



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

Bakalauro baigiamasis darbas

Kelionės maršruto sudarymo sistema

Atliko:

Deividas Grušnius

parašas

Vadovas:

dr. Joana Katina

Vilnius

2020

Turinys

Santrauka	3
Summary	4
Įvadas.....	5
1. Panašių sistemų palyginimas.....	6
2. Sistemos modelis.....	7
2.1 Sistemos funkcionalumas	7
2.1.1 Funkciniai reikalavimai.....	7
2.1.2 Nefunkciniai reikalavimai.....	8
2.2 UML diagramos.....	8
2.2.1. Naudojimo atvejų diagrama	8
2.2.2. Veiklų diagrama	9
2.2.3. Būsenų diagrama.....	10
2.3 Sistemos navigacijos struktūra.....	11
2.4 Duomenų bazės struktūra.....	12
3. Programinė realizacija.....	14
3.1 Programinei realizacijai naudojami įrankiai	14
3.2 Internetinės svetainės objektai	14
3.3 Lankytinų vietų paieškos puslapis	16
3.4 Maršrutų paieškos puslapis.....	18
3.5 Lankytinos vietos informacinis puslapis.....	20
3.6 Administracinis puslapis.....	21
Išvados ir rekomendacijos	24
Literatūros šaltiniai	25

Santrauka

Darbo tikslas – sukurti internetinę svetainę su jam valdyti skirtais įrankiais, kuris suteiktų naudotojams galimybę lengvai prisijungti prie sistemos, iš duoto sąrašo pasirinkti miestą, iš kurio norite išvykti bei kur norite nuvykti, ir gauti rezultatų sąrašą su atitinkama informacija: pilnas maršrutas su miestais, miestuose esančios lankytinos vietos, visas kelionės atstumas kilometrais, atstumai tarp dviejų miestų. Taip pat naudotojas gali prie lankytinų vietų palikti komentarą, uždėti reitingą, pažymėti, kad jos patiko. Tikslui pasiekti buvo aprašyti uždaviniai, palygintos panašios, jau egzistuojančios sistemos. Darbo rezultatas – sukurta informacinė sistema su jai skirtais administruoti įrankiais, kuriuose skirtingų rūšių naudotojai gali atlikti jiems leistinus veiksmus. Sistema turi lankytinų vietų peržiūrėjimo, paieškos bei filtravimo funkcijas pagal miestus arba įvestą norimą pavadinimą, maršrutų radimo funkciją pagal pradinį ir galinį maršruto taškus.

Summary

Travel routing system

The goal of this project is to create a website with the tools for its control, which would give an opportunity for users to easily access the system, choose a city where you want to start and where to finish from a given list, and get the results with the following information: full route with cities, places to visit in those cities, the full distance of the trip in kilometers, the distances between two cities. In addition, the user can leave a comment, rate and leave a like on places to visit. In order to achieve the goal certain assignments were created, similar systems were looked over, which already exist. The result is a created information system with administrative tools, which can be used by various user types with the rights to take possible actions. The system has a route search function by the start and end point, also review, search and filtering functions for places to visit.

Įvadas

Išmanieji telefonai ir kompiuteriai šiomis dienomis yra neatsiejami beveik visose gyvenimo srityse, įskaitant ir keliavimą. Žmonėms dažnai kyla klausimų kelionės metu dėl trumpiausio maršruto suradimo, esančių lankytinų vietų [6], įeinančių į maršruto miestus [5]. Šio darbo tikslas – sukurti internetinę svetainę, kuri duotų galimybę naudotojams be jokių didelių pastangų prisijungti prie informacijos šaltinio bei gauti jiems aktualią informaciją apie maršutus, į maršutus įeinančius miestus ir juose esančias lankytinas vietas, pasirinkus norimą kelionės pradžios bei pabaigos miestą.

Pagrindinis uždavinys internetinei svetainei yra pateikti naudotojams visą reikalingą informaciją apie Lietuvos miestuose esančias lankytinas vietas ir suteikti galimybę naudotojams parašyti savo nuomonę, palikti lankytinos vietos įvertinimą nuo vieno iki penkių bei pažymėti, kad vieta patiko. Šiomis dienomis egzistuoja panašių internetinių svetainių ir mobiliųjų aplikacijų, bet jose visą laiką reikia asmeninės informacijos iš paskyrų, tokių kaip Facebook, Google, Twitter ir taip toliau. Internetuose esančių apklausų rezultatai rodo, kad naudotojams būtų priimtinau komentuoti anonimiškai tokiose sistemose, tačiau norint valdyti komentarų turinį, reikia sukurti šioje svetainėje paskyrą, kur administratorius galėtų kontroliuoti naudotojų komentarus. Internetinėje svetainėje įvykdytas ir toks uždavinys: galimybė naudotojui rasti trumpiausią maršrutą nuo pradinio miesto A iki galutinio miesto B panaudojant Dijkstra grafų algoritmą, taip pat pateiktame trumpiausiam maršrute rodomi ir atstumai tarp kiekvieno sustojimo (dviejų miestų) bei visas atstumas kilometrais, kartu su į maršrutą įeinančių miestų lankytinomis vietomis.

Tikslui pasiekti iškeltas ir dar vienas uždavinys – internetinei svetainei administruoti paskirtos priemonės, prie kurių prieigą turi tik administratoriai. Šio tikslo paskirtis – kontroliuoti svetainėje visiems matomą informaciją: lankytinas vietas bei naudotojų komentarus, kas būtų lankytinų vietų ir naudotojų komentarų sukūrimas, redagavimas arba ištrynimasis.

1. Panašių sistemų palyginimas

Prieš pradėdant daryti darbą buvo lyginamos dar kelios su turizmu susijusios internetinės svetainės. Jas vertinant buvo atsižvelgiama į tai, ar buvo atitikti svetainėms iškelti kriterijai [7]. Pats svarbiausias iš jų – tai informacinės sistemos pritaikomumas lietuviškai kalbantiems žmonėms. Sekantys kriterijai yra techniniai, kur yra atsižvelgiama į tai, kokias funkcijas naudotojams leistina atlikti. Pirmasis aspektas palyginimui buvo tai, kokia informacija pateikiama naudotojams, ar tik administratoriaus sukurta, ar ir kitų naudotojų pasidalinama. Kadangi sistemoje yra labai svarbu parodyti kuo įvairesnę informaciją, buvo atliktas sprendimas ieškoti svetainių, kuriose būtų leidžiama dalintis informacija ir paprastiems naudotojams. Antrasis aspektas buvo galimybė naudotojui pačiam susiorganizuoti kelionę – susidaryti maršrutus, kurie būtų vėliau peržiūrimi iš kitų įrenginių. Kelios svetainės turėjo šią funkciją, tačiau tai buvo padaryta naudojantis slapukais, todėl naudotojo maršrutas buvo prieinamas tik iš to įrenginio, kuriame jis buvo sukurtas. Trečiasis aspektas ypatingai svarbus vertinant buvo globalios padėties nustatymo sistemos panaudojimas sukuriant kelionės maršrutą ir pateikiant informaciją, kaip atsidurti iki norimos vietos. Šis funkcionalumas labiau svarbus naudotojams, naudojančiams mobiliuosius įrenginius, nes galima paprastai patikslinti atstumą iki vietos, kad ir pačios kelionės metu.

Atsižvelgus į aukščiau paminėtus aspektus buvo palygintos šios internetinės svetainės:

- **vilnius-tourism.lt** – svetainėje trūksta lankytinų vietų komentavimo, įvertinimo balais, „like“ palikimo funkcijų. Yra rekomenduojami maršrutai, bet kad sudaryti savo maršrutą, įmanoma lankytinas vietas tik pažymėti, todėl negalima visų vietų sujungti į bendrą maršrutą ir gauti jo tolimesnę informaciją.
- **lithuania.travel** - svetainė suteikia galimybę naudotojui sudaryti maršrutus, kuriam yra priskiriamos lankytinos vietos, bet trūksta GPS funkcijos maršrutui iki norimos vietos rasti. Taip pat yra aktualios informacijos trūkumas naudotojams, kaip bilietų kainų ar darbo laiko.
- **turizmas.lt** - svetainė turi ganėtinai nemažai reikalingų funkcijų, bet yra naudotojų maršruto sudarymo ir lankytinų vietų komentavimo, įvertinimo balais bei „like“ palikimo funkcijų trūkumas.

2. Sistemos modelis

Šiame skyriuje bus aprašyta informacija, kuri buvo surenkama iki darbo praktinės dalies atlikimo pradžios. Pateikiamas sukurtas teorinis sistemos modelis – įvardintas sistemos funkcionalumas, apibrėžiant iškeltus funkcinius ir nefunkcinius reikalavimus, pavaizduojamos pagrindinį funkcionalumą apibūdinančios UML diagramos, pateikiamos ir apibūdinamos sistemos navigacijos ir duomenų bazės struktūros.

2.1 Sistemos funkcionalumas

Sistemos funkcionalumui aprašyti pateikiamas sąrašas funkcinių ir nefunkcinių reikalavimų, atsižvelgiant į sistemos paskirtį. Dėmesys skiriamas sistemos naudotojams, kad būtų užtikrintas kuo paprastesnis internetinės svetainės funkcijų naudojimas bet kokio amžiaus žmonėms.

