sėkmingą įrašymą. Lambda perduoda programėlei patvirtinimą apie sėkmingą žymeklio sukūrimą. Galiausiai, programėlė atvaizduoja naują žymeklį žemėlapyje.



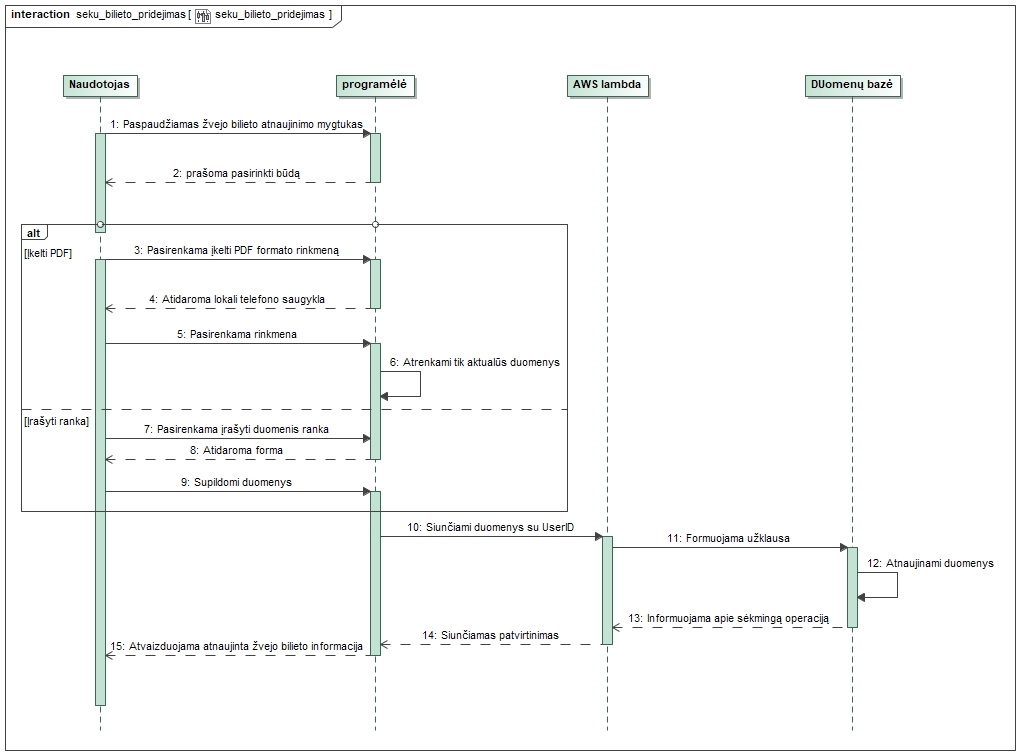
24 pav. Statistikos formavimo sekos diagrama

Šioje diagramoje (žr. 24 pav.) pavaizduojama statistikos formavimo seka. Naudotojui atidarius statistikos skiltį, programėlė išsiunčia užklausą į AWS Lambda funkciją, prisegdama naudotojo identifikatorių (UserID). Lambda suformuoja SQL užklausą ir persiunčia ją duomenų bazei. Duomenų bazė surenka visus reikalingus duomenis ir grąžina juos Lambda funkcijai. Gavusi duomenis, Lambda persiunčia juos programėlei, kuri vizualizuoja statistinę informaciją naudotojui. Jei naudotojas pasirenka kitą laikotarpį, programėlė išrenka tik atitinkamus duomenis ir atnaujina vizualizaciją.



25 pav. Žvejo bilieto atvaizdavimo sekos diagrama

Diagramoje (žr. 25 pav.) pavaizduojama žvejo bilieto atvaizdavimo sekos diagrama. Naudotojas atidaro žvejo bilieto skiltį, o programėlė išsiunčia užklausą į Lambda funkciją kartu su naudotojo identifikatoriumi (UserID). Lambda funkcija patikrina, ar naudotojas turi žvejo bilietą, kreipdamasi į duomenų bazę. Duomenų bazė ieško atitinkamos informacijos. Jeigu bilietas nerandamas, ji informuoja apie tai Lambdą, kuri perduoda šią informaciją programėlei. Programėlė naudotojui pateikia pranešimą, kad žvejo bilietas nėra įkeltas, ir prašo jį įkelti prieš tęsiant. Jei duomenų bazė randa žvejo bilietą, ji apie tai informuoja Lambdą. Tada Lambda pateikia dar vieną užklausą duomenų bazei, kad būtų surinkti visi susiję duomenys. Duombazė grąžina informaciją Lambdai, kuri ją persiunčia programėlei. Galiausiai, programėlė atvaizduoja naudotojui žvejo bilieto duomenis.

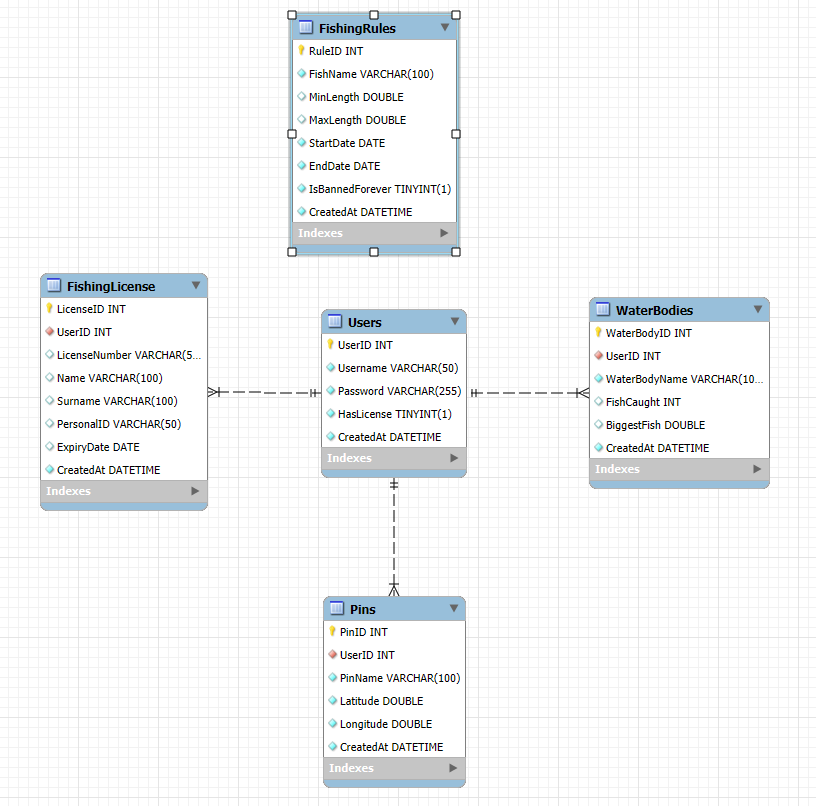


26 pav. Žvejo bilieto atnaujinimo sekos diagrama

Diagramoje (žr. 26 pav.) atvaizduojama žvejo bilieto atnaujinimo sekos diagrama. Naudotojas paspaudžia žvejo bilieto atnaujinimo mygtuką, tuomet programėlė suteikia galimybę pasirinkti vieną iš dviejų būdų: įkelti PDF rinkmeną arba įrašyti duomenis rankiniu būdu. Jei naudotojas pasirenka „Įkelti PDF“, programėlė atidaro telefono vietinę saugyklą, kurioje naudotojas pasirenka norimą dokumentą. Programėlė iš atrinktos PDF rinkmenos išgauna reikiamą informaciją. Jei pasirenkamas rankinis įvedimas, programėlė atidaro formą, kurią naudotojas užpildo. Surinkti duomenys kartu su naudotojo identifikatoriumi (UserID) siunčiami į Lambda funkciją. Lambda suformuoja užklausą ir perduoda ją duomenų bazei. Duomenų bazė atnaujina įrašus ir informuoja apie sėkmingą operaciją. Galiausiai, Lambda funkcija grąžina patvirtinimą, o programėlė atvaizduoja atnaujintus duomenis naudotojui.

