



Kauno technologijos universitetas

Informatikos fakultetas

Restoranų maisto likučių skelbimai labdaros grupėms

Baigiamasis bakalauro studijų projektas

Eimontas Dambrauskas

Projekto autorius

Lekt. Voldemaras Žitkus

Vadovas

Kaunas, 2020



Kauno technologijos universitetas

Informatikos fakultetas

Restoranų maisto likučių skelbimai labdaros grupėms

Baigiamasis bakalauro studijų projektas

Programų sistemos (612I30002)

Eimontas Dambrauskas

Projekto autorius

Lekt. Voldemaras Žitkus

Vadovas

Justas Šalkevičius

Recenzentas

Kaunas, 2020



Kauno technologijos universitetas

Informatikos fakultetas

Eimontas Dambrauskas

Restoranų maisto likučių skelbimai labdaros grupėms

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Eimonto Dambrausko, baigiamasis projektas tema „Restoranų maisto likučių skelbimai labdaros grupėms“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

Dambrauskas, Eimontas. Restoranų maisto likučių skelbimai labdaros grupėms. Bakalauro studijų baigiamasis projektas / vadovas lekt. Voldemaras Žitkus; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos mokslai, Programų sistemos.

Reikšminiai žodžiai: maisto likučiai, labdaros grupės, restoranai.

Kaunas, 2020. 43 p.

Santrauka

Ataskaitoje pristatoma informacinė sistema, skirta restoranų maisto dalinimuisi su labdaros grupėmis skelbimų pagalba. Įvade nustatomas programos kūrimo planas, bei temos pasirinkimo priežastis. Analizės dalyje apžvelgiami konkurentai bei projekto kūrimo priemonės. Projekto dalyje nagrinėjami sistemos funkciniai bei nefunkciniai reikalavimai, naudojamos funkcijos, kurios išdėstytos UML diagramomis. Šioje dalyje aprašomi pasirinkti projektavimo sprendimai, bei įvairūs dizaino maketai. Testavimo dalyje aprašomi pasirinkti testavimo būdai bei priemonės. Tolesnėje šio skyriaus dalyje detalios aprašomi gauti rezultatai bei šių rezultatų svarba projektui. Sekančioje dalyje – dokumentacijoje, aprašoma visa vartotojui reikalinga informacija, norint išbandyti sukurta projektą. Darbo pabaigoje aprašomos išvados, bei gauti rezultatai.

Dambrauskas, Eimontas. Adverts of Restaurant Leftovers for Charity Groups. Bachelor's Final Degree Project / supervisor lect. Voldemaras Žitkus. Informatics Faculty, Kaunas University of Technology.

Study field and area (study field group): Computer Sciences, Software Systems.

Keywords: adverts, restaurant leftovers, charity groups.

Kaunas, 2020. 43 p.

Summary

The report presents an information system for sharing restaurant food with charities through ads. The introduction sets out the program development plan and the reasons for choosing the topic. The analysis part reviews the competitors and project development tools. The part of the project deals with the functional and non-functional requirements of the system, the functions used, which are set out in UML diagrams. This section describes the selected design solutions and various design layouts. The testing section describes the selected testing methods and tools. The following section of this chapter describes in detail the outcome obtained and the relevance of these results to the project. The next part - the documentation, describes all the information required by the user to test and use the created project. At the end of the work, the conclusions and the obtained results are described.

Turinys

Lentelių sąrašas.....	7
Paveikslų sąrašas.....	8
Santrumpų ir terminų sąrašas.....	9
Įvadas	10
1. Analizė	11
1.1. Techninis pasiūlymas	11
1.1.1. Sistemos apibrėžimas	11
1.1.2. Bendras veiklos tikslas	11
1.1.3. Sistemos pagrįstumas	11
1.1.4. Konkurencija rinkoje.....	11
1.1.5. Prototipai ir pagalbinė informacija.....	12
1.1.6. Ištekiai, reikalingi sistemai sukurti	12
1.2. Galimybių analizė	13
1.2.1. Techninės galimybės	13
1.2.2. Vartotojų pasiruošimo analizė	13
2. Projektas.....	14
2.1. Reikalavimų specifikacija.....	14
2.1.1. Komercinė specifikacija	14
2.1.2. Sistemos funkcijos.....	14
2.1.3. Vartotojo sąsajos specifikacija.....	21
2.1.4. Realizacijai keliami reikalavimai	25
2.1.5. Techninė specifikacija	25
2.2. Projektavimo metodai.....	25
2.2.1. Projektavimo valdymas ir eiga.....	25
2.2.2. Projektavimo technologija	26
2.2.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinė sistemos	26
2.3. Sistemos projektas.....	26
2.3.1. Statinis sistemos vaizdas	26
2.3.2. Dinaminis sistemos vaizdas	29
3. Testavimas	34
3.1. Testavimo planas.....	34
3.2. Testavimo kriterijai	34
3.3. Integracinis testavimas	34
3.4. Vartotojo sąsajos testavimas	35
4. Dokumentacija naudotojui	37
4.1. Apibendrintas sistemos galimybių aprašymas.....	37
4.2. Vartotojo vadovas	37
4.3. Diegimo vadovas.....	41
Rezultatai ir išvados.....	42
Literatūros sąrašas.....	43

Lentelių sąrašas

1 lentelė Konkurentų analizė.....	12
2 lentelė Autentifikacijos testavimo scenarijus.	35
3 lentelė GetUser testavimo scenarijus.	35
4 lentelė PutUser testavimo scenarijus.....	35
5 lentelė ConfirmUser testavimo scenarijus.....	36
6 lentelė DeleteUser testavimo scenarijus.....	36
7 lentelė ConfirmOrder testavimo scenarijus.....	36

Paveikslų sąrašas

1 pav. Sistemos panaudojimo atvejų diagrama.	14
2 pav. Veiklos diagrama (Registruoti vartotoją).....	15
3 pav. Veiklos diagrama (Prisijungti).	15
4 pav. Veiklos diagrama (Atsijungti).....	16
5 pav. Veiklos diagrama (Redaguoti paskyrą).....	16
6 pav. Veiklos diagrama (Šalinti paskyrą).	17
7 pav. Veiklos diagrama (Patvirtinti vartotojo registraciją).	17
8 pav. Veiklos diagrama (Skelbimų filtravimas).....	18
9 pav. Veiklos diagrama (Peržiūrėti skelbimą).	18
10 pav. Veiklos diagrama (Šalinti skelbimą).	19
11 pav. Veiklos diagrama (Redaguoti skelbimą).....	20
12 pav. Veiklos diagrama (Kurti skelbimą).	21
13 pav. Prisijungimo lango maketas.	22
14 pav. Pagrindinio lango maketas.	22
15 pav. „Mano užsakymai“ lango maketas.	23
16 pav. Skelbimo lango maketas.	23
17 pav. Paskyros redagavimo lango maketas.	24
18 pav. Skelbimo redagavimo lango maketas.	24
19 pav. Iteracinis projektavimo modelis.	25
20 pav. Diegimo diagrama.	27
21 pav. Paketų diagrama.	27
22 pav. Vartotojo klasės diagrama.....	28
23 pav. Skelbimo klasės diagrama.....	28
24 pav. Maisto klasės diagrama.....	29
25 pav. Išrašo klasės diagrama.	29
26 pav. Sekų diagrama (Atsijungti).	30
27 pav. Sekų diagrama (patvirtinti vartotojo registraciją).....	30
28 pav. Sekų diagrama (Peržiūrėti skelbimą).....	31
29 pav. Sekų diagrama (Prisijungimas).	31
30 pav. Sekų diagrama (Paskyros redagavimas).	31
31 pav. Sekų diagrama (Skelbimo redagavimas).	32
32 pav. Sekų diagrama (Registracija).	32
33 pav. Sekų diagrama (Skelbimo šalinimas).	33
34 pav. Sekų diagrama (Skelbimo kūrimas).	33
35 pav. Testavimo aplinka.....	34
36 pav. Postman autentifikacijos scenarijus.....	35
37 pav. Registracijos puslapis.	37
38 pav. Prisijungimo puslapis.....	38
39 pav. Pagrindinis puslapis.	38
40 pav. Skelbimo puslapis.....	39
41 pav. Vartotojo sukurtų skelbimų puslapis.	39
42 pav. Skelbimo kūrimo puslapis.....	40
43 pav. Vartotojo paskyros redagavimo puslapis.	40

Santrumpų ir terminų sąrašas

- SSMS – (angl. *SQL Server Management studio*) - aplikacija skirta SQL tipo serverių valdymui.
- SQL – (angl. *Structured Query Language*.) - programavimo kalba, skirta aprašyti duomenis ir manipuluoti jais reliacinių duomenų bazių valdymo sistemose.
- UML – (angl. *Unified modeling language*) – grafinė kalba, skirta objektinių programų projektavimui.
- API – (angl. *Application program interface*) – programų sąsaja, skirta pasiekti papildomą funkcionalumą iš išorinių šaltinių, tokių kaip biblioteka ar programa.
- JWT – (angl. *JSON Web Token*) - interneto standartas, skirtas kurti JSON pagrįstus prieigos raktus, kurie patvirtina vartotojo autentiškumą.
- JSON – (angl. *JavaScript Object Notation*) - failo formatas, perduodantis duomenų objektus, sudarytus iš atributo ir reikšmės porų.
- NPM – (angl. *Node Packet Manager*) - *JavaScript* programavimo kalbos paketų tvarkyklė.
- SDK – (angl. *Software Development Kit*) - programinės įrangos kūrimo priemonių, esančių viename įdiegiamame pakete, rinkinys.