2.1.1 Funkciniai reikalavimai

Sistemai keliami tokie funkciniai reikalavimai:

- Sistemoje turi veikti lankytinų vietų paiešką pagal įvestą norimą reikšmę.
- Sistema turi gebėti atlikti lankytinų vietų paiešką pagal pasirinktą miestą iš duoto sąrašo.
- Sistemoje turi būti leidžiama atlikti lankytinų vietų rikiavimą pagal komentarų skaičių, vidutinį įvertinimą bei „Like“ kiekį.
- Sistema turi leisti atlikti maršruto paiešką pagal pasirinktus pradžios ir pabaigos miestus iš duotų sąrašų.
- Prisijungęs naudotojas privalo turėti galimybę parašyti komentarą šalia atidarytos lankytinos vietos.
- Prisijungęs naudotojas privalo turėti galimybę įvertinti balais pravertą lankytiną vietą.
- Prisijungęs naudotojas privalo turėti galimybę palikti „Like“ ant pasirinktos lankytinos vietos.
- Bet kokiam neregistruotam naudotojui turėtų būti įgalintas užsiregistravimas sistemoje.
- Užsiregistravęs vartotojas turėtų galėti prisijungti prie sistemos.
- Administratoriui turėtų būti suteikta prieiga, kad pasiekti administracinį puslapį.

2.1.2 Nefunkciniai reikalavimai

Siekiant užtikrinti sklandų sistemos veikimą, buvo iškelti tokie nefunkciniai reikalavimai:

- Sistema turi užtikrinti, kad nenustatyto tapatumo asmuo negalėtų atlikti prieigos teisių reikalaujamų veiksmų.
- Sistema turi veikti sklandžiai, be žymių strigimo (vėlavimo) požymių.
- Sistema turi būti pritaikyta naudoti su stacionariu ar nešiojamu kompiuteriu.

2.2 UML diagramos

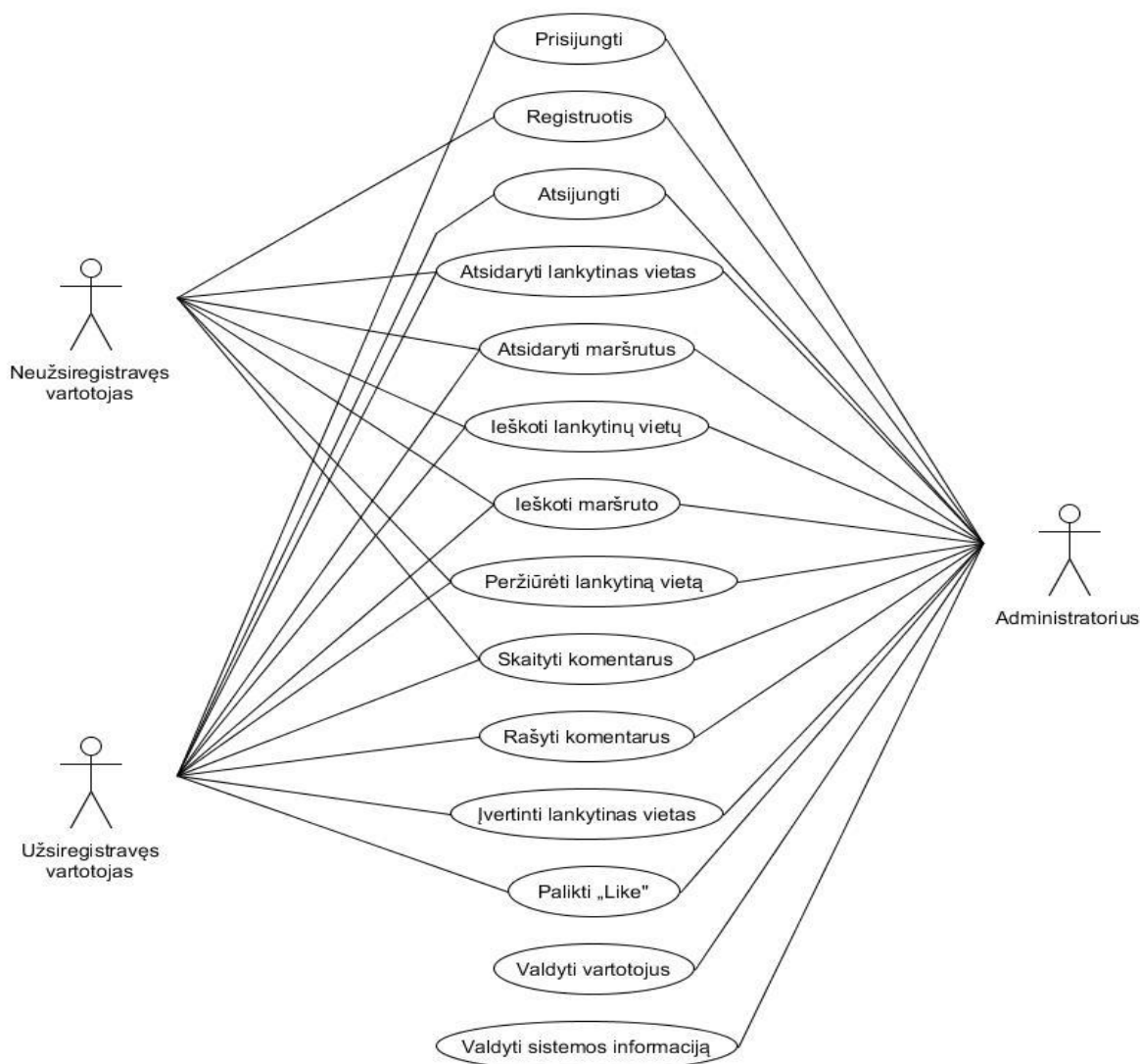
Šiame poskyryje pateikiamos ir aprašomos pagrindinį sistemos funkcionalumą apibūdinančios UML diagramos – naudojimo atvejų, veiklų bei būsenų.

2.2.1. Naudojimo atvejų diagrama

Tipų teisės yra paskirstytos taip, kad administratorius gali valdyti visą duomenų bazę administraciniame puslapyje, užsiregistravęs naudotojas gali ant lankytinų vietų rašyti komentarus, įvertinti balais ir palikti „Like“, o visas likusias funkcijas, tokias, kaip maršruto ar lankytinos vietos paiešką, gali atlikti ir neužsiregistravęs naudotojas. Diagramoje (žiūrėti 1 pav.) nurodytų naudojimo atvejų aprašymas:

- **Prisijungti** – prisijungti prie internetinės svetainės pasinaudojantis sukurtos paskyros duomenimis.
- **Registruotis** – susikurti paskyrą nurodant naudotojo vardą, pavardę, slapyvardį, slaptažodį, elektroninio pašto adresą bei nuotrauką.
- **Atsijungti** – nutraukti prisijungimo metu sukurtą sesiją.
- **Atsidaryti lankytinas vietas** – apsilankyti „Lankytinos vietos“ puslapyje.
- **Atsidaryti maršrutus** – apsilankyti „Maršrutai“ puslapyje.
- **Ieškoti lankytinų vietų** – įvedus norimą reikšmę arba pasirinkus miestą iš sąrašo rasti lankytinas vietas.
- **Ieškoti maršruto** – pasirinkus pradžios ir pabaigos miestus iš sąrašo rasti maršrutą.
- **Peržiūrėti lankytiną vietą** – apsilankyti pasirinktos lankytinos vietos puslapyje, kur yra aprašyta detalesnė informacija.
- **Skaityti komentarus** – peržiūrėti paskelbtas viešas nuomones apie tam tikrą lankytiną vietą.

- **Rašyti komentarus** – skelbti viešai matomą nuomonę apie tam tikrą lankytiną vietą.
- **Įvertinti lankytinas vietas** – paskirti balą nuo vieno iki penkių ant tam tikros lankytinos vietos.
- **Palikti „Like“** – pažymėti, kad tam tikra lankytina vieta patiko.
- **Valdyti naudotojus** – keisti naudotojų būsenas ir ištrinti paskyras iš duomenų bazės.
- **Valdyti sistemos informaciją** – koreguoti lankytinas vietas, komentarus.