## 4.3 Sistemos struktūrinis modelis

Šiame poskyryje pateikiamas programų sistemos struktūrinis modelis, kuris vaizduoja pagrindines sistemos klases bei jų tarpusavio ryšius. Struktūrinis modelis parengtas remiantis duomenų bazės sandara ir atspindi esminius duomenų objektus, jų savybes bei ryšius tarp jų. Klasių diagrama padeda aiškiai suprasti, kaip yra organizuoti duomenys sistemoje, kaip jie sąveikauja tarpusavyje bei kokie loginiai ryšiai egzistuoja tarp skirtingų funkcinių komponentų (žr. 27 pav.). Tai svarbus žingsnis projektuojant ir įgyvendinant sistemą, užtikrinantis aiškią ir nuoseklią architektūrą tiek duomenų saugojimui, tiek jų apdorojimui.



27 pav. Duomenų bazės modulio diagrama

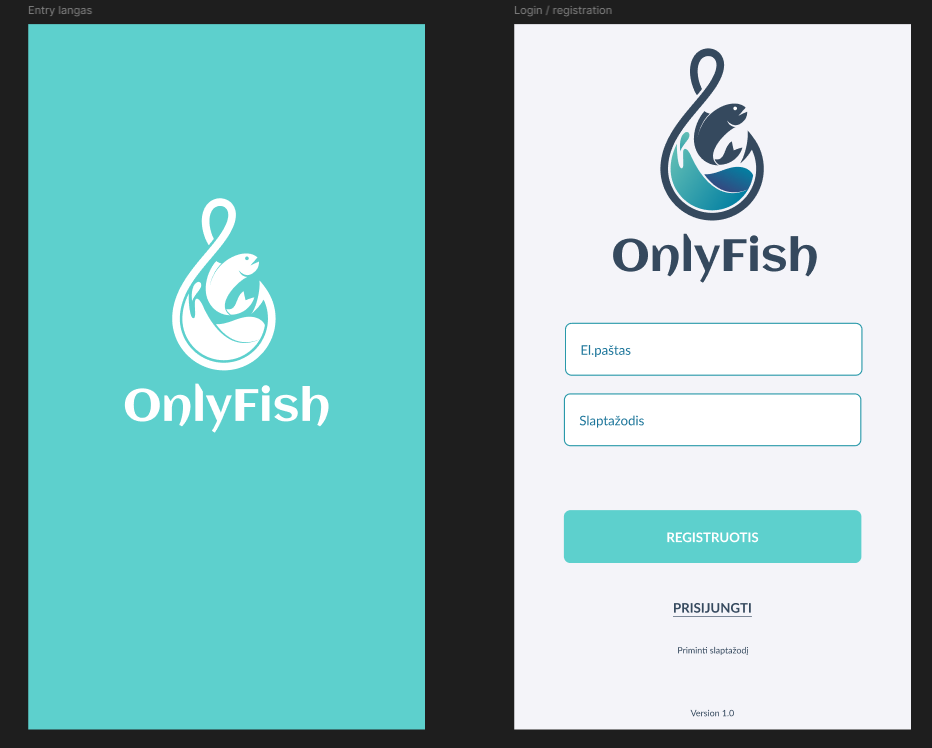
Pagrindinė sistemos dalis yra naudotojų valdymas, kurį apibrėžia klasė „Users“. Šioje klasėje saugomi esminiai naudotojo duomenys, įskaitant unikalų identifikatorių, naudotojo vardą, slaptažodį, informacija apie turimą žvejybos licenciją bei įrašo sukūrimo laikas. Ši klasė yra centrinė – su ja susijusios visos kitos naudotojo specifinės informacijos klasės.

* Klasė „Pins“ saugo naudotojo pažymėtas žvejybos vietas, nurodant vietovės pavadinimą bei geografines koordinates (platumą ir ilgumą). Kiekvienas įrašas yra susietas su konkrečiu naudotoju per išorinį raktą „UserID“, todėl tarp „Users“ ir „Pins“ egzistuoja vienas su daugeliu (1:N) ryšys.
* Klasė „WaterBodies“ aprašo vandens telkinius, kuriuose naudotojas žvejojo. Čia saugoma informacija apie telkinio pavadinimą, pagautų žuvų skaičių, didžiausią žuvį bei naudotoją, kuris užpildė duomenis. Ši klasė taip pat turi vienas su daugeliu ryšį su „Users“ lentele.
* Klasė „FishingLicense“ saugo naudotojams išduotų žvejybos licencijų duomenis, įskaitant licencijos numerį, naudotojo vardą, pavardę bei galiojimo datą. Ryšys su naudotojais realizuotas per „UserID“, o lentelė turi loginę priklausomybę nuo „Users“ ryšiu 1:1.
* Nepriklausoma nuo naudotojų klasė „FishingRules“ apibrėžia bendrąsias sistemos taikomas žvejybos taisykles. Kiekvienoje taisyklėje pateikiama žuvies rūšis, leidžiami ilgiai (minimalus ir maksimalus), draudimo laikotarpis bei vėliava ar iš vis galima imti šią žuvį. Papildomai įtraukta validavimo sąlyga (CHECK), tikrinanti, ar nurodytas laikotarpis yra logiškai pagrįstas.

Apibendrinant galima teigti, kad klasių diagrama efektyviai aprašo sistemos duomenų struktūrą ir jų tarpusavio priklausomybes. Toks modeliavimas leidžia užtikrinti nuoseklų duomenų apdorojimą, palengvina duomenų analizę ir sistemos plėtrą bei padeda pagrįsti loginę programinės sistemos architektūrą.

## Galutinio produkto dizainas

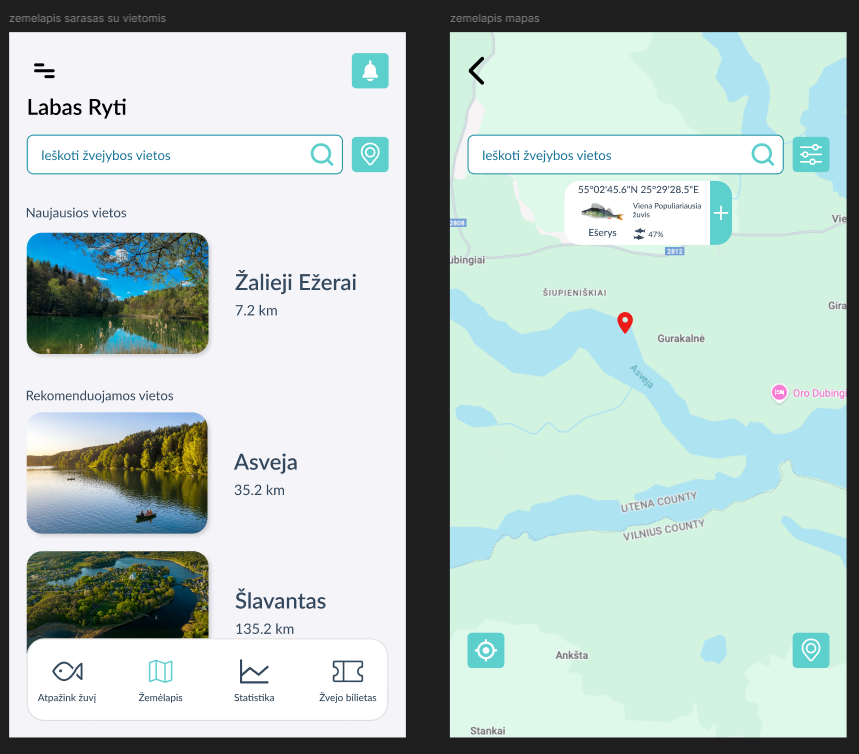
Plėtojant mobiliąją informacinę sistemą, skirta žvejybos veiklai palaikyti, vienu iš esminių programinės įrangos kūrimo etapų tampa vartotojo sąsajos (UI) bei naudotojo patirties (UX) dizaino sprendimų parengimas. Šiame poskyryje pateikiami galutinio produkto vizualinio dizaino prototipai, kurie atspindi esminius sistemos funkcinius komponentus, informacijos išdėstymo logiką bei estetinę išraišką. Sąsajos dizainas parengtas vadovaujantis šiuolaikinėmis UX/UI projektavimo gairėmis, siekiant užtikrinti aukštą naudojimo patogumą, intuityvų funkcijų pasiekiamumą bei vizualinį nuoseklumą.