Ivadas

Didėjant žmonių populiacijai ir nepakankamai apšviečiant visuomenę apie maisto stokos problemą, vis aktualesne bėda tampa maisto švaistymas. Užsienyje yra keletas projektų, kurie bando tvarkyti šią problemą pasitelkdami restoranų ir panašių vietų pagalbą, surenkant jų nepanaudotus ingredientus, bei maistą ir išdalinant juos maisto neturintiems. Lietuvoje apie maisto saugojimą niekas dar taip negalvoja. Iš to ir kilo mintis plėsti šią idėją.

Sistemos kūrimas abstrakčiai suskaidytas į 7 dalis:

1. Konkurentų analizė;
2. Sistemos projektavimas;
3. Abstraktus sistemos sukūrimas;
4. Sukuriamas kliento ir serverio sąsaja;
5. Sistemos apjungimas su Google žemėlapiais;
6. Sistemos testavimas, bei klaidų taisymas;
7. Dokumentacijos paruošimas.

Darbas prasidės nuo konkurentų analizės, kadangi reikia atsižvelgti į užsienio pavyzdžius, bandyti pataisyti jų klaidas, pamatyti vietas, kuriose būtų galima įterpti naujovių. Sekantis žingsnis būtų projektavimas, kuriame svarbūs kiekvienos klasės reikalingi laukai, bei jų sąryšiai. Suformavus sistemos architektūrą, reikia apgalvoti patrauklią ir patogią sistemos išvaizdą. Sistemos patogumui, bus prijungti Google žemėlapiai, kurie palengvins maisto atsiėmimą bei kelią iki restorano radimą. Sudėjus šias pagrindines funkcijas, bus galima pažiūrėti ar kažko netrūksta, ar nereikia papildimo. Sekantis etapas – testavimas, bei klaidų taisymas, kurie be abejo reikalingi, norint užtikrinti sklandžią sistemos veiklą ateičiai. Sėkmingai užbaigus testavimo etapą, liks paruošti dokumentaciją, kurioje bus pateikti sistemos veikimo principai, API specifikacija, naudojimo vadovas. Po to dokumentacijoje seks rezultatų apibendrinimas ir išvados.

1. Analizė

1.1. Techninis pasiūlymas

1.1.1. Sistemos apibrėžimas

Restoranų ir labdaros grupių maisto dalinimosi informacinė sistema – skirta restoranų maisto, bei ingredientų atidavimui labdaroms grupėms. Taip būtų skatinamas maisto taupymas, bei restoranų reputacijos didinimas. Sistemą sudarys vartotojų ir skelbimų posistemės. Vartotojų posistemėi priklausys vartotojų registracija, prisijungimas, bei jų valdymas. Skelbimų posistemė bus atsakinga už skelbimų kūrimą, šalinimą bei redagavimą.

1.1.2. Bendras veiklos tikslas

Bendras veiklos tikslas yra padėti kovoti su globalaus masto problema - maisto švaistymu, kuri šiais laikais yra labai aktuali ir didėjanti. Jeigu kovoti nepavyks, manau jog žmonės vistiek bent susimastys apie šią bėdą. Be šios problemos taip pat yra svarbu išbandyti dabar populiarias kūrimo priemones, kadangi React ir NodeJs yra gan populiarios ir tai suteiks daug naujų žinių bei patirties ateičiai.

1.1.3. Sistemos pagrįstumas

Didėjant žmonių skaičiui pasaulyje, kasdien neišvengiamai kyla maisto svarba bei jo poreikis. Pasaulyje esama panašių sistemų, kurios bando šią problemą siek tiek gerinti, tačiau mūsų šalyje, apie tai niekas per daug nekalba, todėl šios sistemos kūrimas paskatintų žmones su maistu elgtis atsakingiau, juo pasidalinant. Taipogi, maistą dalinantys restoranai, pagerintų savo reputaciją ir sulauktų šios tokios reklamos.

1.1.4. Konkurencija rinkoje

Šiuo metu egzistuoja sistemos, skirtos maisto likučių padalinimui kitose šalyse, tačiau jos labiau orientuotos į bendradarbiavimą ir maisto išvežiojimą, bei pinigų aukojimą. Tačiau mano kuriamai sistemai analogų nėra, kadangi ji būtų skirta parodyti, kurie restoranai turi atliekamo maisto (skelbimai). Artimiausi konkurentai veikimo principu yra skelbimų portalai.

Pirmasis konkurentas – *rescuingleftovercuisine.org* yra skirtas rinkti lėšas maisto produktams, bei transportui gabenant maistą iš tų, kurie jį atiduoda. Pats tinklalapis nėra skelbimų stiliaus, čia siekiama žmogų pritraukti statistikomis, kurios pateiktos pagrindiniame puslapyje. Norintys pagelbėti neturintiems maisto, gali arba suteikti vienkartinę piniginę paramą, arba pasidaryti nuolatinį abonimentą, kuris kas mėnesį nuimtų pasirinktą pinigų sumą. Prisijungti galima su Google paskyra, arba susikurti jų tinklalapio paskyrą, kurią kuriantis nėra jokių apribojimų išskyrus el. Paštui. Bendravimo funkcijos nėra, galima parašyti jiems laišką, paliekant savo el. Pašto adresą.

Sekantis konkurentas būtų skelbimų portalas *skelbiu.lt*. Nors šios svetainės įdėja skiriasi nuo kuriamos sistemos, tačiau principas panašus – žmonės deda skelbimus, tam, kad informuotų kitus, jog jie kažką atiduoda/parduoda. Šiame tinklalapyje galima prisijungti su Google ir su Facebook paskyromis, bei susikurti jų svetainės profilį. Registracija labai paprasta, užtenka suvesti savo el. Paštą, bei susigalvotą slaptažodį. Jokių patvirtinimo nuorodų ar panašių saugos funkcijų nėra. Svetainėje kaip ir kuriamoje sistemoje yra atskiras puslapis savo skelbimams peržiūrėti. Be komentarų po skelbimais, bendravimas šios sistemos viduje taipogi negalimas. Skelbimus galima rūšiuoti, bei filtruoti pagal kelis kriterijus (kaina, miestas, data).

Trečiasis konkurentas yra panašaus tipo į pirmąjį, tačiau jie papildomai dar ir ieško savanorių, vežiojančių maistą, bei dalyvaujančių jų rengiamuose susirinkimuose. Ši sistema pranašesnė tuo, jog yra sujungta su programėle telefone. Svetainėje taipogi pabrėžiami iki šiol jų pasiekti nuopelnai, išsaugoto maisto kiekis ir panašios statistikos. Registracija taipogi labai paprasta, užtenka vardo, pavardės ir el. Pašto adreso, beje kuriame taipogi įdiegta validacija. Užsiregistravus prisijungti prie sistemos negalima, į prieš tai nurodytą el. Paštą tiesiog yra siunčiami naujienlaiškiai. Bendravimas sistemoje taipogi negalimas, viskas vykdoma telefonu arba el. Pašto laiškais.

1 lentelė Konkurentų analizė

Lyginimo kriterijai	https://www.rescuingleftovercuisine.org/	https://www.skelbiu.lt/	https://rescuefood.ca/	Kuriama sistema
Paskyros susikūrimas	Taip	Taip	Ne	Taip
Skelbimo talpinimas	Ne	Taip	Ne	Taip
Skelbimų filtravimas	Ne	Taip	Ne	Iš dalies
Google Maps implementacija	Ne	Iš dalies	Ne	Taip
Prisitaikanti aplinka	Taip	Taip	Taip	Taip

Kriterijai buvo pasirinkti peržiūrėjus kelias panašias svetaines, kurių pagrindinės funkcijos buvo paminėtos lentelėje. Funkcijų be abejo yra daugiau, tačiau buvo parinktos pagrindinės funkcijos, kurios yra svarbios šiame projekte.

1.1.5. Prototipai ir pagalbinių informacija

Sistemos kūrimui daugiausia pagalbos gaunama iš internetinių šaltinių – forumų, vaizdo įrašų, bei mokomųjų straipsnių. Buvo panaudoti *Google* siūlomi *Google Maps*, bei *Google Geocoding API*. *Google Maps* platforma yra API ir SDK rinkinys kuris yra valdomas su *Google Cloud Platform Console* (1). *Google Geocoding* yra adresų konvertavimas į geografines koordinatas, kurios yra naudojamos žymeklių pažymėjimui ir žemėlapių pozicijos priartinimui prie duoto adreso (2). Vartotojo sąsajai kurti bus pasinaudota *Visual Studio Code* įrankiu, su *NodeJs* vykdymo aplinka. Serverio pusei naudojama *Visual Studio 2019* aplinka. *Microsoft SQL* duomenų bazei atvaizduoti buvo pasitelkta SSMS 18. SSMS yra integruota aplinka, skirta SQL infrastruktūrai pasiekti, valdyti ir kurti (3).