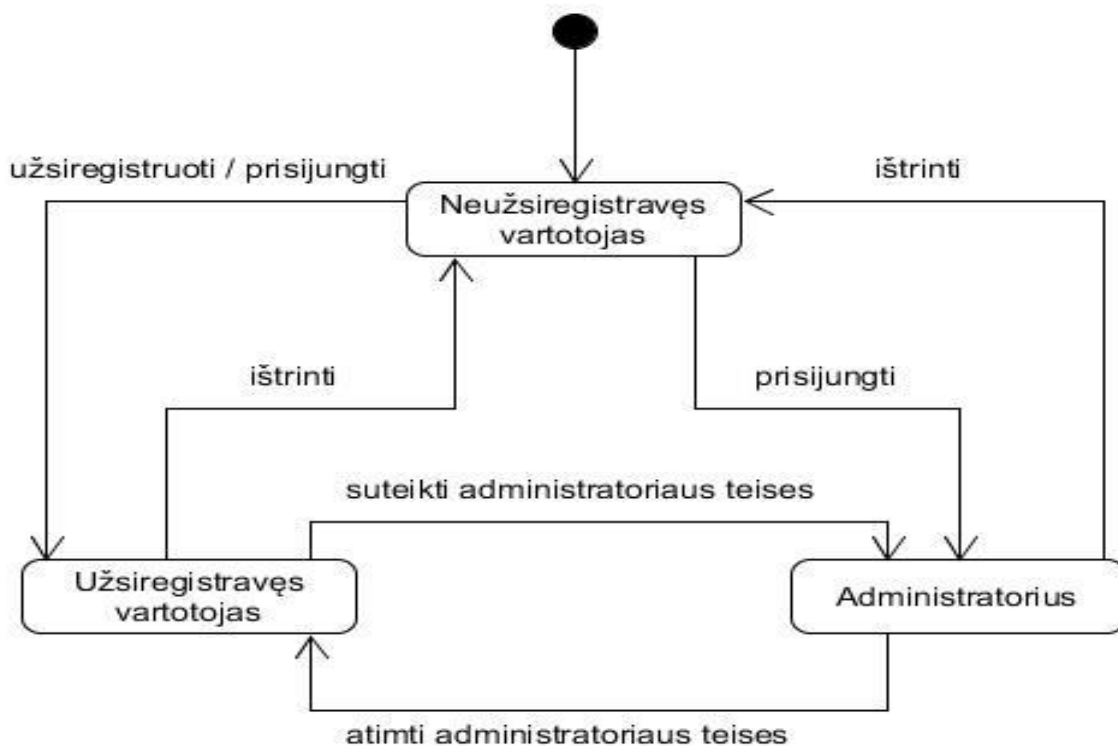


1 pav. Sistemos naudojimo atvejų diagrama

2.2.2. Veiklų diagrama

Šioje informacinėje sistemoje naudotojas gali būti trijų rūšių: užsiregistravęs lankytojas, neužsiregistravęs lankytojas arba administratorius. Iš pat pradžių patekus į internetinę svetainę ir neprisijungus, naudotojas laikomas kaip neužsiregistravęs lankytojas. Ši būsena pasikeičia

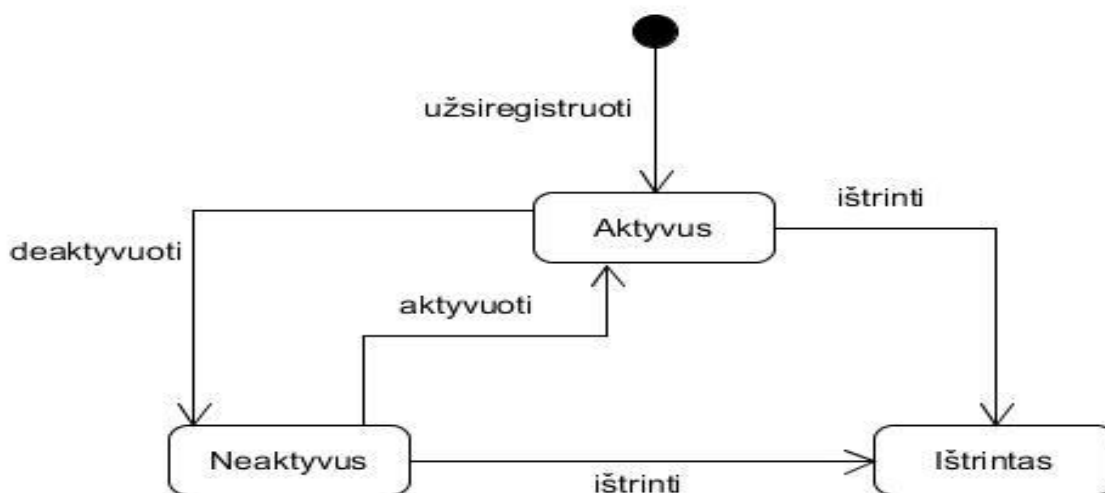
užsiregistravus, ir priklausomai nuo paskyros rūšies naudotojas tampa arba užsiregistravusiu lankytoju, arba administratoriumi. Užsiregistravęs lankytojas turi galimybę tapti administratoriumi, jeigu administratorius suteikia tam teises. Nutinka atvirkščias reiškinys, jeigu teisės iš administratoriaus yra atimamos. Tiek administratorius, tiek užsiregistravęs naudotojas gali vėl tapti neregistruotais tuo atveju, jeigu jų paskyros yra ištrinamos. Naudotojų rūšys ir jų galimi pasikeitimai atvaizduojami duotame 2 paveikslėlyje.



2 pav. Sistemos naudotojo veiklų diagrama

2.2.3. Būsenų diagrama

Šioje internetinėje svetainėje visi užsiregistravę naudotojai, tiek administratoriai, tiek registruoti lankytojai, yra skirstomi ir pagal būsenas. Tai būtų aktyvūs, neaktyvūs arba ištrinti naudotojai. Užsiregistravęs naudotojas tampa aktyviu. Tolimesnė naudotojo būsena priklauso jau nuo administratoriaus, jis gali paskyrą deaktyvuoti ir aktyvuoti, būsenų pasikeitimai tarp šių dviejų gali įvykti daug kartų. Gali atsitikti ir taip, kad administratorius nusprendžia paskyrą pašalinti iš duomenų bazės, tokiu atveju naudotojas tampa ištrintas (ši būsena yra negrįžtama). Naudotojų būsenų ir jų pasikeitimų schema pavaizduota 3 paveikslėlyje.

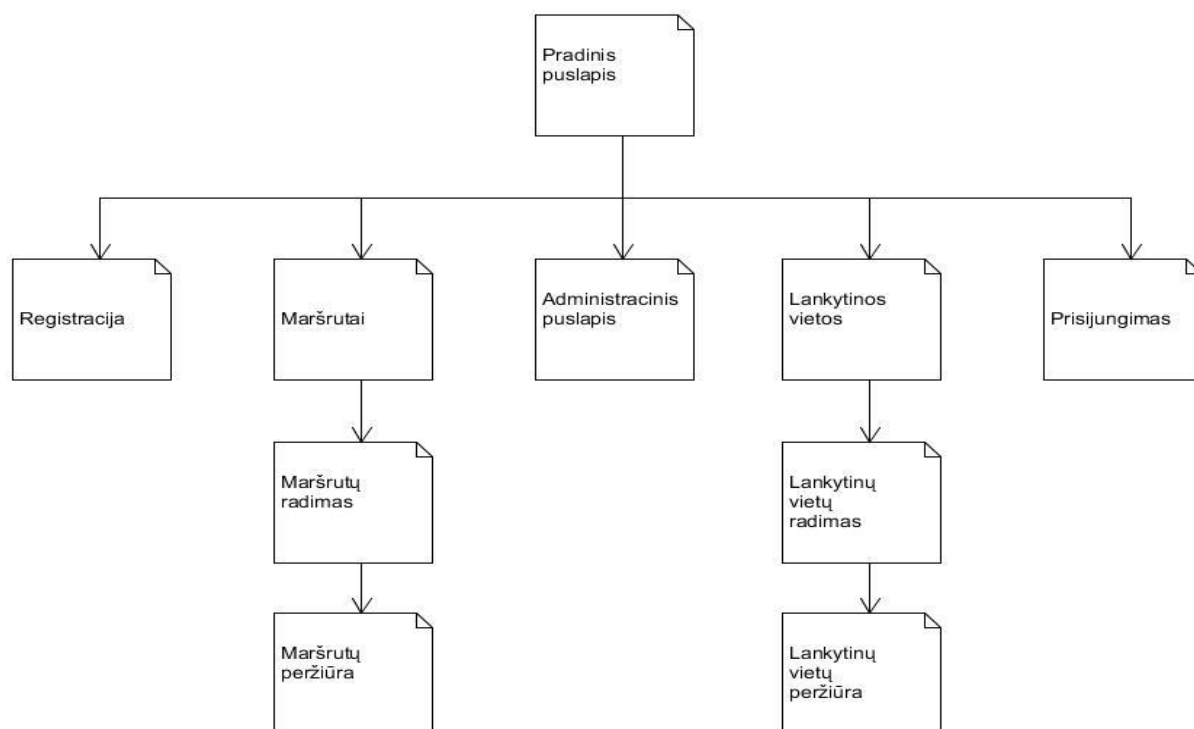


3 pav. Sistemos naudotojo būsenų diagrama

2.3 Sistemos navigacijos struktūra

Navigavimas puslapiuose yra glaudžiai susijęs su naudotojų tipais, kurie yra trys – administratorius, užsiregistravęs naudotojas ir neužsiregistravęs naudotojas. Iš pirminio puslapio galima pasiekti didžiąją dalį kitų, kadangi siekta išvengti didelio paspaudimų kiekio, kuris, be abejo, internetinės svetainės naudotojams yra nepatogus.