28 pav. Paleidimo ir prisijungimo ekranų dizainai

Paleidimo ekranas („loading screen“) (žr. 28 pav. kairėje) realizuoja aplikacijos įsikėlimo procesą kartu su logotipu, taip formuojant prekės ženklo atpažįstamumą ir profesionalų įspūdį. Šis vaizdas išlaiko dizaino nuoseklumą ir spalvinę sistemą, identišką kitoms sąsajoms.

Sekančiame prototipe pateiktas autentifikacijos (prisijungimo ir registracijos) langas(žr. 28 pav. dešinėje), kuriame vartotojas gali įvesti prisijungimo duomenis. Vartotojo sąveika čia optimizuota siekiant greito ir saugaus autentifikacijos proceso. Naudojamos formos yra aiškios, pateikiami papildomi mygtukai registracijai bei slaptažodžio atstatymui.



29 pav. Žymeklių ir žemėlapio ekranų dizainas

Pradiniame naudotojo sąsajos vaizde (žr. 29 pav. kairėje) pateikiama personalizuota pasisveikinimo žinutė bei paieškos laukas, skirtas žvejybos vietų identifikavimui pagal pavadinimą ar kitus kriterijus. Informacinis srautas toliau struktūruojamas į rekomenduojamas ir naujausias žvejybos vietas, kurios pateikiamos kortelių formatu su vizualine medžiaga bei papildomais atributais (atstumu iki objekto, pavadinimu, vietos žymėjimu).

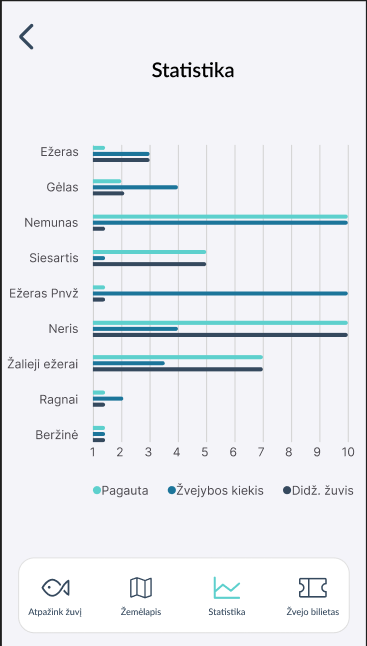
Antrasis vaizdas (žr. 29 pav. dešinėje) orientuotas į žemėlapio integraciją, kur naudotojas gali stebėti pažymėtas žvejybos vietas interaktyvioje erdvėje. Papildomai pateikiama kontekstinė informacija – vietos koordinatės, žuvų rūšys ir jų populiarumo lygis. Šiame kontekste užtikrinamas efektyvus duomenų vizualizavimas geografinėje plokštumoje bei sąveika per filtravimo ar informacijos iškvietimo komponentus.



30 pav. Žvejo bilieto ekrano dizainas

Pateiktame dizaino sprendinyje numatyta naudotojo žvejybos leidimo (bilieto) duomenų peržiūra ir valdymas. Pirmajame ekrane (žr. 30 pav. kairėje) atvaizduojama esama leidimo informacija – vardas, pavardė ir bilieto numeris. Taip pat integruoti du aiškiai išskirti veiksmo mygtukai: „Atnaujinti bilietą“, kuris nukreipia į bilieto atnaujinimo formą, ir „Pridėti PDF“, leidžiantis įkelti leidimo dokumentą PDF formatu. Įkėlus dokumentą, leidimo duomenys automatiškai atsinaujina programėlėje.

Žvejybos leidimo atnaujinimo formoje (žr. 30 pav. dešinėje) naudotojui pateikiami būtini duomenų laukai: vardas, pavardė, licencijos numeris bei leidimo galiojimo data. Forma pasižymi nuoseklia struktūra, užtikrinančia patogų ir greitą duomenų įvedimą. Duomenų išsaugojimui skirtas ryškus, kontrastingas mygtukas „Saugoti“, užtikrinantis intuityvų naudotojo veiksmų vykdymą.



31 pav. Statistikos ekrano dizainas

Paskutinis vaizdas (žr. 31 pav.) skirtas naudotojo žvejybos veiklos statistikos pateikimui per laiko perspektyvą. Čia pasitelkta linijinė diagrama, leidžianti vizualiai identifikuoti žvejybos rezultatų dinamiką. Šis funkcionalumas atlieka informacinės grįžtamosios sąsajos vaidmenį, kuris yra reikšmingas formuojant naudotojo motyvaciją bei leidžia analizuoti asmeninius veiklos modelius.

## Skyriaus išvados

Mobiliosios programėlės projektavimo ir realizavimo etapas parodė, kaip teoriniai principai gali būti praktiškai įgyvendinti, siekiant sukurti technologiškai pažangią ir vartotojui patogią sistemą. Pasirinkta trijų sluoksnių architektūra užtikrino sklandų funkcinių komponentų bendradarbiavimą bei palengvino jų atskyrimą, leidžiant kiekvieną dalį tobulinti atskirai. AWS debesijos paslaugų integravimas leido efektyviai apdoroti sudėtingesnius užklausų scenarijus, o „Fishial“ atpažinimo modelis suteikė galimybę įgyvendinti vieną iš esminių programėlės funkcijų – žuvų atpažinimą iš nuotraukų. Pateiktos loginės ir sekų diagramos aiškiai pavaizdavo, kaip vyksta informacijos srautai tarp sistemos komponentų – nuo naudotojo veiksmų iki duomenų bazės ar išorinių API. Tai padėjo ne tik patikslinti programėlės veikimo principus, bet ir leido įvertinti sprendimų tinkamumą našumo, saugumo ir plėtros galimybių požiūriu. Įgyvendinti sprendimai suformavo tvirtą pagrindą tolesniam funkcionalumo kūrimui ir sistemos evoliucijai.

## 5. Testavimas

## 5.1 Testavimo aprašymas

Šiame poskyryje pateikiama sukurto programinės sistemos prototipo testavimo proceso analizė, apimanti taikytus testavimo principus, eigą ir rezultatus. Testavimas atliktas rankiniu būdu, siekiant įvertinti sistemos atitikimą iškeltiems funkcinės ir nefunkcinės kokybės reikalavimams. Rankinis testavimas pasirinktas dėl galimybės tiesiogiai stebėti sistemos elgseną įvairiose vartotojo sąveikos situacijose, taip užtikrinant išsamesnę kokybinių savybių analizę. Testavimo metu buvo siekiama identifikuoti galimus sistemos veikimo trūkumus, įvertinti jos patikimumą, stabilumą bei naudotojo sąsajos funkcionalumą. Kiekvienam scenarijui buvo sukurti keli testavimo atvejai, apimantys šiuos elementus:

* Testavimo scenarijaus numeris;
* Testavimo scenarijaus santrauka;
* Sąlygos prieš;
* Testavimo scenarijaus eiga (žingsniai);
* Tikėtinas ir gautas rezultatas;
* Scenarijaus išvada (ar veikia tinkamai).

Testų tikslas – įsitikinti, kad programėlė tinkamai apdoroja tiek teisingus, tiek neteisingus naudotojų įvestus duomenis, ir atitinka keliamus funkcinius reikalavimus.