1.1.6. Išteklių, reikalingi sistemai sukurti

Restoranų ir labdaros grupių sistemai sukurti užtenka vieno žmogaus ir 2-3 mėnesių darbo (300h), tačiau pabaigus kūrimą ir paleidus sistemą, būtų reikalingi bent dar 2 žmonės. Vienas prižiūrėti sistemai, kitas valdyti bei bendrauti su vartotojais, kuriems iškilo bėdų, o trečias žmogus padėtų prieš tai minėtiems darbuotojams, bei galvotų apie sistemos praplėtimo galimybes, naujas funkcijas. Kodo apimties nustatyti negalima, kadangi projekto funkcijas galima aprašyti begalę skirtingų būdų, kurių kodo apimtis irgi skiriasi.

1.2. Galimybių analizė

1.2.1. Techninės galimybės

Sistamai realizuoti bus pasitelkta viena populiariausių *Javascript* vykdymo aplinkų – *NodeJS*. *NodeJs* yra veikimo aplinka, sukurta *V8 Javascript* varikliui (4). Vartotojo sąsajai sukurti bus pasitelkta *React* biblioteka, kadangi ji suteikia daugiau laisvės, bei pasirinkimo, tiek išvaizdoje, tiek pačiame sistemos veikime. Dauguma dizaino dalių yra implementuota iš *Material-ui* bibliotekos. Serverio daliai (*.Net Core 3.1.4*) naudojama *C#* programavimo kalba, *Visual studio 2019* aplinkoje. *.Net Core* yra multiplatforminis, aukšto funkcionalumo, atviro kodo karkasas, skirtas modernių internetinių programų kūrimui (5). Duomenų bazei peržiūrėti bei valdyti naudojama *SSMS 18*.

1.2.2. Vartotojų pasiruošimo analizė

Sukurta sistema naudotis bus paprasta, kadangi ji bus panaši į kitus skelbimų tinklalapius, kurie šiais laikais yra daug kam žinomi bei išbandyti. Vartotojo sąsają bus stengiamasi supaprastinti kuo labiau, kad būtų suprantama, kaip naudotis sistema prisijungus pirmą kartą. Paprastumas svarbus ir dėl to, kad žmonės nepatingėtų sukurti skelbimą, kadangi jie piniginės naudos iš paramos grupių negaus. Pagrindinis dalykas, į kurį turėtų atsižvelgti naujas klientas, kad jo paskyrą turi patvirtinti už tai atsakingas svetainės darbuotojas, dėl to, po registracijos jis iškart prisijungti negalės.

2. Projektas

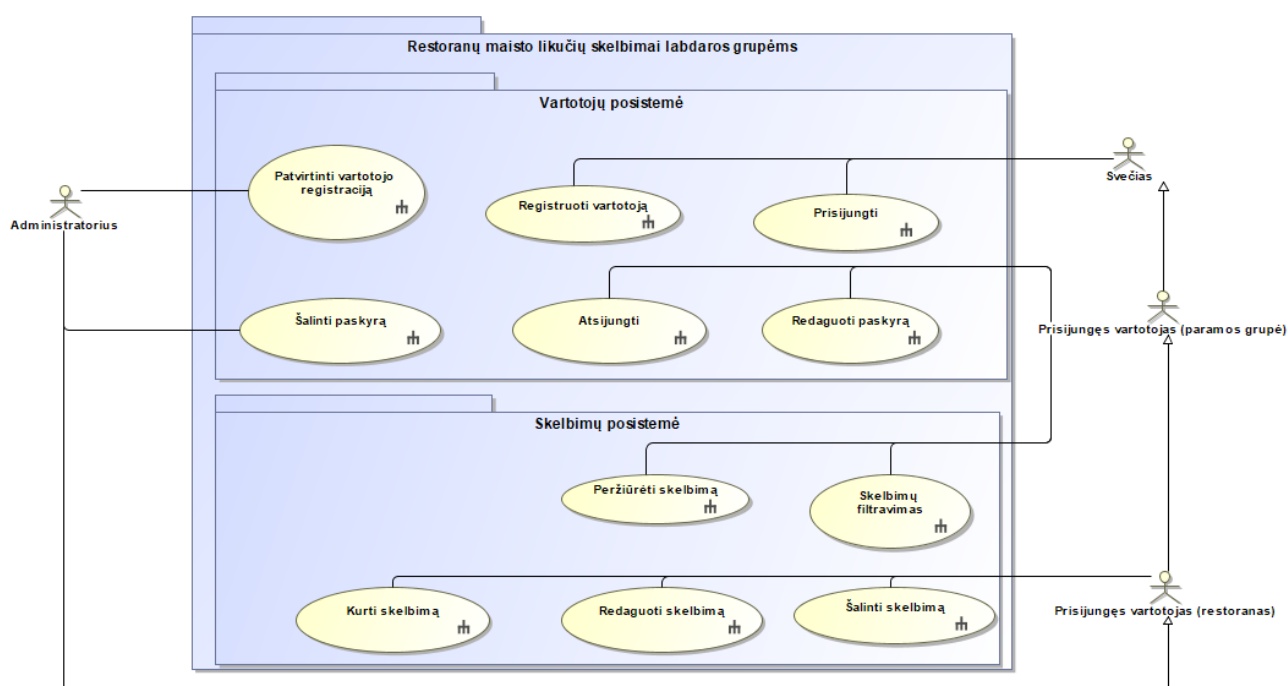
2.1. Reikalavimų specifikacija

2.1.1. Komerčinė specifikacija

Šis projektas yra mano asmeninė įdėja, kuri užsakovo neturi. Kūrime dalyvauja tik vienas asmuo. Vartotojai bus restoranai, bei labdaros grupės, jokių pašalinių asmenų nebus galima aptarnauti. Ateityje galvojama įtraukti savanorius, kurie gabentų maistą iš restoranų į labdaros grupių patalpas. Sistema bus skirta gan nedideliui vartotojų grupei. Projektą planuojama pabaigti iki gegužės 18d.

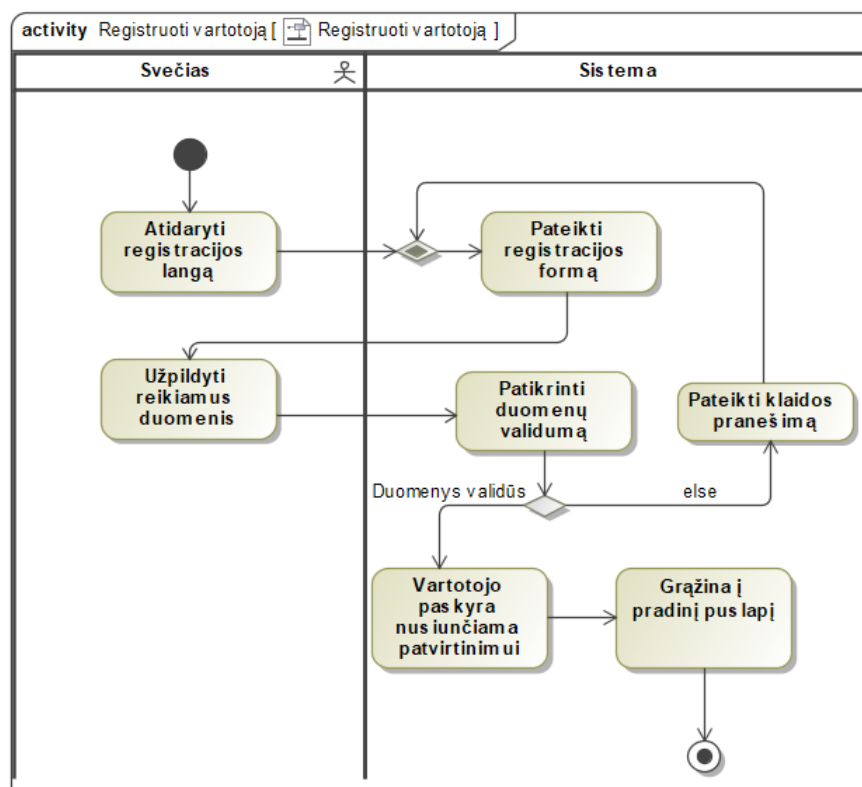
2.1.2. Sistemos funkcijos

Sistemos funkcijoms bei jų panaudojimui, geriausiai tinka panaudos atvejų ir veiklos diagramos (1 pav. - 12 pav.).