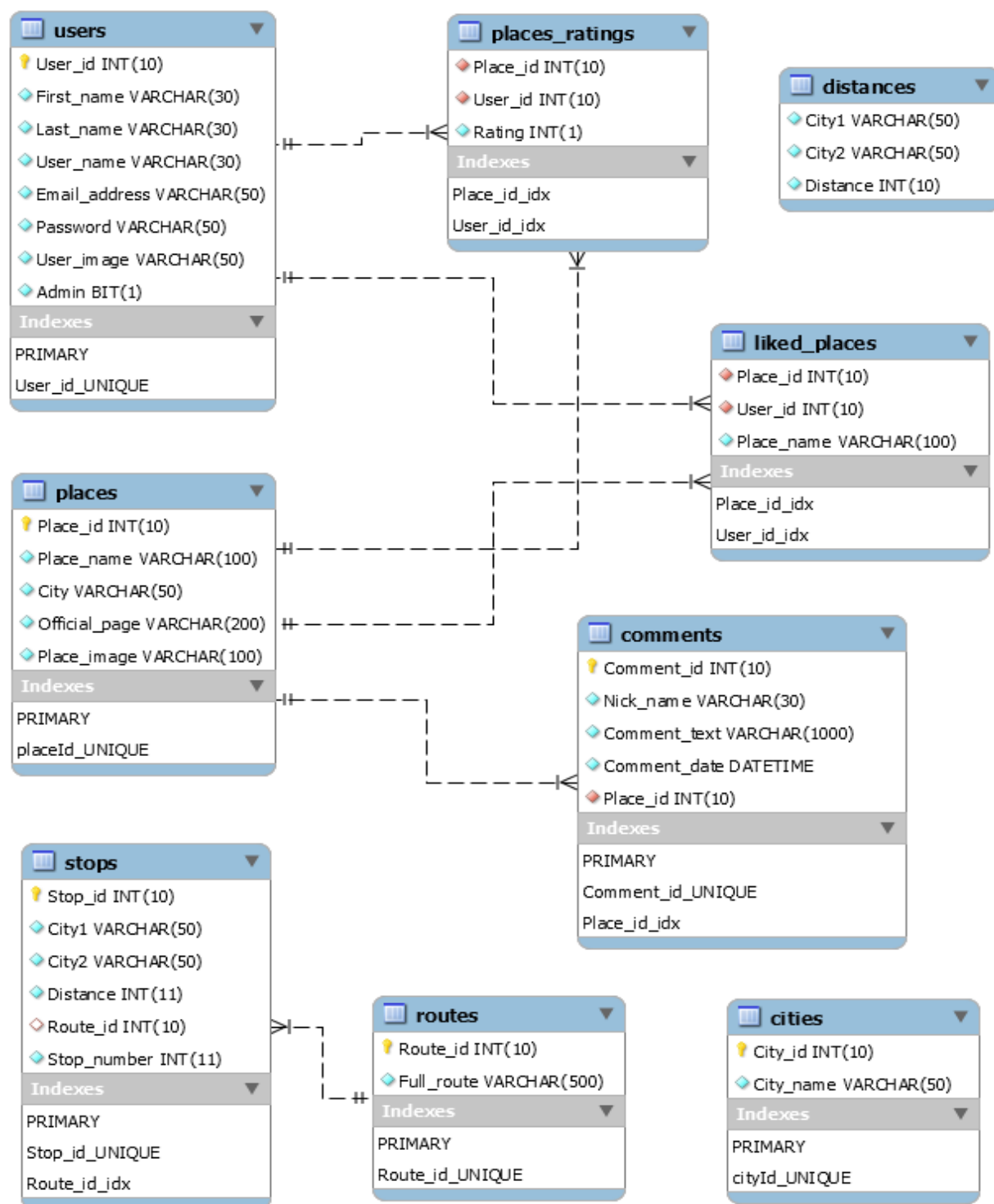
Neužsiregistravusiems naudotojams yra pasiekiami visi puslapiai, išskyrus vienas, kuris yra skirtas tik administratoriams valdyti duomenų bazę. Tas pats galioja ir užsiregistravusiems naudotojams, bet jie gali atlikti kelias funkcijas, kurių negali neužsiregistravę naudotojai, ir tai yra lankytinų vietų komentavimas, įvertinimas balais bei „like“ palikimas. Administratoriai gali pasiekti visus puslapius, įskaitant ir tik jiems skirtą puslapį valdyti duomenų bazę. Sistemos navigacijos schema pateikta 4 paveikslėlyje.



4 pav. Sistemos navigacijos schema

2.4 Duomenų bazės struktūra

Šioje informacinėje sistemoje naudojama duomenų bazė skirta tinklalapiui sukurtų objektų saugojimui. Kiekvienas objektas turi savo lentelę su būtent jam būdingais atributais, apribojimais, pirminiu raktu, sąryšiais tarp kitų objektų. Šiai internetinei svetainei yra naudojama MySQL duomenų bazė ir jos schema yra pavaizduota 5 paveikslėlyje.



5 pav. Sistemos duomenų bazės schema

3. Programinė realizacija

Šiame skyriuje pateikiama sistemos funkcionalumo programinė realizacija. 3.1 poskyryje įvardinti programinei realizacijai naudojami įrankiai, visuose kituose poskyriuose paaiškinami pagrindinį sistemos funkcionalumą apibūdinantys sprendimai.

3.1 Programinei realizacijai naudojami įrankiai

Internetinei svetainei sukurti naudotos technologijos buvo pasirinktos atsižvelgus į jų saugumą, aktualumą, populiarumą, galimybes ir asmeninę patirtį.

Pagrindiniai stilizavimo darbai buvo realizuoti naudojantis HTML, CSS ir JavaScript karkasu Bootstrap [3] bei jo grafiniais elementais, tokiais kaip navigaciniai meniu, nuotraukų karuselės, kortelės, mygtukai, įvesties laukai, konteineriai, puslapiavimas, pranešimų lentelės bei išskleidžiamieji sąrašai.

Serverio pusės programavimui buvo naudojamas PHP programavimo kalba parašytas karkasas CodeIgniter [4]. Šis karkasas yra pats primityviausias, skirtas pradedantiesiems kurti internetines svetaines, turintis didelį kiekį bibliotekų, pagelbiklių ir funkcijų, skirtų sukurti pilnai funkcionuojančią svetainę. Šio karkaso veikimas pagrįstas MVC (Model-View-Controller) principu. Model dalis yra atsakinga už darbą su duomenų baze, joje visos funkcijos yra konvertuojamos į SQL užklausas, kad būtų galima sukurti, ištrinti, pakeisti arba parodyti duomenis. Ši dalis taip pat gali atlikti ir patikrinimo veiksmus dėl kažkokios įvesties validacijos iš Controller dalies. Controller dalis atsako už įvesties priimimą, jos patikrinimą ir perdavimą Model daliai. Ši dalis taip pat perduoda duomenis, gautus iš Model dalies funkcijų rezultatų. View dalis atvaizduoja informaciją puslapyje, gautą iš Model dalies funkcijų per Controller dalį. Iš šios dalies taip pat galima paimti įvesti ir perduoti Model daliai, kartu su Controller dalies pagalba. Naudojantis šiuo karkasu buvo plačiai naudojamos tokios bibliotekos, kurios susijusios su puslapiavimu, sesija, įvesties formos validacija, datos manipuliavimu, puslapio perkrovimu bei pilno puslapio adreso arba adreso dalies gavimu.

Internetiniai svetainei buvo naudojama MySQL reliacinė duomenų bazių valdymo sistema, nes ji yra ganėtinai plačiai paplitusi tarp CodeIgniter projektų ir turi gerą suderinamumą su CodeIgniter duomenų bazės užklausų formavimu. CodeIgniter ir MySQL duomenų bazės susiejimui buvo naudojami „MySQL Workbench“ ir „phpMyAdmin“ įrankiai.

3.2 Internetinės svetainės objektai

Tinklalo objektai yra labai svarbūs šio projekto įgyvendinimui ir be jų informacinė sistema neturėtų prasmės. Juose talpinama pagrindinė informacija apie naudotojus, miestus, atstumus tarp miestų, lankytinas vietas, pilnus maršrutus, maršrutų sustojimus, naudotojus, komentarus, įvertinimus, „Like“ palikimus. Į objektus įeinančios lentelės:

- cities – „Miestai“. Ši lentelė turi laukus: miesto ID (City_id) ir miesto pavadinimas (City_name) atributus. Ji yra skirta gauti sąrašą visų miestų, kai norima susirasti lankytinų vietų pagal tam tikrą miestą. Taip pat lentelė yra naudojama administraciniame puslapyje, kai reikia sukurti arba pakoreguoti tam tikrą lankytiną vietą. Pats svarbiausias šios lentelės panaudojimas – tai maršruto paieškoje, kai reikia pasirinkti iš sąrašo, iš kokio miesto ir į kokį norima nuvykti.