15 lentelė Testų secenarijai ir atvėjai

|  |  |  |  |
| --- | --- | --- | --- |
| Scenarijau Nr. | Testavimo scenarijaus santrauka | Testo Nr. | Testo santrauka |
| TS\_login | Prisijungimas | TA\_login\_1 | Įvedami teisingi duomenys |
| TA\_login\_2 | Įvedami neteisingi duomenys |
| TA\_login\_3 | Neįvedami duomenys |
| TS\_reg | Registracija | TA\_reg\_1 | Įvedami teisingi duomenys |
| TA\_reg\_2 | Įvedamas blogas pakartotinis slaptažodis |
| TA\_reg\_3 | Neįvedami duomenys |
| TS\_recog\_photo | Atpažynimas darant nuotrauką | TA\_recog\_photo\_1 | Fotografuojama žuvis |
| TA\_recog\_photo\_2 | Fotografuojama ne žuvis |
| TS\_recog\_upload | Atpažynimas įkeliant nuotrauką iš galerijos | TA\_recog\_upload\_1 | Įkeliama žuvies nuotrauka |
| TA\_recog\_upload\_2 | Įkeliama ne žuvies nuotrauka |
| TA\_recog\_upload\_3 | Įkeliama ne nuotrauka |
| TS\_map | Žemėlapio ir žymeklių testavimas | TA\_map\_1 | Sukuriamas naujas žymeklis užpildant visus duomenis |
| TA\_map\_2 | Sukuriamas naujas žymeklis užpildant tik būtinus laukus |
| TA\_map\_3 | Sukuriamas naujas žymeklis nepildant duomenų |
| TS\_stat | Statistikos testavimas | TA\_stat\_1 | Atidaroma statistikos skiltis, patikrinami duomenys ir pakeičiamas laikotarpis |
| TS\_lic\_PDF | Atnaujinimas žvejo bilieto įkeliant PDF | TA\_lic\_PDF\_1 | Įkeliamas teisinga PDF rinkmena |
| TA\_lic\_PDF\_2 | Įkeliama ne PDF rinkmena |
| TS\_lic\_manual | Atnaujinimas žvejo bilieto įrašant ranka | TA\_lic\_manual\_1 | Įrašoma teisinga informacija |
| TA\_lic\_manual\_2 | Neįrašomi duomenys |
| TS\_logout | Atsijungimas | TA\_logout\_1 | Atsijungiama |

Aukščiau pateikti (žr. 15 lentelėje) testavimo scenarijai ir atvejai apima pagrindines programėlės funkcijas – nuo naudotojo autentifikacijos ir registracijos iki sudėtingesnių funkcijų, tokių kaip žuvies atpažinimas, žemėlapio naudojimas bei statistikos peržiūra. Kiekvienai funkcijai sukurti testų atvejai, kurie patikrina ir teigiamus, ir neigiamus naudotojo veiksmų scenarijus. Tai leidžia įsitikinti, kad programėlė veikia stabiliai, o naudotojo sąveika su sistema yra saugi ir logiškai apdorojama. Tokia testavimo strategija padeda užtikrinti sistemos kokybę ir atitiktį funkciniams reikalavimams

16 lentelė Prisijungimas teisingi duomenys

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_login\_1 |
| Testo atvejo aprašymas | Įvedami teisingi duomenys |
| „Prieš“ sąlygos | Naudotojas atsidaręs prisijungimo ekraną ir turi egzsistuojančią paskyrą |
| Duomenys | Egzistuojantys prisijungimo duomenys |
| Testo žingsniai | 1. Paspaudiamas naudotojo vardo laukas 2. Įvedamas užsiregistravusio naudotojo vardas  3. Paspaudiamas slaptažodžio laukas 4. Įvedamas slaptažodis  5. Spaudžiamans prisijungimo mygtukas  6. Stebimas ekrano vaizdas |
| Tikimasis rezultatas | Atsidaro pagrindinis ekranas |
| Gautas rezultatas | Atsidaro pagrindinis ekranas |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_login\_1 skirtas patikrinti, ar sistema tinkamai apdoroja korektiškai suvestus prisijungimo duomenis ir leidžia naudotojui pasiekti pagrindinį programėlės ekraną. Testo vykdymo metu sistema reagavo, kaip numatyta – naudotojas buvo sėkmingai autentifikuotas ir nukreiptas į pagrindinį langą, todėl testavimo rezultatas vertinamas kaip teigiamas.

17 lentelė Prisijungimas neteisingi duomenys

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_login\_2 |
| Testo atvejo aprašymas | Įvedami neteisingi duomenys |
| „Prieš“ sąlygos | Naudotojas atsidaręs prisijungimo ekraną |
| Duomenys | Neegzistuojantys prisijungimo duomenys |
| Testo žingsniai | 1. Paspaudiamas naudotojo vardo laukas 2. Įvedamas neužsiregistravusio naudotojo vardas  3. Paspaudiamas slaptažodžio laukas 4. Įvedamas slaptažodis  5. Spaudžiamans prisijungimo mygtukas  6. Stebimas ekrano vaizdas |
| Tikimasis rezultatas | Iššoka pranešimas „prisijungimo duomenys neteisingi“ |
| Gautas rezultatas | Iššoka pranešimas „prisijungimo duomenys neteisingi“ |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_login\_2 skirtas įvertinti, kaip sistema reaguoja į neteisingus prisijungimo duomenis – ar tinkamai identifikuojamas naudotojo autentifikacijos nesėkmės atvejis. Testavimo metu sistema korektiškai atpažino neteisingus duomenis ir pateikė klaidos pranešimą, todėl testas laikomas sėkmingu.

18 lentelė Prisijungimas neįvedami duomenys

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_login\_3 |
| Testo atvejo aprašymas | Neįvedami duomenys |
| „Prieš“ sąlygos | Naudotojas atsidaręs prisijungimo ekraną |
| Duomenys | Duomenų nėra |
| Testo žingsniai | 1. Nieko neįvedant iškart spaudžiamas prisijungimo mygtukas 2. Stebimas ekranas |
| Tikimasis rezultatas | Iššoka pranešimas „Prašome įrašyti prisijungimo duomenis“ |
| Gautas rezultatas | Iššoka pranešimas „Prašome įrašyti prisijungimo duomenis“ |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_login\_3 skirtas patikrinti, ar sistema tinkamai apdoroja situaciją, kai naudotojas bando prisijungti neįvedęs jokių duomenų. Testo metu sistema korektiškai sureagavo – pateikė informacinį pranešimą apie trūkstamus duomenis, todėl testavimo rezultatas vertinamas kaip teigiamas.

19 lentelė Registracija teisingi duomenys

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_reg\_1 |
| Testo atvejo aprašymas | Įvedami teisingi duomenys |
| „Prieš“ sąlygos | Naudotojas atsidaręs registracijos langą |
| Duomenys | Nudoromas dar nepanaudotas naudotojo vardas ir slaptažodis |
| Testo žingsniai | 1. Į naudotojo vardo lauką įrašomas naujas naudotojo vardas 2. Į slaptažodžio lauką įrašomas slaptažodis 3. Pakartotinai suvedamas slaptažodis  4. Spaudžiamas registracijos mygtukas  5. Stebimas ekranas |
| Tikimasis rezultatas | Atidaromas pagrindinis langas |
| Gautas rezultatas | Atidaromas pagrindinis langas |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_reg\_1 skirtas įvertinti, ar sistema leidžia naujam naudotojui sėkmingai užsiregistruoti, kai pateikiami visi reikalingi ir teisingi duomenys. Testavimo metu, įvedus galiojančius registracijos duomenis, naudotojas buvo nukreiptas į pagrindinį programėlės langą, todėl testas laikomas sėkmingu.