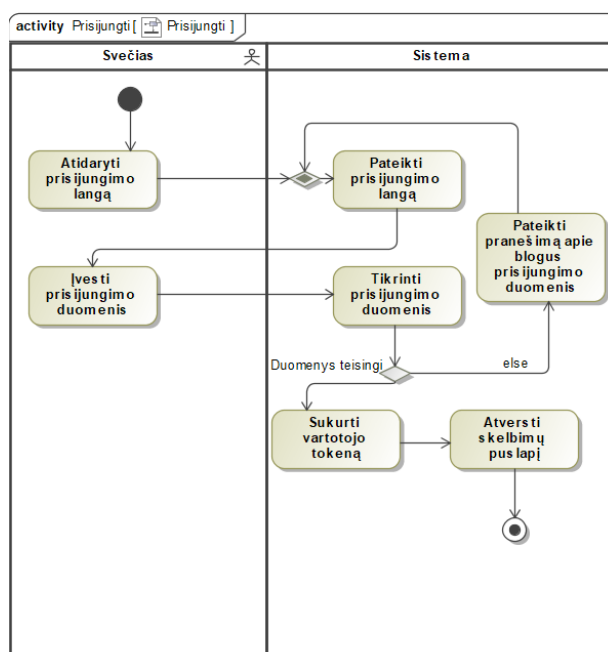
1 pav. Sistemos panaudojimo atvejų diagrama.

Sistemą sudaro keturių tipų vartotojai: Svečias, paramos grupė, restoranas, bei administratorius. Svečio galimybės mažiausios – tik prisijungti ir registruoti vartotoją. Antroje pakopoje yra paramos grupė, kuri įgauna galimybių valdyti savo paskyrą, bei peržiūrėti ir užsisakyti skelbimus. Restorano lygmenyje prisideda skelbimų valdymo funkcijos. Administratorius turi galimybę ištrinti arba patvirtinti norintį užsiregistruoti svečią.



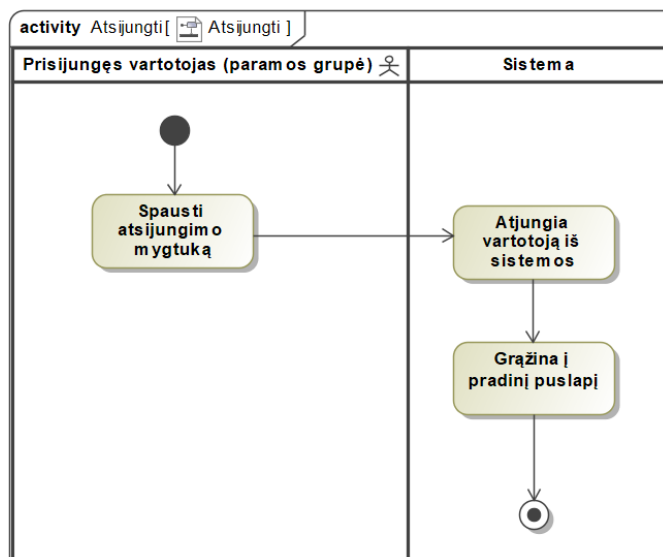
2 pav. Veiklos diagrama (Registruoti vartotoją).

Registracijos atvejis (2 pav.) gana paprastas – vartotojas turi užpildyti jam duotus laukus ir spausti registracijos mygtuką, kuris inicijuoja sistemoje duomenų tikrinimą, kurio rezultatai nusprendžia tolimesnę vartotojo veiksmų eigą. Klaidos pranešimas gali atsirasti dėl paliktų tuščių laukų arba dėl nekorektiškos el. Pašto sudėties. Po šio etapo, vartotojui lieka laukti, kol administratorius ją patvirtins arba atmes.



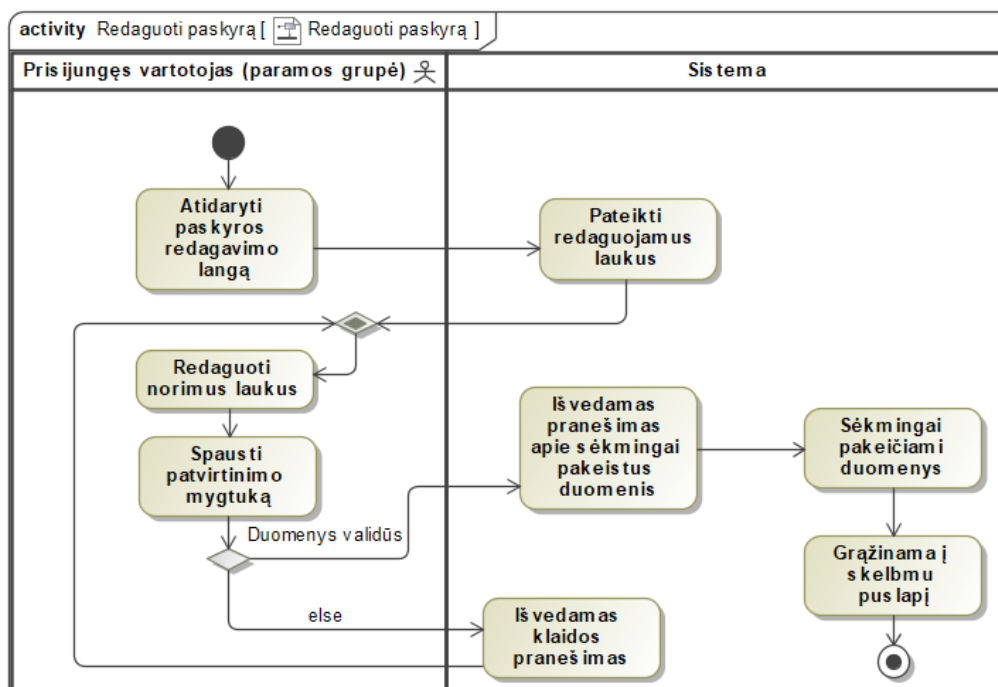
3 pav. Veiklos diagrama (Prisijungti).

Prisijungimo atvejis (3 pav.) irgi nėra sudėtingas. Viskas ko reikia – suvesti savo el. Pašto adresą bei slaptažodį, kurie buvo įvesti registracijos etape. Po prisijungimo duomenų pateikimo, laukiama žinutės iš sistemos, kuri patikrina ar duomenys teisingi. Neteisingai suvedus duomenis, bus pateikiamas klaidos pranešimas, kitu atveju bus sukuriamas vartotojo tokenas, kuris rodytų, kad vartotojas yra prisijungęs ir vartotojas bus nukreiptas į pagrindinį puslapį.



4 pav. Veiklos diagrama (Atsijungti).

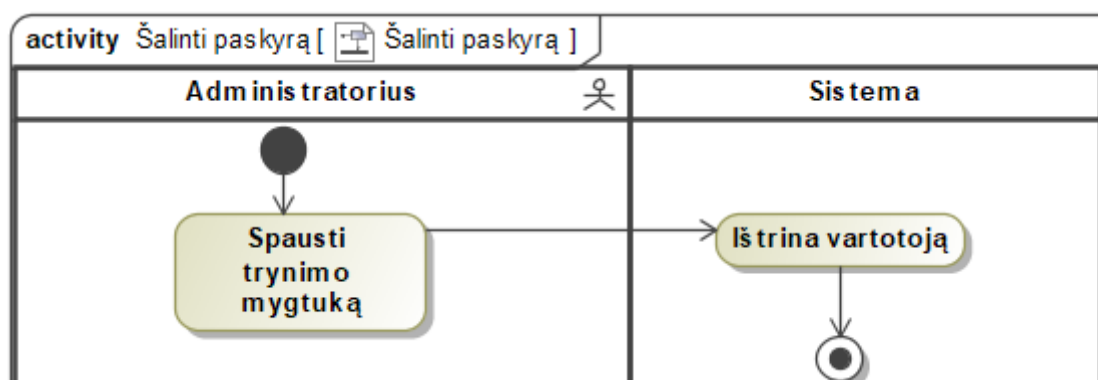
Atsijungimui (4 pav.) tereikia pasirinkti atsijungimo mygtuką esantį lango viršuje sukurtoje navigacijos juostoje. Vartotojas bus atjungtas nuo sistemos ir grąžintas į prisijungimo langą.



5 pav. Veiklos diagrama (Redaguoti paskyrą).