- **comments** – „Nuomonės“. Ši lentelė turi laukus: komentaro ID (Comment_id), slapyvardis (Nick_name), komentaro tekstas (Comment_text), komentaro data (Comment_date) ir lankytinos vietos ID (Place_id) atributus. Ji skirta parodyti naudotojų nuomones apie tam tikras lankytinas vietas bei tas nuomones išsaugoti, kai užsiregistravę naudotojai jas parašo. Taip pat lentelė naudojama administraciniame puslapyje pakoreguoti arba ištrinti komentarus.
- **distances** – „Atstumai tarp miestų“. Ši lentelė turi laukus: miesto pavadinimas (City1), miesto pavadinimas (City2) ir atstumas (Distance) atributus. Ji naudojama pagrindinėje šios internetinės svetainės funkcijoje, kurioje reikia rasti trumpiausią maršrutą pateikiant pilną atstumą bei maršrutą.
- **liked_places** – „Patikusios lankytinos vietos“. Ši lentelė turi laukus: lankytinos vietos ID (Place_id), naudotojo ID (User_id) ir lankytinos vietos pavadinimas (Place_name) atributus. Ji skirta parodyti skaičių naudotojų, kuriems patiko tam tikra lankytina vieta bei išsaugoti „Like“ paspaudimus užsiregistravusių naudotojų arba administratorių.
- **places** – „Lankytinos vietos“. Ši lentelė turi laukus: lankytinos vietos ID (Place_id), lankytinos vietos pavadinimas (Place_name), miesto pavadinimas (City), oficialus internetinis puslapis (Official_page) ir lankytinos vietos nuotrauka (Place_image) atributus. Ji skirta parodyti visas lankytinas vietas paieškos puslapyje, atidaryti pasirinktą lankytiną vietą dėl detalesnės informacijos bei administraciniame puslapyje sukurti, pakoreguoti arba ištrinti lankytinas vietas. Taip pat lentelė naudojama išgauti lankytinas vietas (pavadinimus bei nuotraukas) įvykdžius maršruto paiešką.
- **places_ratings** – „Lankytinų vietų įvertinimai“. Ši lentelė turi laukus: lankytinos vietos ID (Place_id), naudotojo ID (User_id) ir įvertinimas (Rating) atributus. Ji skirta parodyti naudotojų įvertinimų vidurkį ant tam tikros lankytinos vietos, parodyti naudotojo asmeninį įvertinimą bei išsaugoti juos su sąlyga, kad naudotojas yra užsiregistravęs arba administratorius.
- **routes** – „Maršrutai“. Ši lentelė turi laukus: maršruto ID (Route_id) ir pilnas maršrutas (Full_route) atributus. Ji skirta išsaugoti kiekvieną trumpiausią maršrutą, kaip patekti iš vieno miesto į visus likusius. Taip pat lentelė naudojama maršrutų paieškos puslapyje parodyti visų maršrutų informacijos dalį.
- **stops** – „Sustojimai“. Ši lentelė turi laukus: sustojimo ID (Stop_id), miesto pavadinimas (City1), miesto pavadinimas (City2), atstumas tarp miestų (Distance), maršruto ID (Route_id) ir sustojimo numeris (Stop_number) atributus. Ji skirta išgauti kiekvieno pilno maršruto informaciją su atstumais tarp maršrute esančių miestų bei maršruto sustojimų eilės numerius. Tai atliekama jau susiradus norimą maršrutą, kai yra pilno maršrutas ir atstumas, tada per maršruto ID yra surandama likusi informacija. Taip pat lentelė yra naudojama išgauti lankytinas vietas (pavadinimus bei nuotraukas) įvykdžius maršruto paiešką.
- **users** – „Naudotojai“. Ši lentelė turi laukus: naudotojo ID (User_id), vardas (First_name), pavardė (Last_name), slapyvardis (User_name), elektroninio pašto adresas

(Email_address), slaptažodis (Password), naudotojo nuotrauka (User_image) ir administratorius (Admin) atributus. Ji skirta naudotojui užsiregistruoti, prisijungti, parodyti informaciją maršrutų ir lankytinų vietų paieškos puslapiuose. Taip pat lentelė naudojama administraciniame puslapyje pakoreguoti arba ištrinti naudotojus, suteikti administratoriaus teises.

3.3 Lankytinų vietų paieškos puslapis

Į lankytinų vietų aprašymą įeina bendra informacija, leidžianti susipažinti su lankomu objektu – tai būtų pavadinimas, miestas, oficialus internetinis puslapis bei nuotrauka. Kita informacija – komentarai, įvertinimai ir „Like“, yra užsiregistravusių naudotojų pateikiama informacija. Kitos informacijos dėka naudotojai gali pareikšti savo nuomonę, leidžiančią susidaryti tikresnę įspūdį apie turistinį objektą.

Lankytinos vietos internetinėje svetainėje yra objektai, talpinantys informaciją apie objektus, kurie yra visada pasiekiami. Šiame puslapyje yra navigacija tarp puslapio dalių, kuri yra pavaizduota puslapio viršuje, meniu juostoje. Pagrindinės puslapio funkcijos, kuriomis gali naudotis bet koks naudotojas, yra šios:

- Lankytinų vietų rikiavimas pagal komentarų skaičių (Sort by comments), vidutinį įvertinimą (Sort by rating) arba „Like“ kiekį (Sort by likes). Kiekvienas iš rikiavimų veikia taip – naudotojus paspaudus vieną iš mygtukų, aktyvuojasi viena iš lankytinų vietų Controller dalies funkcijų, kurioje krepiamasi į vieną iš Model dalies funkcijų. Tada PHP karkaso pagalba, naudojantis specialiomis funkcijomis, yra užrašoma SQL užklausa, kurios rezultatas yra išrikiuotas įrašų masyvas su lankytinos vietos pavadinimu, miestu, nuotrauka, oficialiu puslapiu bei komentarų skaičiumi, vidutiniu įvertinimu, „Like“ kiekiu, atitinkamai pagal tai, koks rikiavimo mygtukas buvo paspaustas. Turint funkcijos rezultatus, jie grąžinami atgal į Controller dalies funkciją, kuri paduoda informaciją naujam puslapiui arba perkrauna dabartinį, tada jau yra rodomos išrikiuotos lankytinos vietos. SQL užklausa yra pateikiama 6 paveikslėlyje [8], išrūšiuota pagal „Like“ skaičių.
- Lankytinų vietų paieška pagal įvestą norimą reikšmę. Naudotojas į įvesties lauką (Search for a place by it's name) įveda ieškomą tekstą ir nuspaudus mygtuką (Search), aktyvuojasi Controller dalies funkcija, kurioje patikrinama, ar įvestas tekstas nėra tuščias, ir tada tas tekstas kaip parametras perduodamas į Model dalies funkciją. Tada PHP karkaso pagalba, naudojantis specialiomis funkcijomis, yra užrašoma SQL užklausa, kurios rezultatas yra įrašų masyvas su lankytinos vietos pavadinimu, miestu, nuotrauka, oficialiu puslapiu bei komentarų skaičiumi, vidutiniu įvertinimu, „Like“ kiekiu. SQL užklausa tikrina, ar yra lankytinų vietų su pavadinimais, kuriuose būtų įvestas naudotojo tekstas.
- Lankytinų vietų paieška pagal miestą. Naudotojas iš duoto miestų sąrašo (Choose a city to search for places) pasirenka vieną miestą ir nuspaudus mygtuką (Search), aktyvuojasi Controller dalies funkcija, kurioje patikrinama, ar buvo pasirinktas miestas, ir tada pasirinkto miesto reikšmė kaip parametras perduodamas į Model dalies funkciją. Tada PHP karkaso pagalba, naudojantis specialiomis funkcijomis, yra užrašoma SQL užklausa, kurios rezultatas yra įrašų masyvas su lankytinos vietos pavadinimu, miestu, nuotrauka, oficialiu puslapiu bei komentarų skaičiumi, vidutiniu įvertinimu, „Like“ kiekiu. SQL

užklausa tikrina, ar yra lankytinų vietų, kur miestų pavadinimai sutaptų su pasirinktu naudotojo miestu.

- Lankytinos vietos atidarymas. Naudotojas paspaudžia ant lankytinos vietos nuotraukos arba pavadinimo, kas aktyvuoja trumpą Javascript kodą, kuriame paduodamas lankytinos vietos ID ir sudėliojamas pilnas internetinės svetainės adresas, susidedantis iš bazinio URL, Controller dalies funkcijos ir lankytinos vietos ID. Tada yra iškviečiama Controller dalies funkcija, kur pasinaudojus specialia PHP karkaso biblioteka, yra išgaunamas tinklalapio adreso segmentas su lankytinos vietos ID, kuris yra perduodamas Model dalies funkcijai gauti visą tos lankytinos vietos informaciją. Model dalies funkcijos rezultatas perduodamas atgal į Controller dalies funkciją, kurioje yra užkraunamas visiškai kitas puslapis su lankytinos vietos informacija.

```
SELECT A.Place_id, A.Amount_of_likes, B.Average_rating, C.Amount_of_comments, A.Place_name, A.City, A.Place_image
from
  (SELECT A.Place_id, count(A.Place_id) as Amount_of_likes, A.Place_name, A.City, A.Place_image
  FROM places as A
  left join bakalauras.liked_places as B
  on A.Place_id = B.Place_id
  group by A.Place_id) as A

join

  (SELECT A.Place_id, round(avg(Rating), 1) as Average_rating
  FROM bakalauras.places as A
  left join bakalauras.places_ratings as B
  on A.Place_id = B.Place_id
  group by A.Place_id) as B

join

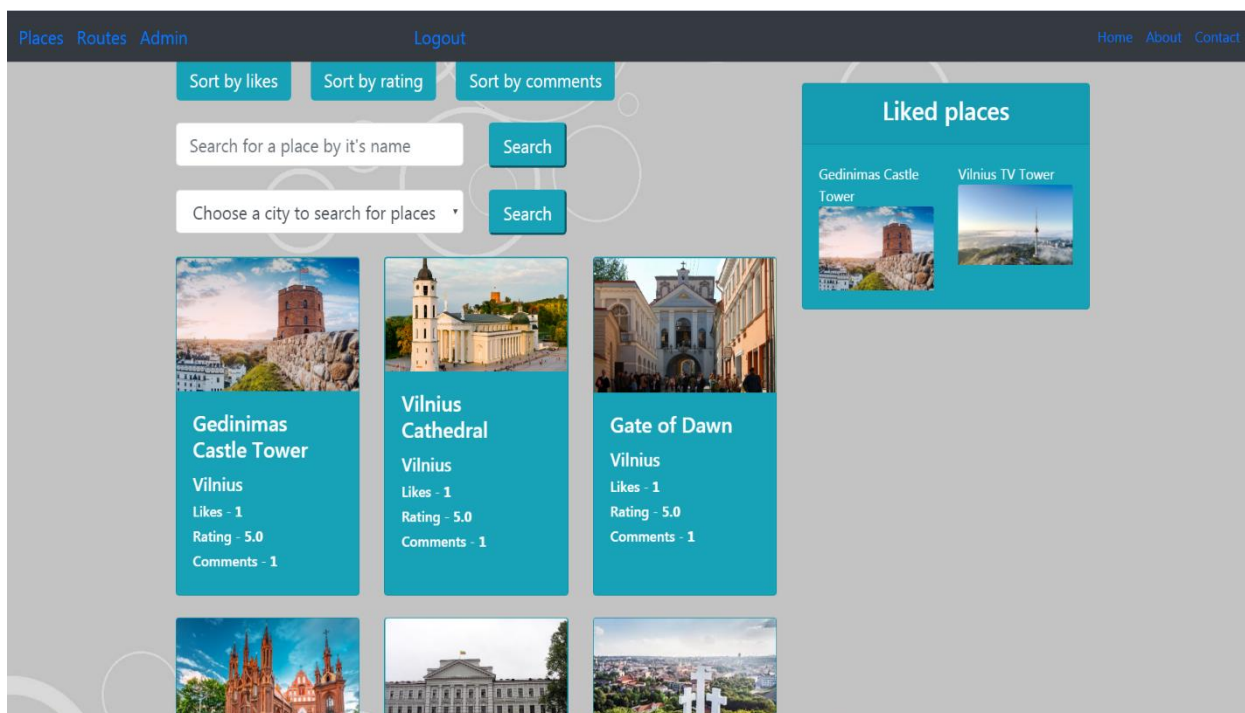
  (SELECT A.Place_id, count(A.Place_id) as Amount_of_comments
  FROM bakalauras.places as A
  left join bakalauras.comments as B
  on A.Place_id = B.Place_id
  group by A.Place_id) as C

on A.Place_id = C.Place_id

order by A.Amount_of_likes desc, A.Place_name asc
```