20 lentelė Registracija nesutampa slaptažodis

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_reg\_2 |
| Testo atvejo aprašymas | Įvedamas blogas pakartotinis slaptažodis |
| „Prieš“ sąlygos | Naudotojas atsidaręs registracijos langą |
| Duomenys | Nudoromas dar nepanaudotas naudotojo vardas ir slaptažodis |
| Testo žingsniai | 1. Į naudotojo vardo lauką įrašomas naujas naudotojo vardas 2. Į slaptažodžio lauką įrašomas slaptažodis 3. Pakartotinai suvedamas neteisingas slaptažodis  4. Spaudžiamas registracijos mygtukas  5. Stebimas ekranas |
| Tikimasis rezultatas | Iššoka pranešimas „Slaptažodžiai nesutampa“ |
| Gautas rezultatas | Iššoka pranešimas „Slaptažodžiai nesutampa“ |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_reg\_2 skirtas patikrinti, ar sistema tinkamai identifikuoja situaciją, kai pakartotinai įvestas slaptažodis nesutampa su pirminiu. Testavimo metu sistema korektiškai pateikė klaidos pranešimą apie nesutampančius slaptažodžius, todėl testas įvertintas kaip sėkmingas.

21 lentelė Registracija tušti duomenys

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_reg\_3 |
| Testo atvejo aprašymas | Neįvedami duomenys |
| „Prieš“ sąlygos | Naudotojas atsidaręs registracijos langą |
| Duomenys | Nėra duomenų |
| Testo žingsniai | 1. Spaudžiamas registracijos mygtukas neįvedant jokių duomenų  2. Spaudžiamas registracijos mygtukas  3. Stebimas ekranas |
| Tikimasis rezultatas | Iššoka pranešimas „Prašome užpildyti visus laukus“ |
| Gautas rezultatas | Iššoka pranešimas „Prašome užpildyti visus laukus“ |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_reg\_3 skirtas įvertinti sistemos gebėjimą reaguoti į registracijos bandymą neįvedus jokių duomenų. Testavimo metu sistema pateikė atitinkamą pranešimą, raginantį užpildyti visus laukus, todėl testas buvo atliktas sėkmingai.

22 lentelė Atpažynimas fotografuojama žuvis

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_recog\_photo\_1 |
| Testo atvejo aprašymas | Fotografuojama žuvis |
| „Prieš“ sąlygos | Naudotojas prisijungęs ir atsidaręs atpažynimo ekraną |
| Duomenys | Nėra duomenų |
| Testo žingsniai | 1. Naudotojas nukreipia kamerą į žuvį ir spusteli mygtuką „Fotografuoti“ 2. Stebimas ekranas |
| Tikimasis rezultatas | Atsidaro naujas ekranas su žuvies kątik padaryta nuotrauka ir informacija |
| Gautas rezultatas | Atsidaro naujas ekranas su žuvies kątik padaryta nuotrauka ir informacija |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_recog\_photo\_1 tikrina sistemos funkcionalumą fotografuojant žuvį ir automatiškai pateikiant jos atpažinimo informaciją. Vykdant testą, sistema tinkamai užfiksavo vaizdą bei pateikė atitinkamą rezultatą naujame lange, todėl testas įvykdytas sėkmingai.

23 lentelė Atpažynimas fotografuojama ne žuvis

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_recog\_photo\_2 |
| Testo atvejo aprašymas | Fotografuojama ne žuvis |
| „Prieš“ sąlygos | Naudotojas prisijungęs ir atsidaręs atpažynimo ekraną |
| Duomenys | Nėra duomenų |
| Testo žingsniai | 1. Naudotojas nukreipia kamerą į koją ir spusteli mygtuką „Fotografuoti“ 2. Stebimas ekranas |
| Tikimasis rezultatas | Iššoka pranešimas „Žuvis neatpažinta“ |
| Gautas rezultatas | Iššoka pranešimas „Žuvis neatpažinta“ |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_recog\_photo\_2 skirtas patikrinti, kaip sistema elgiasi fotografuojant objektą, kuris nėra žuvis. Testavimo metu sistema teisingai reagavo į netinkamą objektą, pateikdama pranešimą apie neatpažintą žuvį, todėl testas buvo atliktas sėkmingai.

24 lentelė Atpažynimas įkeliama žuvies nuotrauka

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_recog\_upload\_1 |
| Testo atvejo aprašymas | Įkeliama žuvies nuotrauka |
| „Prieš“ sąlygos | Naudotojas prisijungęs, atsidaręs atpažynimo ekraną ir turi galerijoje žuvies nuotrauką |
| Duomenys | Žuvies nuotrauka |
| Testo žingsniai | 1. Paspaudžiamas mygtukas „Galerija“ 2. Išsirenkama žuvies nuotrauka galerijoje ir spustelėjama ant jos 3. Stebimas ekranas |
| Tikimasis rezultatas | Atsidaro naujas ekranas su žuvies įkelta nuotrauka ir informacija |
| Gautas rezultatas | Atsidaro naujas ekranas su žuvies įkelta nuotrauka ir informacija |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_recog\_upload\_1 tikrina sistemos funkcionalumą įkeliant žuvies nuotrauką iš galerijos ir automatiškai pateikiant jos atpažinimo informaciją. Vykdant testą, sistema tinkamai įkėlė nuotrauką ir pateikė atitinkamą rezultatą naujame lange, todėl testas įvykdytas sėkmingai.

25 lentelė Atpažynimas įkeliama ne žuvies nuotrauka

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_recog\_upload\_2 |
| Testo atvejo aprašymas | Įkeliama ne žuvies nuotrauka |
| „Prieš“ sąlygos | Naudotojas prisijungęs, atsidaręs atpažynimo ekraną ir turi galerijoje ne žuvies nuotrauką |
| Duomenys | Ne žuvies nuotrauka |
| Testo žingsniai | 1. Paspaudžiamas mygtukas „Galerija“ 2. Išsirenkama ne žuvies nuotrauka galerijoje ir spustelėjama ant jos 3. Stebimas ekranas |
| Tikimasis rezultatas | Iššoka pranešimas „Žuvis neatpažinta“ |
| Gautas rezultatas | Iššoka pranešimas „Žuvis neatpažinta“ |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_recog\_upload\_2 tikrina sistemos reakciją įkeliant ne žuvies nuotrauką iš galerijos. Vykdant testą, sistema teisingai atpažino, kad nuotrauka nėra žuvis, ir pateikė pranešimą „Žuvis neatpažinta“, todėl testas įvykdytas sėkmingai.

26 lentelė Atpažynimas įkeliama ne nuotrauka

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_recog\_upload\_3 |
| Testo atvejo aprašymas | Įkeliama ne nuotrauka |
| „Prieš“ sąlygos | Naudotojas prisijungęs, atsidaręs atpažynimo ekraną ir turi telefone PDF rinkmeną |
| Duomenys | PDF rinkmena |
| Testo žingsniai | 1. Paspaudžiamas mygtukas „Galerija“ 2. Pasirenkama PDF rinkmena ir spustelėjama ant jos 3. Stebimas ekranas |
| Tikimasis rezultatas | Iššoka pranešimas „Nuotrauka netinkamo formato“ |
| Gautas rezultatas | Iššoka pranešimas „Nuotrauka netinkamo formato“ |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_recog\_upload\_3 tikrina sistemos elgseną įkeliant netinkamo formato rinkmeną, pvz., PDF dokumentą, vietoj nuotraukos. Vykdant testą, sistema teisingai atpažino, kad rinkmena nėra nuotrauka, ir pateikė pranešimą „Nuotrauka netinkamo formato“, todėl testas įvykdytas sėkmingai.

27 lentelė Žemėlapio žymeklis užpildomi visi duomenys

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_map\_1 |
| Testo atvejo aprašymas | Sukuriamas naujas žymeklis užpildant visus duomenis |
| „Prieš“ sąlygos | Naudotojas prisijungęs ir atsidaręs žemėlapio ekraną |
| Duomenys | Žymeklio pavadinimas, ežero pavadinimas ir žvejybos duomenys |
| Testo žingsniai | 1. Užvedamas raudonas taškas virš Galvės ežero 2. Spaudžiamas mygtukas „Pridėti žymeklį“ 3. Įrašomas žymeklio pavadinimas „Gerai kibo karšiai“  4. Įrašomas Galvės ežero pavadinimas  5. Įrašoma pagauta 5 žuvys  6. Įrašoma didžiausia 2.5  7. Spaudžiamas mygtukas „Pridėti“  8. Uždaroma programėlė ir atidaroma iš naujo  9. Atidaromas žemėlapio erkranas  10. Stebimas Galvės ežeras |
| Tikimasis rezultatas | Atsirado naujas žymeklis prie Galvės ežero su pavadinimu „Gerai kibo karšiai“ |
| Gautas rezultatas | Atsirado naujas žymeklis prie Galvės ežero su pavadinimu „Gerai kibo karšiai“ |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_map\_1 tikrina sistemos funkcionalumą kuriant naują žymeklį žemėlapyje, užpildant visus reikalingus duomenis. Vykdant testą, žymeklis su pavadinimu „Gerai kibo karšiai“ buvo sėkmingai pridėtas prie Galvės ežero, ir uždarius bei vėl atidarius programą, žymeklis buvo teisingai atvaizduotas. Todėl testas įvykdytas sėkmingai.