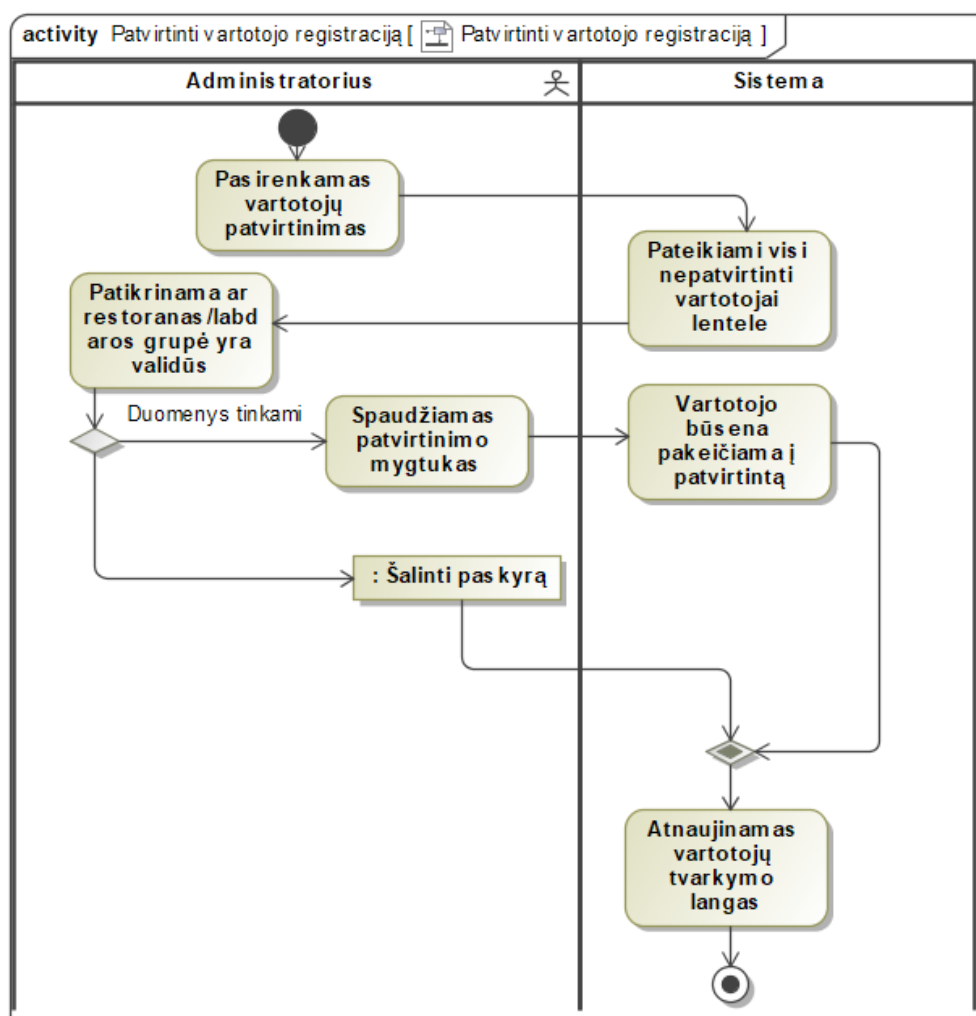
Vartotojui pasirinkus paskyros redagavimą (5 pav.), jam bus pateikiami visi redaguojami laukai, su jo dabartinėmis tų laukų reikšmėmis, kurias pakeitus ir paspaudus mygtuką redaguoti, bus įvykdytas redagavimas. Jeigu laukai bus ištrinti visiškai, arba naujas el. Paštas neatitiks reikalavimų, bus

išvestas klaidos pranešimas. Kitu atveju duomenys bus redaguoti ir vartotojas grąžinamas į skelbimų puslapį.



6 pav. Veiklos diagrama (Šalinti paskyrą).

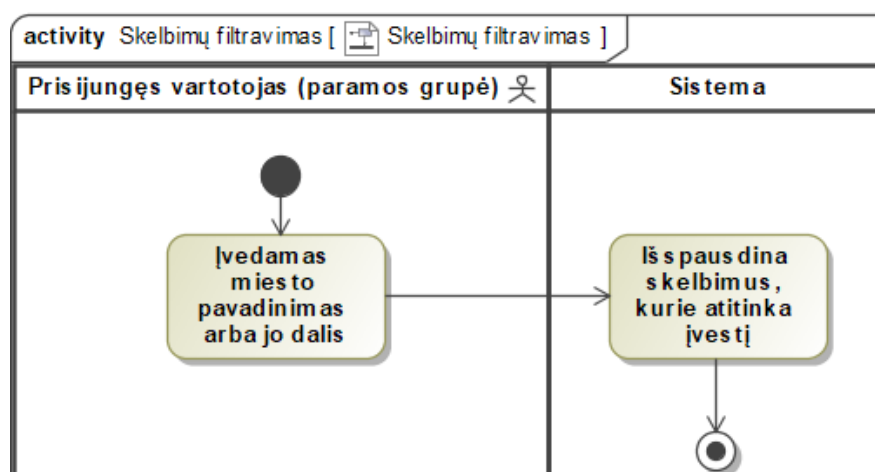
Norint ištrinti neregistruotą vartotoją (6 pav.), reikia vartotojo registracijos puslapyje (7 pav.) paspausti trynimo mygtuką.



7 pav. Veiklos diagrama (Patvirtinti vartotojo registraciją).

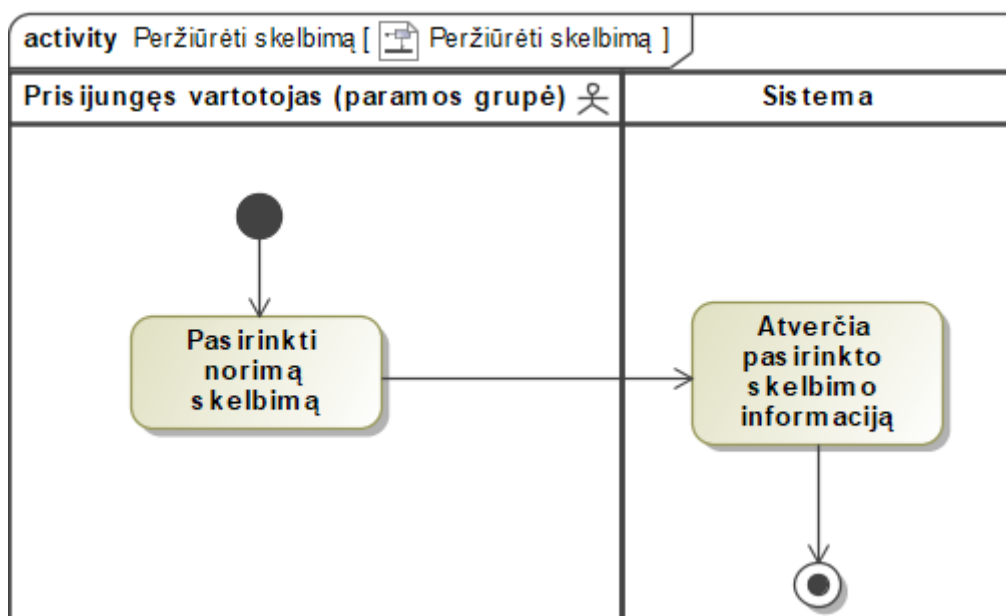
Vartotoją patvirtinti arba atmesti (7 pav.) gali tik administratoriaus teises turintis vartotojas. Norint pasiekti šią funkciją, pirmiausia reikia pasirinkti nepatvirtintų vartotojų mygtuką esantį navigacijos

juostoje. Patekus į vartotojų patvirtinimo/atmetimo puslapį, yra matomi visi užsiregistravę nariai, kurie dar nėra patvirtinti. Administratorius patikrinęs informaciją arba susisiekęs su norinčiu užsiregistruoti restoranu/paramos grupe, patvirtina arba atmeta jų registracijos formą. Kadangi sistema jau turi sukūrusi vartotoją, atmetimo atveju, jis yra ištrinamas, kitu atveju jo patvirtinimo laukas yra pakeičiamas ir nuo tada jis gali prisijungti prie sistemos. Administratoriui, po kiekvieno pridėjimo ar atmetimo, yra perkraunama lentelė su nepatvirtintais vartotojais.