6 pav. SQL užklausa, skirta gauti išrikiuotas lankytinas vietas

Visas lankytinų vietų paieškos puslapis išdėstytas taip, kad dešiniajame šone matytųsi naudotojo informacija, po ja eitų patikusių lankytinų vietų skiltis. Puslapio viduryje prasideda su nuotraukų karuselė, po ja eitų trys išrūšiavimo mygtukai, tada lankytinų vietų paieškos laukai kartu su paieškos vykdymo mygtukais, bei lankytinų vietų sąrašas, kur kiekvienai vietai yra paskirta atskira kortelė rodyti nuotrauką, pavadinimą, miestą ir kitą informaciją. Puslapio dalis pavaizduota 7 paveikslėlyje.



7 pav. Lankytinų vietų paieškos puslapio dalis informacinėje sistemoje

3.4 Maršrutų paieškos puslapis

Į maršrutų aprašymą įeina informacija, leidžianti detaliai išnagrinėti, pro kokius miestus pasirinktu maršrutu galima pravažiuoti, kokie atstumai tarp kiekvieno sustojimo tarp dviejų miestų bei kokia sustojimų eilės tvarka. Šios informacijos pagalba galima išgauti lankytinas vietas ieškomo maršruto pagal tai, kokie miestai į jį patenka.

Maršrutų internetinėje svetainėje yra objektai, talpinantys informaciją apie objektus, kurie yra visada pasiekiami. Šiame puslapyje yra navigacija tarp puslapio dalių, kuri yra pavaizduota puslapio viršuje, meniu juostoje. Pagrindinė puslapio funkcija, kuria gali naudotis bet koks naudotojas, tai būtų:

- Maršruto paieška pagal miestus. Naudotojas iš duoto miestų sąrašo pasirenka tiek pradžios, tiek pabaigos vietą, ir nuspaudus mygtuką (Search), aktyvuojasi Controller dalies funkcija, kurioje patikrinama, ar buvo pasirinkti miestai, ir tada jų reikšmės kaip parametrai yra perduodami į Model dalies funkciją. Tada PHP karkaso pagalba, naudojantis specialiomis funkcijomis, yra iškviečiama SQL programavimo kalba parašyta funkcija (stored procedure), kuri yra išsaugota tekstiniame faile ir ji pasiama kaip įvestį du parametrus – pradžios ir pabaigos taškus, o rezultate išveda pilną maršrutą (A-B-C-D) ir atstumą. Ši funkcija yra pavaizduota 8 paveikslėlyje ir ji yra realizuota panaudojant Dijkstra grafų algoritmą. Problema ta, kad ši funkcija pateikia tik pilną maršrutą ir atstumą, o norint pateikti informaciją tarp maršrute esančio kiekvieno sustojimo dviejų miestų ir jų atstumų, buvo panaudota kita Model dalies funkcija, gaunanti informaciją apie visus maršrutus ir juose esančius sustojimus. Tada Controller dalis paima abiejų Model dalies funkcijų rezultatus ir juos paduoda naujam arba perkrautam maršrutų puslapiui. Ten pateikiamas ieškotas maršrutas su pilnu maršrutu, atstumu, visais sustojimais tarp dviejų miestų ir jų atstumais. Tai susieta per maršruto ID. Taip pat pateikiamas lankytinų vietų sąrašas, gautas per tai, kokie miestai patenka į pilną maršrutą.

```

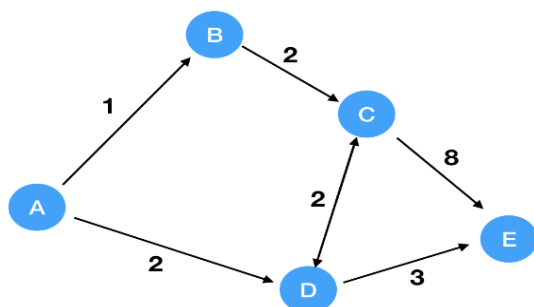
CREATE PROCEDURE bakalauras.FindShortestPath (@starting_point VARCHAR (30), @ending_point VARCHAR (30))
AS
BEGIN
    SET NOCOUNT ON;
    WITH Table1
    AS (SELECT city2,
        CASE
            WHEN City1 IS NULL
            THEN CAST ('-' + ISNULL (City1, city2) + '-' AS VARCHAR (MAX))
            WHEN City1 IS NOT NULL
            THEN CAST ('-' + City1 + '-' + City2 + '-' AS VARCHAR (MAX))
        END FullPath,
        Distance TotalDistance
    FROM bakalauras.distances
    WHERE (City1 = @starting_point)
    UNION ALL
    SELECT a.City2,
        c.FullPath + '-' + a.City2 + '-' FullPath,
        TotalDistance + a.Distance TotDistance
    FROM bakalauras.distances a, Table1 c
    WHERE a.City1 = c.City2
    ),
    Table2
    AS (SELECT *, RANK () OVER (ORDER BY TotalDistance) TheRank
    FROM Table1
    WHERE City2 = @ending_point AND PATHINDEX ('%' + @ending_point + '%', FullPath) > 0)
    SELECT FullPath, TotalDistance
    FROM Table2
    WHERE TheRank = 1;
    SET NOCOUNT OFF
END
GO

```

8 pav. Trumpiausio maršruto radimo funkcija, užrašyta SQL programavimo kalba

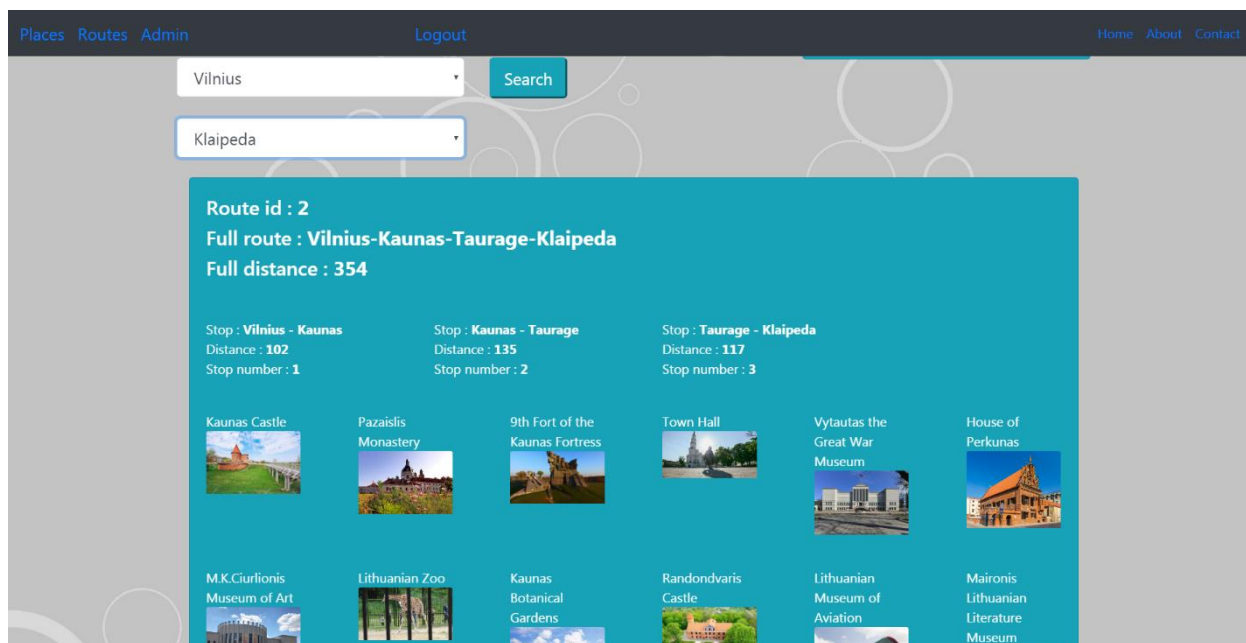
Grafo pavyzdys pateikiamas 9 paveikslėlyje [1]. Dijkstra grafų algoritmas aprašomas taip [2] :