28 lentelė Žemėlapio žymeklis užpildomi būtini duomenys

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_map\_2 |
| Testo atvejo aprašymas | Sukuriamas naujas žymeklis užpildant tik būtinus laukus |
| „Prieš“ sąlygos | Naudotojas prisijungęs ir atsidaręs žemėlapio ekraną |
| Duomenys | Žymeklio pavadinimas, ežero pavadinimas |
| Testo žingsniai | 1. Užvedamas raudonas taškas virš Balto ežero 2. Spaudžiamas mygtukas „Pridėti žymeklį“ 3. Įrašomas žymeklio pavadinimas „Gerai kibo karšiai“  4. Įrašomas Balto ežero pavadinimas  5. Spaudžiamas mygtukas „Pridėti“  6. Uždaroma programėlė ir atidaroma iš naujo  7. Atidaromas žemėlapio erkranas  8. Stebimas Baltas ežeras |
| Tikimasis rezultatas | Atsirado naujas žymeklis prie Balto ežero su pavadinimu „Gerai kibo karšiai“ |
| Gautas rezultatas | Atsirado naujas žymeklis prie Balto ežero su pavadinimu „Gerai kibo karšiai“ |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_map\_2 tikrina sistemos funkcionalumą kuriant naują žymeklį žemėlapyje, užpildant tik būtinus laukus. Vykdant testą, žymeklis su pavadinimu „Gerai kibo karšiai“ buvo sėkmingai pridėtas prie Balto ežero, ir uždarius bei vėl atidarius programą, žymeklis buvo teisingai atvaizduotas. Todėl testas įvykdytas sėkmingai.

29 lentelė Žemėlapio žymeklis nepildomi duomenys

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_map\_3 |
| Testo atvejo aprašymas | Sukuriamas naujas žymeklis nepildant duomenų |
| „Prieš“ sąlygos | Naudotojas prisijungęs ir atsidaręs žemėlapio ekraną |
| Duomenys | Duomenų nėra |
| Testo žingsniai | 1. Užvedamas raudonas taškas virš Salotės ežero 2. Spaudžiamas mygtukas „Pridėti žymeklį“  3. Spaudžiamas mygtukas „Pridėti“ neįrašant duomenų  4. Stebimas ekranas |
| Tikimasis rezultatas | Iššoko pranešimas „Prašome užpildyti būtinus laukus“ |
| Gautas rezultatas | Iššoko pranešimas „Prašome užpildyti būtinus laukus“ |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_map\_3 tikrina sistemos funkcionalumą kuriant naują žymeklį, nepildant reikiamų laukų. Vykdant testą, paspaudus mygtuką „Pridėti“ be jokių įrašytų duomenų, buvo teisingai iššokęs pranešimas „Prašome užpildyti būtinus laukus“. Todėl testas įvykdytas sėkmingai.

30 lentelė Statistika, skirtingi laikotarpiai

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_stat\_1 |
| Testo atvejo aprašymas | Atidaroma statistikos skiltis, patikrinami duomenys ir pakeičiamas laikotarpis |
| „Prieš“ sąlygos | Suvesti duomenys prieš mėnesį, prieš metus ir senesni |
| Duomenys | Ežero pavadinimas, sugautų žuvų kiekis ir dižiausia sugauta žuvis |
| Testo žingsniai | 1. Atidaromas statistikos ekranas 2. Pasirenkamas esamo mėnesio duomenis 3. Stebimas ekranas  4. Pasirenkamas metų laikoptarpio duomenis  5. Stebimas ekranas  6. Pasirenkama visų laikų duomenis  7. Stebimas ekranas  8. Paspaudžiamas mygtukas „Pridėti užrašų“  9. Įrašomas dar neegzsistuojantis ežero pavadinimas  10. Įrašomi likę duomenys  11. Saudžiamas mygtukas „Pridėti“  12. Stebimas ekranas |
| Tikimasis rezultatas | Keičiant laiko rėžes turėtų būti rodomi atitinkami duomenys. Pridėjus naujus duomenis jie turėto atsirasti statistikoje |
| Gautas rezultatas | Keičiant laiko rėžes buvo rodomi atitinkami duomenys. Pridėjus naujus duomenis jie atsirado statistikoje |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_stat\_1 tikrina sistemos funkcionalumą atidarant statistikos skiltį, keičiant laikotarpį ir pridedant naujus duomenis. Vykdant testą, buvo teisingai rodoma atitinkama statistika keičiant laikotarpį, o pridėjus naujus duomenis, jie sėkmingai atsirado statistikoje., todėl testo rezultatas sėkmingas.

31 lentelė Žvejo bilietas teisingas PDF

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_lic\_PDF\_1 |
| Testo atvejo aprašymas | Įkeliamas teisinga PDF rinkmena |
| „Prieš“ sąlygos | Naudotojas prisijungęs ir atsidaręs žvejo bilieto ekraną |
| Duomenys | PDF licenzijos rinkmena |
| Testo žingsniai | 1. Spaudžiamas mygtukas „Atnaujinti bilietą“ 2. Spaudžiamas mygtukas „Įkelti PDF“ 3. Paspaudžiama ant tinkamos PDF rinkmenos  4. Stebimas bilieto ekranas |
| Tikimasis rezultatas | Bilieto atvaizduojami duomenys buvo atnaujinti |
| Gautas rezultatas | Bilieto atvaizduojami duomenys buvo atnaujinti |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_lic\_PDF\_1 tikrina sistemos funkcionalumą įkeliant teisingą PDF rinkmeną į žvejo bilieto ekraną. Vykdant testą, sistema sėkmingai atnaujino bilieto duomenis po PDF rinkmenos įkėlimo. Todėl testas įvykdytas sėkmingai.

32 lentelė Žvejo bilietas ne PDF

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_lic\_PDF\_2 |
| Testo atvejo aprašymas | Įkeliama ne PDF rinkmena |
| „Prieš“ sąlygos | Naudotojas prisijungęs ir atsidaręs žvejo bilieto ekraną |
| Duomenys | Ne PDF formato rinkmena |
| Testo žingsniai | 1. Spaudžiamas mygtukas „Atnaujinti bilietą“ 2. Spaudžiamas mygtukas „Įkelti PDF“ 3. Paspaudžiama ant ne PDF formato rinkmenos  4. Stebimas bilieto ekranas |
| Tikimasis rezultatas | Iššoka pranešimas „Bilieto rinkmena netinkama“ |
| Gautas rezultatas | Iššoka pranešimas „Bilieto rinkmena netinkama“ |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_lic\_PDF\_2 tikrina sistemos reakciją įkeliant netinkamą (ne PDF) rinkmeną žvejo bilieto ekranui. Vykdant testą, sistema teisingai išmetė pranešimą „Bilieto rinkmena netinkama“, todėl testas įvykdytas sėkmingai.