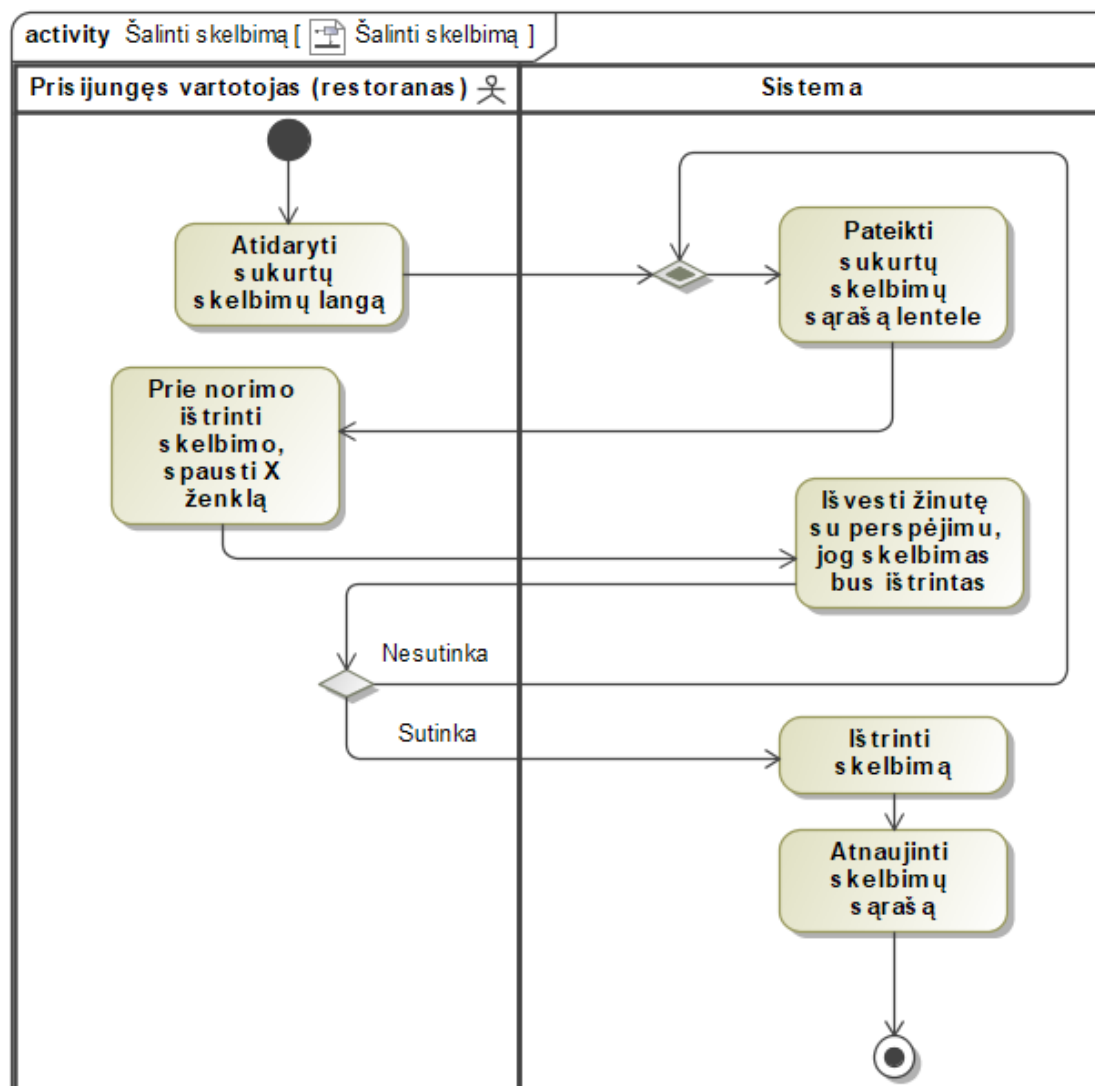
8 pav. Veiklos diagrama (Skelbimų filtravimas)

Skelbimų filtravimas (8 pav.) yra viena iš funkcijų, esančių pagrindiniame skelbimų puslapyje, kuriame tereikia įvesti norimo miesto pavadinimą arba dalį jo, pagal kurią bus atfiltruoti pavadinimo fragmentą atitinkantys skelbimai.



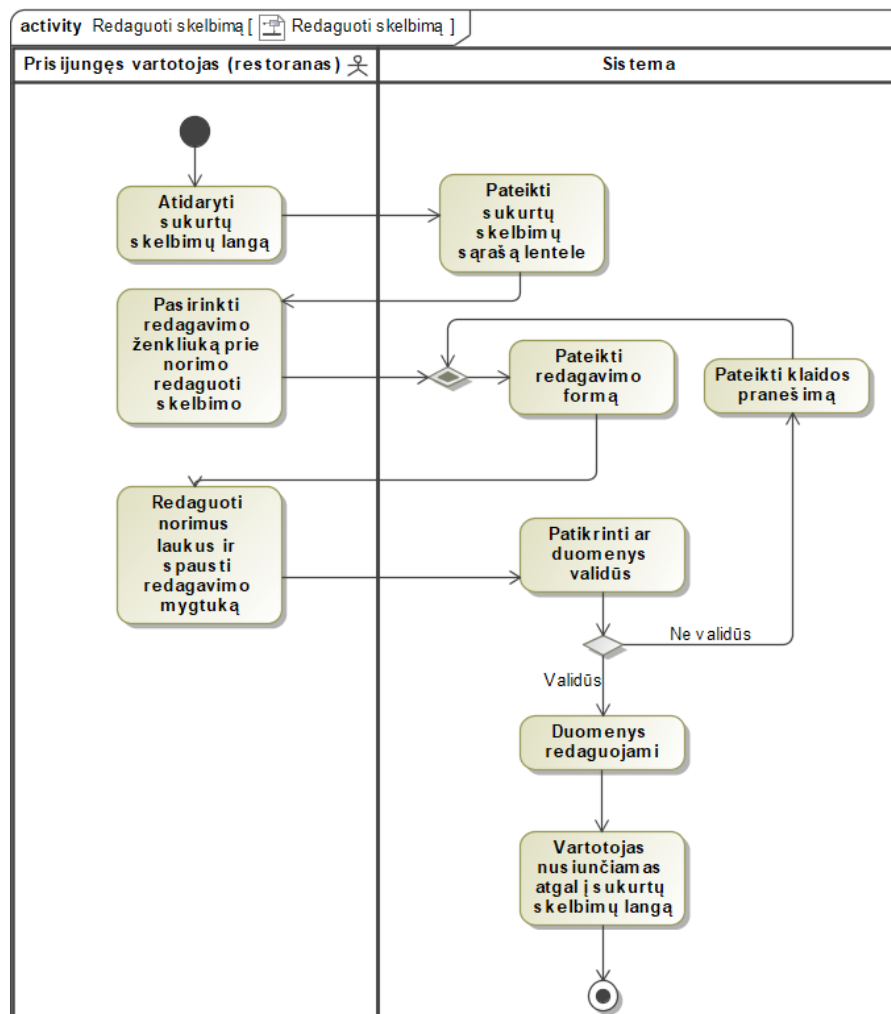
9 pav. Veiklos diagrama (Peržiūrėti skelbimą).

Skelbimo peržiūrėjimas (9 pav.) funkcija, kuri išplečia paspausto skelbimo informaciją kitame lange. Skelbimo informaciją sudaro: restorano kontaktiniai duomenys, skelbimo aprašymas, Google žemėlapis su pažymėtu restorano adresu, bei maisto arba restorano nuotrauka.



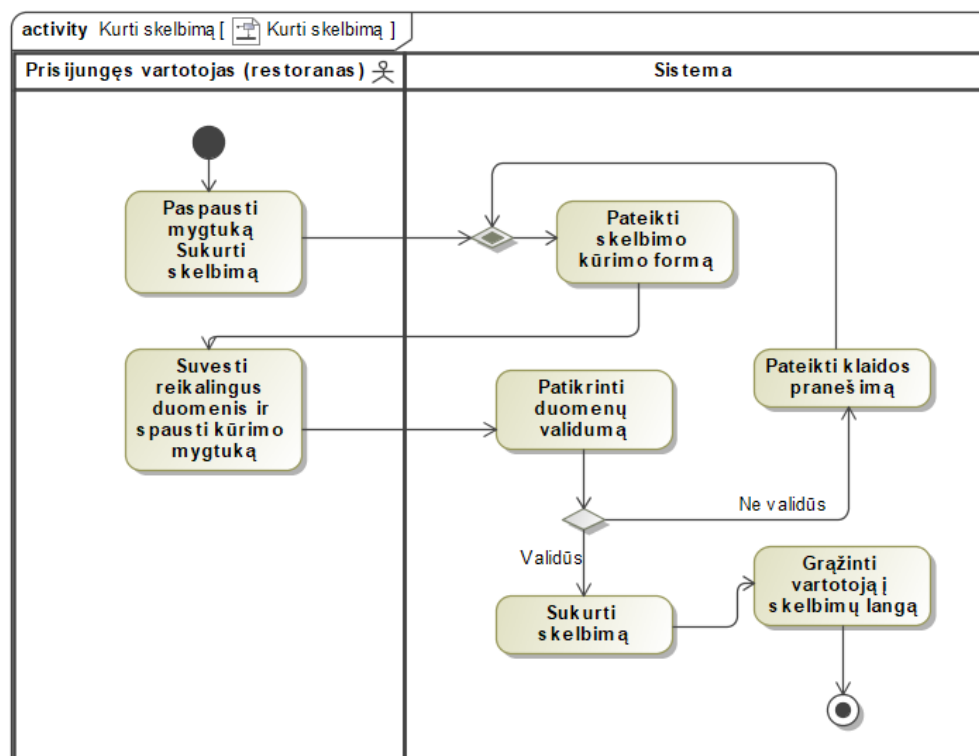
10 pav. Veiklos diagrama (Šalinti skelbimą).

Restoranui, norinčiam ištrinti kažkurį iš savo skelbimų (10 pav.), reikia nueiti į „Mano skelbimai“ puslapį, kuris yra lengvai pasiekiamas navigacijos juostoje. Šiame lange sistema pateikia visus sukurtus skelbimus, kuriuos galima peržiūrėti, trinti arba redaguoti. Vartotojas paspaudęs „X“ ženklą pažymėtą mygtuką, galės ištrinti toje eilutėje esantį skelbimą. Prieš ištrinant, sistema perspės vartotoją, jog sutikus su perspėjimu, skelbimas bus ištrintas. Restoranui sutikus, skelbimas išsitrins ir apie tai bus pranešta apačioje iššokančioje žinutėje. Kitu atveju, vartotojas bus grąžinamas į savo skelbimų langą.