1. Pažymėti visus nematomus mazgus, sudėti juos į rinkinį.
2. Kiekvienam mazgui priskirti preliminarią atstumo vertę – pradiniam mazge nustatyti ją lygią nuliui ir visiems kitiems mazgams lygią begalybei. Pradinį mazgą nustatyti kaip dabartinį.
3. Apsvarstyti visus savo nematomus kaimynus dabartiniam mazgui ir apskaičiuoti jų preliminarinius atstumus per dabartinį mazgą. Palyginti naujai apskaičiuotą bandomąjį atstumą su dabartine priskirtąja verte ir priskirti mažesnę. Pavyzdžiui, jei dabartinis mazgas A yra pažymėtas 6 atstumu, o kraštas, jungiantis jį su kaimynu B, yra 2 ilgio, tada atstumas iki B per A bus $6 + 2 = 8$. Jei B anksčiau buvo pažymėtas atstumas didesnis nei 8, tada pakeiskite jį į 8. Priešingu atveju dabartinė vertė bus išlaikyta.
4. Baigus surasti visus nematomus dabartinio mazgo kaimynus, pažymėti dabartinį mazgą kaip lankomą ir pašalinti jį iš nematomo rinkinio. Aplankytas mazgas niekada nebus patikrintas dar kartą.
5. Jeigu paskirties mazgas buvo pažymėtas kaip lankomas, algoritmas baigtas.
6. Priešingu atveju nematytą mazgą, pažymėtą mažiausiu preliminariniu atstumu, nustatyti kaip naują dabartinį mazgą ir grįžti į 3 veiksmą.



9 pav. Grafo pavyzdys

Visas maršrutų paieškos puslapis išdėstytas taip, kad dešiniajame šone matytųsi naudotojo informacija. Puslapio vidurys prasideda su nuotraukų karuselė, po ja eitų maršruto paieškos laukai kartu su paieškos vykdymo mygtuku, bei maršruto informacija. Ši informacija pateikiama kortelėje, kurioje matosi maršruto ID, pilnas maršrutas ir atstumas, sustojimai tarp dviejų miestų ir jų atstumai, sustojimų numeriai, lankytinos vietos su pavadinimais, miestais ir nuotraukomis. Puslapio dalis pavaizduota 10 paveikslėlyje.



10 pav. Maršrutų paieškos puslapio dalis informacinėje sistemoje

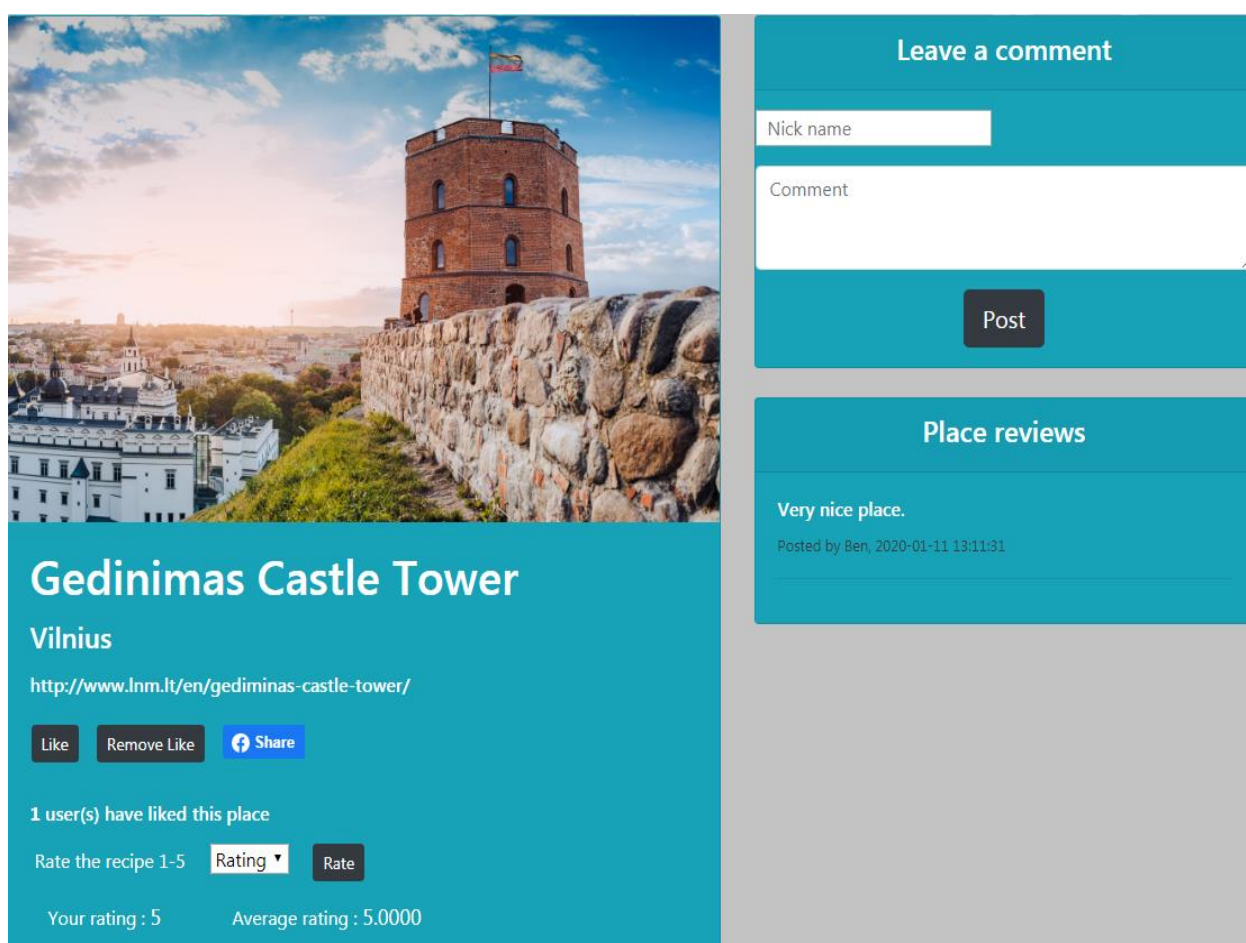
3.5 Lankytinos vietos informacinis puslapis

Lankytinos vietos informaciniame puslapyje pateikiama informacija apie turistinį objektą. Šiame puslapyje yra navigacija tarp pagrindinio puslapio dalių, kuri yra pavaizduota puslapio viršuje, meniu juostoje. Pagrindinės puslapio funkcijos, kuriomis gali naudotis bet koks naudotojas, tai būtų:

- Lankytinos vietos komentavimas. Naudotojas užpildo komentavimo formą, kurioje yra įvesties laukai (Nick name, Comment) ir mygtukas (Post). Tada aktyvuojasi Controller dalies funkcija, kuri patikrina, ar abu įvesties laukai nėra tušti ir ar naudotojas yra prisijungęs, ir perduoda įvesčių reikšmes į Model dalies funkciją. Tada PHP karkaso pagalba, naudojantis specialiomis funkcijomis, yra užrašoma SQL užklausa, kuri išsaugo komentarą duomenų bazėje su šia informacija – komentaro ID, naudotojo slapyvardis, komentaro tekstas, data bei lankytinos vietos ID.
- Lankytinos vietos įvertinimas. Naudotojas pasirenka įvertinimą iš sąrašo nuo vieno iki penkių bei paspaudžia mygtuką (Rate). Tada aktyvuojasi Controller dalies funkcija, kuri patikrina, ar naudotojas buvo prieš tai įvertinęs duotą lankytiną vietą, ir perduoda įvertinimo reikšmę į Model dalies funkciją. Tada PHP karkaso pagalba, naudojantis specialiomis funkcijomis, yra užrašoma SQL užklausa, kuri išsaugo įvertinimą duomenų bazėje su šia informacija – lankytinos vietos ID, naudotojo ID, įvertinimo skaičius.

- Lankytinos vietos „Like“ palikimas. Naudotojas paspaudžia mygtuką (Like). Tada aktyvuojasi Controller dalies funkcija, kuri patikrina, ar naudotojas buvo prieš tai palikęs „Like“ ant duotos lankytinos vietos, ir iškviečia Model dalies funkciją. Tada PHP karkaso pagalba, naudojantis specialiomis funkcijomis, yra užrašoma SQL užklausa, kuri išsaugo „Like“ duomenų bazėje su šia informacija – lankytinos vietos ID, naudotojo ID, lankytinos vietos pavadinimas.

Visas lankytinos vietos informacinis puslapis išdėstytas taip, kad dešiniajame šone matytųsi komentarų įrašymo forma, o po ja eitų visi parašyti naudotojų komentarai būtent tos lankytinos vietos. Puslapio kairiajame šone matosi informacija apie lankytiną vietą – nuotrauka, pavadinimas, miestas ir oficialus puslapis. Tada seka „Like“ mygtukas kartu su tos lankytinos vietos „Like“ skaičiumi, po to seka įvertinimo sąrašas nuo vieno iki penkių kartu su mygtuku (Rate) bei tos lankytinos vietos vidutiniu ir naudotojo įvertinimu, jeigu buvo įvertinta. Puslapio dalis pavaizduota 11 paveikslėlyje.