33 lentelė Žvejo bilietas ranka įrašoma teisinga informacija

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_lic\_manual\_1 |
| Testo atvejo aprašymas | Įrašoma teisinga informacija |
| „Prieš“ sąlygos | Naudotojas prisijungęs ir atsidaręs žvejo bilieto ekraną |
| Duomenys | Žvejo bilieto duomenys |
| Testo žingsniai | 1. Spaudžiamas mygtukas „Atnaujinti bilietą“ 2. Spaudžiamas mygtukas „Įrašyti rankiniu būdu“ 3. Įrašomas bilieto numeris, vardas, pavardė ir bilieto galiojimo pabaiga  4. Spaudžiame išsaugojimo mygtuką  5. Stebimas bilieto ekranas |
| Tikimasis rezultatas | Bilieto atvaizduojami duomenys buvo atnaujinti |
| Gautas rezultatas | Bilieto atvaizduojami duomenys buvo atnaujinti |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_lic\_manual\_1 tikrina sistemos funkcionalumą, kai teisingi duomenys įrašomi rankiniu būdu žvejo bilieto ekrane. Vykdant testą, sistema teisingai atnaujino bilieto duomenis, todėl testas įvykdytas sėkmingai.

34 lentelė Žvejo bilietas neįrašomi duomenys

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_lic\_manual\_2 |
| Testo atvejo aprašymas | Neįrašomi duomenys |
| „Prieš“ sąlygos | Naudotojas prisijungęs ir atsidaręs žvejo bilieto ekraną |
| Duomenys | Duomenų nėra |
| Testo žingsniai | 1. Spaudžiamas mygtukas „Atnaujinti bilietą“ 2. Spaudžiamas mygtukas „Rankiniu būdu“ 3. Spaudžiame išsaugojimo mygtuką  4. Stebimas bilieto ekranas |
| Tikimasis rezultatas | Iššoka pranešimas „Prašome užpildyti visus laukus“ |
| Gautas rezultatas | Iššoka pranešimas „Prašome įrašyti būtinus laukus“ |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_lic\_manual\_2 tikrina sistemos reakciją, kai neįrašomi duomenys žvejo bilieto ekrane. Vykdant testą, sistema teisingai išmetė pranešimą „Prašome įrašyti būtinus laukus“, todėl testas įvykdytas sėkmingai.

35 lentelė Atsijungimas

|  |  |
| --- | --- |
| Testo atvejo Nr. | TA\_logout\_1 |
| Testo atvejo aprašymas | Atsijungiama |
| „Prieš“ sąlygos | Naudotojas prisijungęs |
| Duomenys | Nėra |
| Testo žingsniai | 1. Pagrindineme erkane spaudžiamas mygtukas „Atsijungti“ 2. Stebimas ekranas |
| Tikimasis rezultatas | Atsidaro prisijungimo ekranas |
| Gautas rezultatas | Atsidaro prisijungimo ekranas |
| Testo rezultatas | Testas sėkmingas |

Testavimo atvejis TA\_logout\_1 tikrina sistemos funkcionalumą atsijungiant. Vykdant testą, sistema teisingai atidarė prisijungimo ekraną po mygtuko „Atsijungti“ paspaudimo, todėl testas įvykdytas sėkmingai.

## Skyriaus išvados

Testavimas buvo atliktas rankiniu būdu, siekiant įvertinti sistemos atitikimą funkcinės ir nefunkcinės kokybės reikalavimams. Kiekvienam testavimo scenarijui buvo sukurti atvejai, apimantys tiek teigiamus, tiek neigiamus naudotojo veiksmus. Testai apėmė svarbiausias funkcijas, tokias kaip prisijungimas, registracija, žuvies atpažinimas, žvejo bilieto valdymą, žemėlapio žymekliai ir statistika. Testavimas parodė, kad sistema stabiliai apdoroja teisingus ir neteisingus duomenis bei atitinka nustatytus reikalavimus. Kiekvienas testavimo atvejis buvo įvykdytas sėkmingai, todėl galima teigti, kad prototipas veikia pagal planą.

## Išvados

1. Atlikus žvejybinių sistemų apžvalgą ir palyginimą paaiškėjo, kad nors rinkoje egzistuoja kelios atskiros žvejybos sprendimų platformos, trūksta vieningos ir lokalizuotos sistemos, pritaikytos Lietuvos teisiniam kontekstui ir žvejų poreikiams. Tai pabrėžia poreikį sukurti integruotą žvejybos programėlę, kuri atitiktų šiuos reikalavimus.
2. Išnagrinėjus ir palyginus technologines alternatyvas, buvo pasirinktos tinkamiausios technologijos mobiliosios programėlės kūrimui: „Java“ programavimo kalba, „MySQL“ duomenų bazė ir „Fishial“ žuvų atpažinimo funkcijai. Šios technologijos užtikrina reikiamą funkcionalumą, patikimumą ir atitinka sistemos tikslus.
3. Atsižvelgiant į technologinę analizę, buvo suformuluoti 24 funkciniai ir 15 nefunkcinių reikalavimų, kurie padėjo kryptingai sukurti sistemą. Tai apėmė pagrindines funkcijas, tokias kaip žuvų atpažinimas, žvejybos taisyklių pateikimas pagal žuvį, žvejybos vietų žymėjimas žemėlapyje ir statistikos generavimas. Taip pat buvo apibrėžti pagrindinės sistemos veikimo ypatybės kaip sistemos pasiekiamumas, saugumas bei prieinamumas. Projektavimas buvo atliktas atsižvelgiant į gautus reikalavimus, siekiant sukurti aiškią, patogią ir lengvai plėtojamą sistemą, kuri užtikrintų tiek funkcionalumą, tiek stabilumą ilgalaikėje perspektyvoje.
4. Sukurtas prototipas sėkmingai sujungia žvejybos taisykles, žuvų atpažinimo funkciją ir geografinę informaciją į vieną sistemą, kurios pagrindinis tikslas – palengvinti žvejų darbą ir užtikrinti teisingą žvejybos praktiką pagal teisines nuostatas. Prototipas taip pat leidžia valdyti žvejo bilietą ir generuoti vertingą statistinę informaciją apie žvejybos vietas ir laimikį. Atliktas testavimas nustatė, kad pagrindinės programėlės funkcijos veikia kaip numatyta ir atitinka reikalavimus. Tai užtikrina gerą naudotojo patirtį.

Apibendrinant galima teigti, kad sukurta sistema ne tik išsprendžia praktines problemas, su kuriomis susiduria Lietuvos žvejai, bet ir turi potencialo prisidėti prie žvejybos kultūros tobulinimo bei technologinių sprendimų plėtros šioje srityje.

# LITERATŪROS ŠALTINIAI

Ferguson, R. A., & Tufts, B. L. (1992). *Physiological Effects of Brief Air Exposure in Exhaustively Exercised Rainbow Trout (Oncorhynchus mykiss): Implications for “Catch and Release” Fisheries.* *Canadian Journal of Fisheries and Aquatic Sciences*, *49*(6), 1157–1162. <https://doi.org/10.1139/f92-129>

Schreer, J. F., Resch, D. M., Gately, M. L., & Cooke, S. J. (2005). *Swimming Performance of Brook Trout after Simulated Catch‐and‐Release Angling: Looking for Air Exposure Thresholds*. *North American Journal of Fisheries Management*, *25*(4), 1513–1517. <https://doi.org/10.1577/m05-050.1>

Kesminas, V. ir Stakėnas, S. (2020). Ataskaita. *Rekomendacijų dėl aštriašnipių eršketų populiacijos atkūrimo Lietuvoje 2021–2025 metais programos parengimas*. <https://zum.lrv.lt/uploads/zum/documents/files/GTC%20Ersketu%20programa%20pataisyta%202020.pdf>

Dainys, J., Jakubavičiūtė, E., Gorfine, H., Kirka, M., Raklevičiūtė, A., Morkvėnas, A., Pūtys, Ž., Ložys, L., Audzijonytė, A. (2023). *MĖGĖJIŠKOS ŽVEJYBOS POVEIKIO ĮVERTINIMAS KAUNO MARIŲ ŽUVŲ POPULIACIJOMS JŪROS IR KRANTŲ TYRIMAI 2023 Konferencijos medžiaga*, 34.