11 pav. Veiklos diagrama (Redaguoti skelbimą).

Skelbimo redagavimui pasiekti (11 pav.), taipogi reikia eiti per „Mano skelbimai“ langą, kuriame šįkart reikia rinktis redagavimo mygtuką, su užrašu „Redaguoti“. Redagavimo forma bus tokia pati, kaip ir skelbimų registracijos, tačiau jau su įvestais dabartiniais duomenimis, kuriuos pakeitus ir paspaudus redaguoti, sistema pradės validavimo operacijas. Jeigu validacija sėkminga – skelbimo duomenys redaguojami ir restoranas nusiunčiamas į savo skelbimų puslapį. Kitu atveju – ekrano apačioje išvedamas raudonas klaidos pranešimas.



12 pav. Veiklos diagrama (Kurti skelbimą).

Restoranui panorėjus sukurti skelbimą (12 pav.), pirmiausia reikia pasirinkti „Sukurti skelbimą“ mygtuką, esantį navigacijos juostoje. Sekančiame lange, sistema pateikia skelbimo kūrimo formą, kurią užpildžius ir paspaudžius kūrimo mygtuką, sistema pradeda duomenų validacijos tikrinimą. Jeigu duomenys validūs – skelbimas yra sukuriamas ir vartotojas yra grąžinamas į skelbimų langą. Kitu atveju yra išspausdinamas raudonas klaidos pranešimas.

2.1.3. Vartotojo sąsajos specifikacija

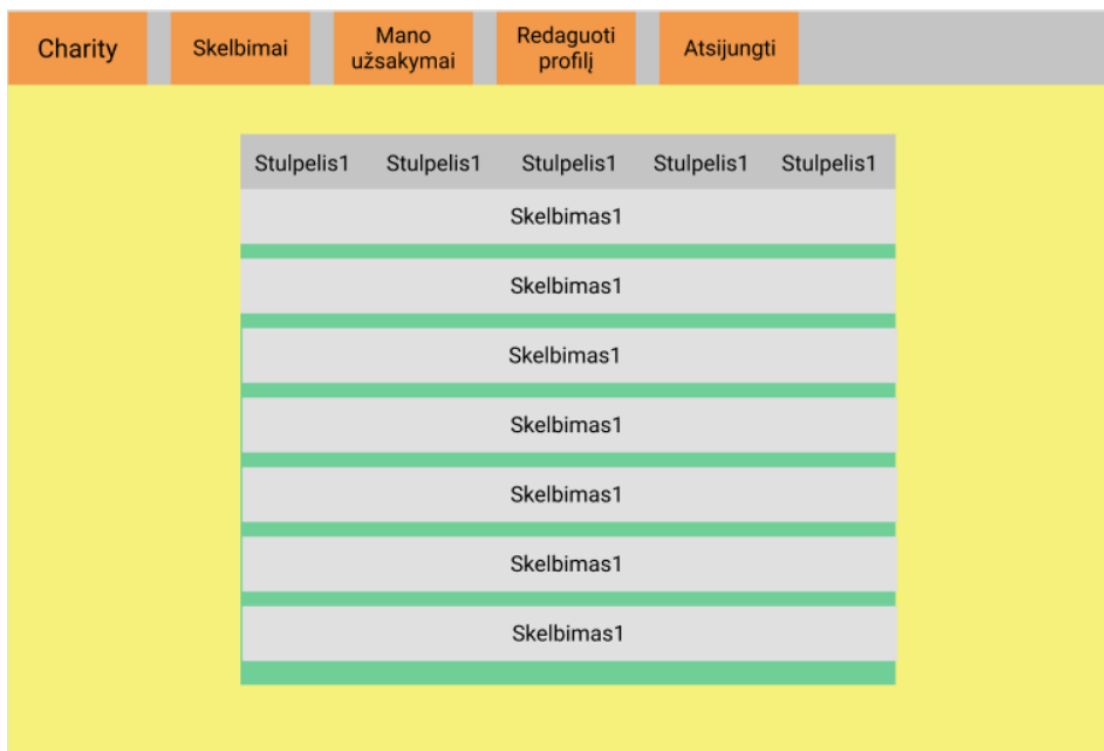
Vartotojo sąsajos specifikavimui buvo pasinaudota *Figma* įrankiu. *Figma* yra galingas komandiniam darbui skirtas grafinio vaizdavimo įrankis (6). Vartotojo sąsajos specifikacijai svarbiausias tikslas yra viską pateikti aiškiai ir lengvai pasiekiamose vietose. Tam tikslui buvo sukurta navigacijos juosta, kuri viso lankymosi svetainėje metu, niekur nedingsta.

13 pav. Prisijungimo lango maketas.

Kaip ir buvo minėta anksčiau, projektui paprastumas yra labia svarbi dalis, dėl to nėra jokių pašalinių langų ar pasirinkimų, puikus pavyzdys yra prisijungimo langas (13 pav.).

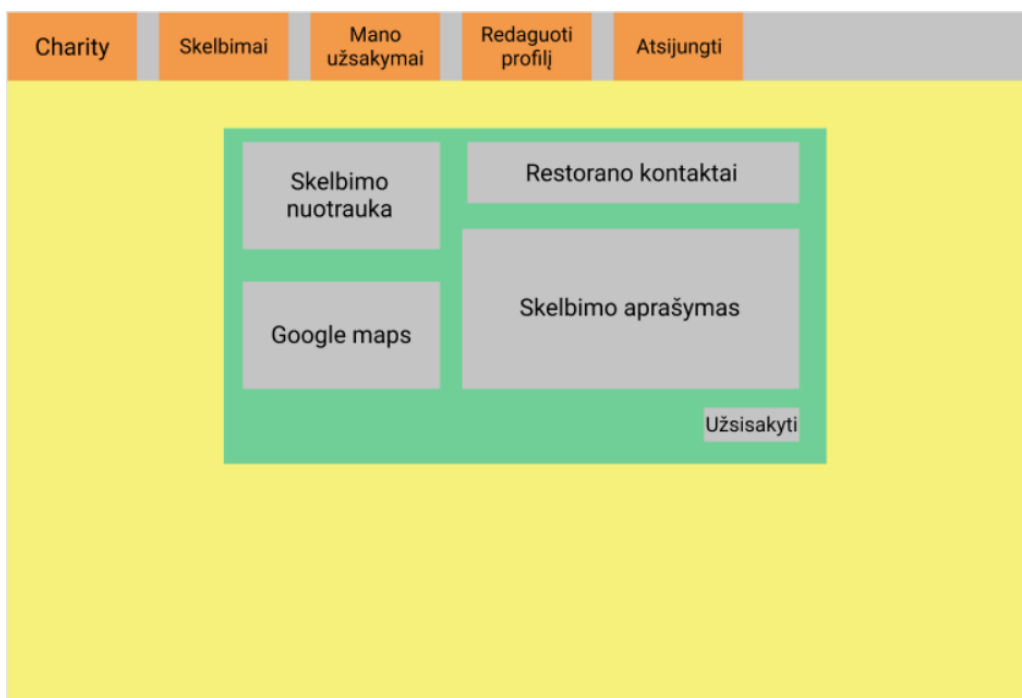
14 pav. Pagrindinio lango maketas.

Pagrindinis projekto puslapis – skelbimų langas. Skelbimų lange kiekvienas skelbimas bus sudėtas į kortelių formos objektus, kuriuos bus galima atfiltruoti pasitelkus pavaizduotą laukelį (14 pav.) miesto vardo arba jo vardo fragmento įrašymui. Valdymo paprastumui, visos galimos funkcijos yra sudėtos į navigacijos juostą, esančią puslapio viršuje.



15 pav. „Mano užsakymai“ lango maketas.

Kiekvieno vartotojo užsirezervuoti skelbimai yra talpinami Mano užsakymai lange, kuriame viskas pateikiama lentelė (15 pav.). Ši lentelės išvaizda truputį keičiasi, priklausomai nuo to, koks vartotojo tipas yra prisijungęs (šiuo atveju nubrėžtas paramos grupių atvejis). Restoranui yra išdėstyti jo sukurti skelbimai ir prie kiekvieno iš jų pasirinkimas jį ištrinti arba redaguoti. Administratoriui tokio tipo lentelė yra matoma pasirinkus vartotojų patvirtinimo mygtuką, kuriame jis gali patvirtinti arba atmesti vartotojo registraciją.



16 pav. Skelbimo lango maketas.

Skelbimo lango išdėstymui (16 pav.) viskas sudėliota paprastai, gerai matomoje vietoje. Google žemėlapiu automatiškai rodytų restorano vietą, bus galima žemėlapią pasididinti. Užsisakyti mygtukas skirtas paramos grupėms, kurios nori užsisakyti rodomą skelbimą.