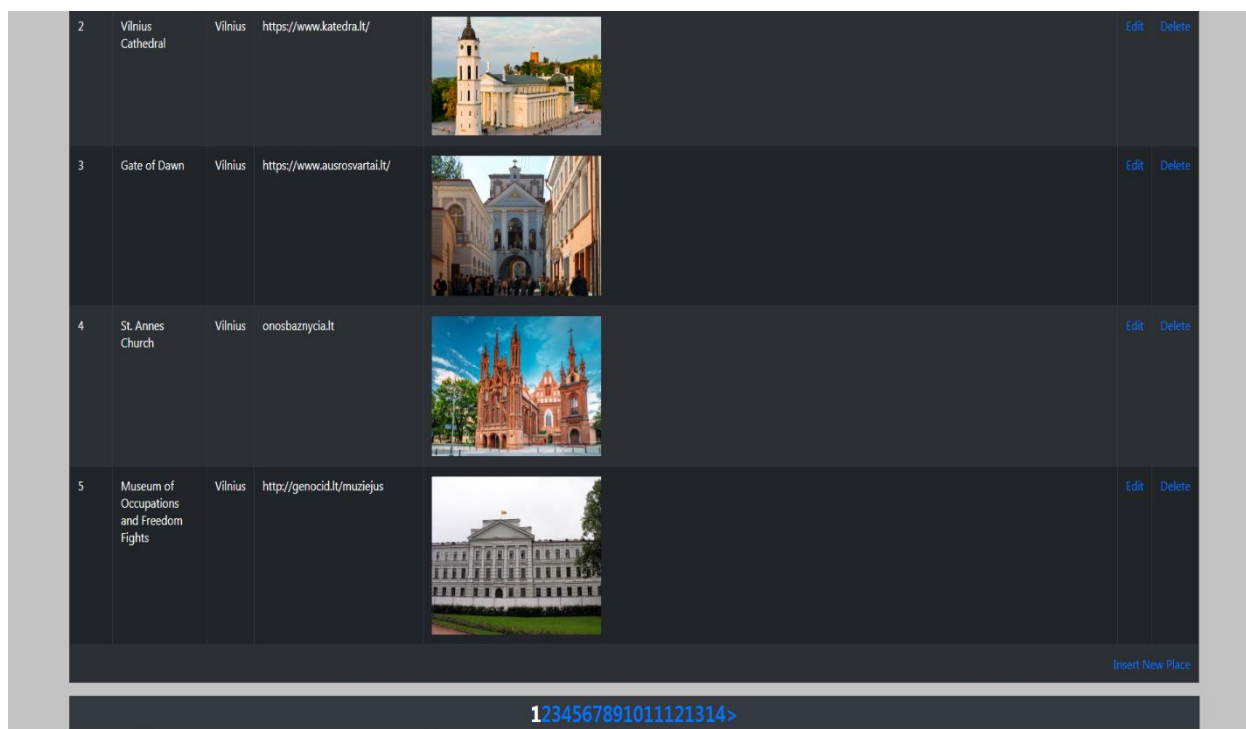


11 pav. Lankytinos vietos informacinio puslapio dalis

3.6 Administracinis puslapis

Ši internetinės svetainės dalis skirta administratoriams ir tik jiems yra pasiekiamą namų (maršrutų arba lankytinųjų vietų paieškos) puslapyje per navigacinio menio „Admin“ laukelį, kuris yra pačiame puslapio viršuje. Paprasti naudotojai nemato navigaciniame meniu „Admin“ laukelio, kuris veikia kaip nuoroda į administracinį puslapį. Joje yra visa informacija apie lankytinas vietas, naudotojus ir komentarus – kiekvienai daliai atskiras puslapis, kuriame administratorius gali

sukurti, ištrinti, pakoreguoti arba peržiūrėti tam tikros dalies įrašus. Puslapio viršuje yra navigacinis meniu, kuriame galima patekti į namų puslapį arba kitas administracines dalis, taip pat yra atliktas puslapiavimas, kai yra užkraunamas tam tikras kiekis įrašų vienu metu. Taip siekiama ženkliai sutrumpinti puslapio užkrovimo laiką. Šio puslapio vaizdas rodomas 12 paveikslėlyje.



12 pav. Administracinis puslapis

Administratorius puslapyje turi tris pagrindines funkcijas, kuriomis gali naudotis – tai įrašų ištrinimas, sukūrimas ir koregavimas. Kad ištrinti įrašą administratorius turi paspausti mygtuką (Delete), tada iššoka lentelė su pranešimu, ar tikrai norite ištrinti pasirinktą įrašą, ir atitinkamai pasirinkus įrašas yra ištrinamas arba neištrinamas. Kad sukurti įrašą administratorius turi paspausti mygtuką (Insert New Record), tada patenkama į naują puslapį, kuriame reikia užpildyti formos laukus su norima informacija. Kai kurie įvesties laukai turi validaciją, kad išvengtų vienodų įrašų duomenų bazėje. Tada paspaudžiamas mygtukas (Create) ir įrašas yra sukuriamas. Kad pakoreguoti įrašą administratorius turi paspausti mygtuką (Edit), tada patenkama į naują puslapį, kuriame yra formos laukai su dabartine informacija. Norimi laukai užpildomi naujomis reikšmėmis. Kai kurie įvesties laukai turi validaciją, kad išvengtų vienodų įrašų duomenų bazėje. Tada paspaudžiamas mygtukas (Update) ir pakoreguotas įrašas yra išsaugomas. Įrašo pakoregavimo forma parodyta 13 paveikslėlyje.

Edit the place

Enter the place name	Gediminas Castle Tower
Select the city	Vilnius
Enter the official page	http://www.lnm.lt/en/gediminas-castle-tow
Select the place image	Pasirinkti failą Nepasirinktas joks failas

Current image



13 pav. Įrašo pakoregavimo forma

Išvados ir rekomendacijos

Darbo metu buvo sukurta informacinė sistema su jai skirtais administruoti įrankiais, kuriuose skirtingų rūšių naudotojai gali atlikti jiems leistinus veiksmus - lankytinų vietų peržiūrėjimas, paieška bei filtravimas pagal miestus arba įvestą norimą pavadinimą, maršrutų radimas pagal pradinį ir galinį maršruto taškus (miestus).

Išvados:

1. Internetinėje svetainėje susiradus maršrutą pateikiama pagrindinė naudotojui reikalinga informacija, kuri galėtų būti ir detalesnė, kas pareikalautų daugiau funkcijų sukūrimo.
2. Informaciniame puslapyje naudotojas labai lengvai gali susidaryti įspūdį apie lankytinas vietas peržiūrėjęs komentarus, įvertinimus, kas padėtų lengviau apsispręsti kokias lankytinas vietas vertėtų apžiūrėti keliaujant ieškotu maršrutu.
3. Naudotojai turi platų pasirinkimą, kaip jie gali ieškoti lankytinų vietų pagal tam tikrus kriterijus, kas leidžia naudotojui rasti būtent to, ko jis ieško.
4. Nemaža dalis sistemoje rodomos informacijos priklauso nuo administratoriaus veiksmų – objektų sukūrimų, pakeitimų arba ištrynimų, kas parodo jo svarbų vaidmenį šiame tinklalapyje.

Rekomendacijos:

1. Informacinėje sistemoje apie lankytinas vietas pateikti papildomą informaciją, kuri tikrai praverstų naudotojams, ir tai galėtų būti darbo laikai ar bilietų kainos.
2. Internetiniame puslapyje į lankytinų vietų sąrašą pridėti maitinimo įstaigas, viešbučius su tikslu tiems naudotojams, kurių maršruto kelionė užtruktų bent porą dienų.
3. Sukurti prenumeravimo funkciją, kad užsiregistravę naudotojai elektroniniu paštu galėtų gauti pranešimus apie naujai įdėtas lankytinas vietas.
4. Įdiegti naudotojams funkciją, kuri leistų jiems išsisaugoti ieškotus maršrutus bei pateiktų jų sąrašą atskiroje skiltyje.
5. Maršruto paieškos rezultate papildomai pateikti ir Lietuvos žemėlapy, kuriame būtų pažymėta maršruto pradžia, pabaiga bei keliavimo trajektorija.
6. Implementuoti maršruto paieškoje funkciją, kuri leistų papildomai pasirinkti į maršrutą įeinančių miestų skaičių arba rodomų lankytinų vietų kiekį.
7. Sukurti lankytinų vietų paieškoje funkciją, kad naudotojams būtų galima susirasti objektus pagal kategorijas, tokias kaip istoriniai paminklai ar muziejai.

Literatūros šaltiniai

- [1] Grafo pavyzdys. Google images.
- [2] Dijkstra's Graph Algorithm. URL: <https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/>
- [3] How the Bootstrap 4 Grid Works. URL: <https://uxplanet.org/how-the-bootstrap-4-grid-works-a1b04703a3b7>
- [4] CodeIgniter Documentation. URL: https://codeigniter.com/user_guide/general/index.html
- [5] Traveling Salesman Problem. URL: https://vc.bridgew.edu/undergrad_rev/vol7/iss1/17/
- [6] Lietuvos lankytinos vietos. URL: <https://www.makalius.lt/kelioniu-ispudziai/lietuva-lankytinos-vietos/>
- [7] Essential Travel Website Features. URL: <https://colorwhistle.com/travel-website-features/>
- [8] Advanced SQL. URL: <https://study.com/academy/lesson/advanced-sql-subqueries-use-examples.html>