Ložys, L., Pūtys, Ž., Dainys, J., Kirka, M., Levickienė, D., Rimkus, R., Jakubavičiūtė, E., Jaselionis, A., (2017). *KAUNO MARIŲ ŽUVŲ IŠTEKLIŲ ĮVERTINIMAS, ŽUVŲ IŠTEKLIŲ NAUDOJIMO IR ATKŪRIMO PLANŲ 2016-2017 M. PARENGIMAS*. *Ataskaita*. [*https://am.lrv.lt/uploads/am/documents/files/Gamtos%20apsauga%20ir%20mi%C5%A1kai/Gamtos%20apsauga/%C5%BEvejyba/moksliniai%20tyrimai/Kauno%20mariu%20zuvu%20tyrimu%20ataskaita%202016-2017.pdf*](https://am.lrv.lt/uploads/am/documents/files/Gamtos%20apsauga%20ir%20mi%C5%A1kai/Gamtos%20apsauga/%C5%BEvejyba/moksliniai%20tyrimai/Kauno%20mariu%20zuvu%20tyrimu%20ataskaita%202016-2017.pdf)

*D1-272 Dėl Lietuvos Respublikos aplinkos ministro 2013 m. sausio 4 d. įsakymo Nr. D1-14 „Dėl Mėgėjų žvej. . . (2021).* [*https://e-seimas.lrs.lt/portal/legalAct/lt/TAD/24df36d2aea411eba6328c92adabc234*](https://e-seimas.lrs.lt/portal/legalAct/lt/TAD/24df36d2aea411eba6328c92adabc234)

*D1-136 Dėl Lietuvos Respublikos aplinkos ministro 2013 m. sausio 4 d. įsakymo Nr. D1-14 „Dėl Mėgėjų žvej. . . (2024).* [*https://e-seimas.lrs.lt/portal/legalAct/lt/TAD/1caa3ac2066011ef8e4be9fad87afa59?positionInSearchResults=0&searchModelUUID=f32c5fe5-995a-4478-9df3-5236edb5c3a6*](https://e-seimas.lrs.lt/portal/legalAct/lt/TAD/1caa3ac2066011ef8e4be9fad87afa59?positionInSearchResults=0&searchModelUUID=f32c5fe5-995a-4478-9df3-5236edb5c3a6)

Duchovskis, P., Baršienė, J., Dapkus, R., Gribauskienė, A., Kupstaitis, N., Makarevičienė, V., ... & Viškelis, P. (2022). *Nacionalinė mokslo programa „Agro-miško ir vandens ekosistemų tvarumas “.* [*https://zum.lrv.lt/uploads/zum/documents/files/Nacionalin%C4%97s%20mokslo%20programos\_ataskaita\_2023.pdf*](https://zum.lrv.lt/uploads/zum/documents/files/Nacionalin%C4%97s%20mokslo%20programos_ataskaita_2023.pdf)

APLINKOS APSAUGOS DEPARTAMENTO PRIE APLINKOS MINISTERIJOS GYVOSIOS GAMTOS APSAUGOS DEPARTAMENTAS, (2025). *DĖL duomenų pateikimo* **.**

*žvejo gidas* (b.d.). <https://zvejogidas.lt/apie-projekta>

Deepersonar. (b.d.). *Fish deeper*. <https://deepersonar.com/lt-lt/pages/fish-deeper>

*FishDetect - Fish Identifier – „Google Play“ programos*. (b.d.). <https://play.google.com/store/apps/details?id=com.fishaidetect&hl=lt>

*Dažniausiai užduodami klausimai - Lietuvos Respublikos aplinkos ministerija*. (b.d.). <https://am.lrv.lt/lt/veiklos-sritys-1/gamtos-apsauga/zvejyba/megeju-zvejyba/dazniausiai-uzduodami-klausimai/>

*Admin. (2022). Importance of datasets in AI. Buff ML.* [*https://buffml.com/importance-of-datasets-in-ai/*](https://buffml.com/importance-of-datasets-in-ai/)

Wu, H., Liu, Q., & Liu, X. (2019). A review on deep learning approaches to image classification and object segmentation. *Computers, Materials & Continua/Computers, Materials & Continua (Print)*, *60*(2), 575–597. <https://doi.org/10.32604/cmc.2019.03595>

*Fishail (b.d.).* [*https://www.fishial.ai/#about-us*](https://www.fishial.ai/%23about-us)

*WHOIGit/FishDetector: Detect fish in images and video*. (b.d.). GitHub. <https://github.com/WHOIGit/FishDetector>

McKenzie, C. (2021). *5 reasons why Java is still the best programming language*. TheServerSide.com. <https://www.theserverside.com/feature/5-reasons-why-Java-is-still-the-best-programming-language>

Bose, S. (2018). A COMPARATIVE STUDY: JAVA VS KOTLIN PROGRAMMING IN ANDROID APPLICATION DEVELOPMENT. *International Journal of Advanced Research in Computer Science*, *9*(3), 41–45. <https://doi.org/10.26483/ijarcs.v9i3.5978>

Grishina, A. (2024,). *Java vs. C#: What is the Right Choice for Your Project?* SoftTeco. <https://softteco.com/blog/java-vs-csharp>

Erickson, J. (2024). *MySQL: Understanding what it is and how it’s used*. <https://www.oracle.com/mysql/what-is-mysql/>

Salunke, S. V., & Ouda, A. (2024). A performance benchmark for the PostgreSQL and MySQL databases. *Future Internet*, *16*(10), 382. <https://doi.org/10.3390/fi16100382>

*PostgreSQL: about*. (b.d.). The PostgreSQL Global Development Group. <https://www.postgresql.org/about/>

Ilić, M., Kopanja, L., Zlatković, D., Trajković, M., & Ćurguz, D. (2021). Microsoft SQL Server and Oracle: Comparative performance analysis. The 7th International Conference Knowledge Management and Informatics, Vrnjačka Banja, June 2021, 33–40.

<https://www.researchgate.net/profile/Dragan-Zlatkovic/publication/352348811_MICROSOFT_SQL_SERVER_AND_ORACLE_COMPARATIVE_PERFORMANCE_ANALYSIS/links/60c43847a6fdcc2e613650dc/MICROSOFT-SQL-SERVER-AND-ORACLE-COMPARATIVE-PERFORMANCE-ANALYSIS.pdf>

*ChatGPT*. (n.d.). <https://chatgpt.com/>

*Gemini*. (n.d.). <https://gemini.google.com/>

## 1 PRIEDAS (išrtrauka iš aplinkos apsaugos departamento prie aplinkos ministerijos gyvosios gamtos apsaugos departamento atsiųstų duomenų)

APLINKOS APSAUGOS DEPARTAMENTO PRIE APLINKOS MINISTERIJOS GYVOSIOS GAMTOS APSAUGOS DEPARTAMENTAS

<...>.

Remiantis Administracinių nusižengimų registro duomenimis, teikiame informaciją apie 2019 - 2024 metais padarytus Mėgėjų žvejybą reglamentuojančių teisės aktų pažeidimus už kuriuos asmenys buvo patraukti administracinėn atsakomybėn pagal ANK 291 straipsnio 1 ir 2 dalis.

2019 m. 291 straipsnio 1 dalis - administracinių nusižengimų skaičius: 944, 2 dalis – 74;

2020 m. 291 straipsnio 1 dalis - administracinių nusižengimų skaičius: 1096, 2 dalis – 95;

2021 m. 291 straipsnio 1 dalis - administracinių nusižengimų skaičius: 1663, 2 dalis – 166;

2022 m. 291 straipsnio 1 dalis - administracinių nusižengimų skaičius: 1455, 2 dalis – 115;

2023 m. 291 straipsnio 1 dalis - administracinių nusižengimų skaičius: 693, 2 dalis – 63;

2024 m. 291 straipsnio 1 dalis - administracinių nusižengimų skaičius: 619, 2 dalis – 36.

Kilus papildomų klausimų maloniai prašome susisiekti žemiau nurodytais kontaktais. Direktorė Mantė Ramanauskienė