Charity Skelbimai Mano užsakymai Redaguoti profilį Atsijungti

Profilio redagavimas

Keičiamas atributas Nr. 1

Keičiamas atributas Nr. 2

Keičiamas atributas Nr. 3

Keičiamas atributas Nr. 4

Keičiamas atributas Nr. 5

Redaguoti Atšaukti

17 pav. Paskyros redagavimo lango maketas.

Paskyros redagavimo forma (17 pav.) beveik nesiskiria nuo paskyros kūrimo formos, dėl šios priežasties registracijos forma nebus maketuojama. Skirsis tik atributų laukų kiekis, redaguojant nebus galima keisti savo rolės.

Charity Skelbimai Mano užsakymai Sukurti skelbimą Redaguoti profilį Atsijungti

Profilio redagavimas

Data

Aprašymas

Paveikslėlio Url

Maisto pavadinimas Maisto kiekis X

Pridėti laukų maistui

Redaguoti Atšaukti

18 pav. Skelbimo redagavimo lango maketas.

Skelbimo redagavimo langas (18 pav.), taip pat yra panašus į skelbimo kūrimo langą, vienintelis skirtumas yra mygtuko „Redaguoti“ pavadinimas ir redagavimo atveju, laukai bus užpildyti dabartine reikšme, o ne tušti. Šiame lange taipogi yra svarbūs 2 mygtukai: „X“ ir „Pridėti laukų maistui“. Pirmasis mygtukas ištrina laukų „Maisto Pavadinimas“ ir „Maisto kiekis“ eilutes, o „Pridėti laukų maistui“ mygtukas laukus prideda. Šie mygtukai skirti lengvesniam maisto produktų bei jų kiekiui išvardinimui.

2.1.4. Realizacijai keliami reikalavimai

Sistemai buvo iškelti šie nefunkciniai reikalavimai:

- Sistema turi veikti ant Google Chrome ir Firefox naršyklių.
- Keičiant naršyklės puslapio dydį, sistema turi į tai reaguoti.
- Pagrindiniams veiksmai atlikus, apie sėkmingą darbą pranešti žinutėmis.

2.1.5. Techninė specifikacija

Sistemai realizuoti reikalinga programinė įranga buvo nustatyta lyginant su projekto kūrimo metu naudoto kompiuterio specifikacijomis:

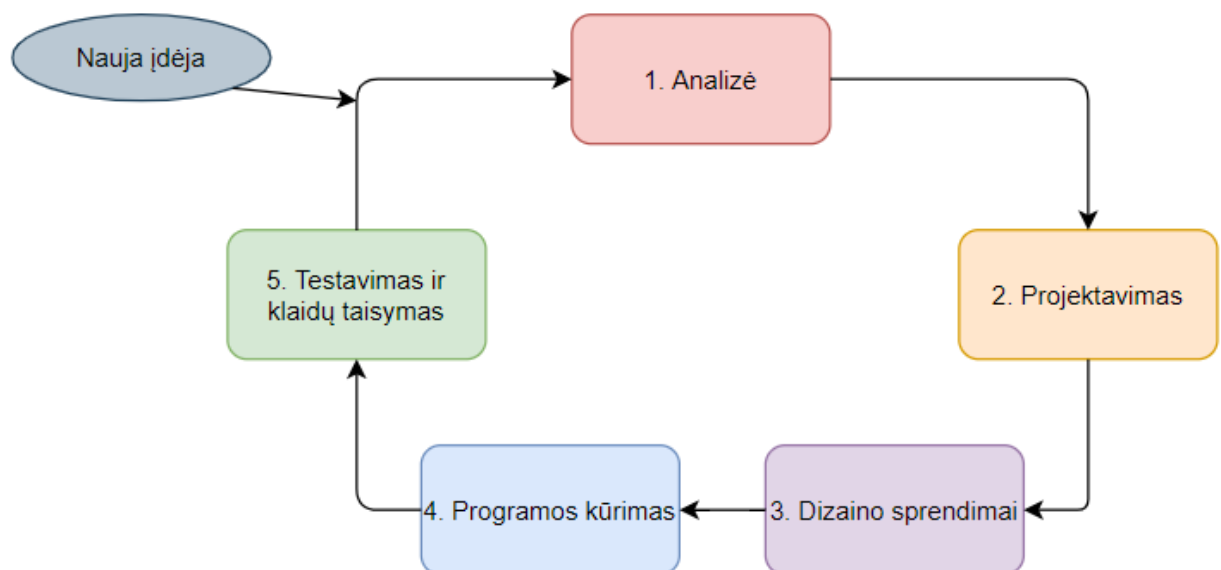
- 16GB operatyvioji atmintis.
- 4 branduolių procesorius.
- 500GB kietasis diskas.

Testavimo bei produkcinei aplinkai, prieš tai paminėtos specifikacijos yra pakankamos. Sistemai panaudoti taipogi reikalingas interneto ryšys, bei naršyklė.

2.2. Projektavimo metodai

2.2.1. Projektavimo valdymas ir eiga

Sistemos kūrimo metu buvo taikomas iteracinis projektavimo modelis (19 pav.).



19 pav. Iteracinis projektavimo modelis.

Naudojantis šio modelio logika, pirmiausia analizuojami norai bei galimybės. Po šio žingsnio sekė surinktos informacijos panaudojimas projektuojant sistemos veikimą. Po to sekė šio veikimo dizaino sprendimai. Sugalvojus kur ir kaip jį implementuoti, prasideda programos kūrimas, po kurio seka testavimas. Nepavykę testo rezultatai parodėdavo, kurios vietos turi būti pataisomos sekančioje iteracijoje. Kaip ir buvo minėta pradžioje, darbas buvo išskirstytas į 7 iteracijas. Žemiau pateiktame sąrašė, jos suspaustos į 5 etapus.

Etapai buvo vykdomi tokiu principu:

- Informacijos rinkimas bei dizaino sprendimai.
- Serverio dalies pagrindinių funkcijų apsirašymas.
- Kliento dalies paprasti sujungimai su minėtu antru etapu.
- Serverio ir kliento kėblesnių funkcijų kūrimas.
- Dizaino pritaikymas.

Taip pat į kiekvieną šių žingsnių yra įtraukiamas testavimas ir klaidų taisymas. Dėmesys pirmiausia buvo skiriamas funkcionalumui, o tik po to dizainui.

2.2.2. Projektavimo technologija

Sistemos projektavimui buvo pasitelkti *UML* grafiniai elementai, tokie kaip: klasių, panaudos atvejų, sekų, bei būsenų diagramos. Diagramoms kurti buvo naudojamas tas pats įrankis, kurį naudojome visą mokymosį laikotarpį – *MagicDraw*. Kad diagramos būtų nesunkiai suprantamos, po kiekvienos iš jų buvo sukurti trumpi aprašymai.

2.2.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinė sistemos

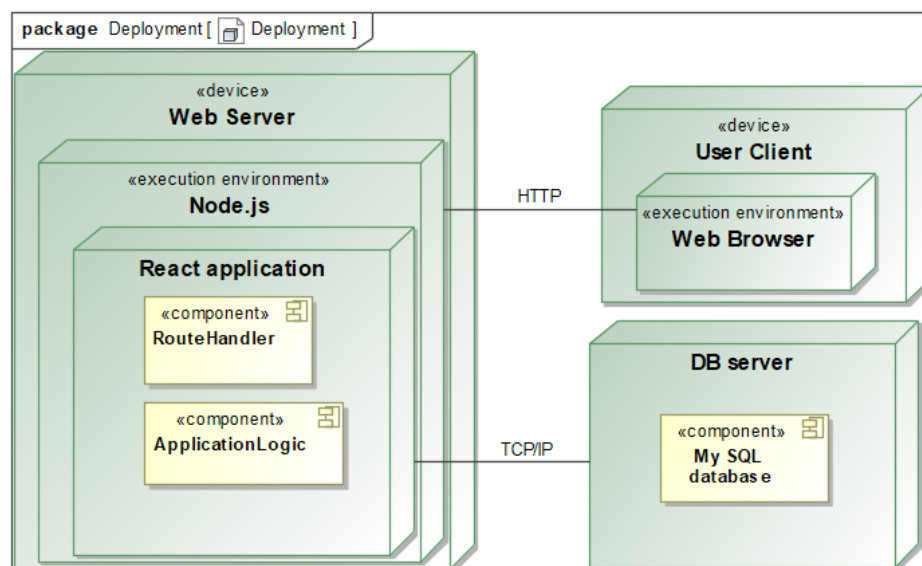
Sistemos serverio dalis buvo sukurta naudojant *C#* programavimo kalbą ir *.NET core* karkasą. Kliento daliai kurti buvo pasitelkta *Javascript* programavimo kalba su *Node.js* vykdymo aplinka ir *React* biblioteka.

Programavimas bei testavimas buvo atliekamas naudojantis *Windows 10* operacine sistema, *Visual Studio 2019* bei *Visual studio code* programavimo aplinkomis. Duomenų bazės valdymui ir derinimui buvo naudojama *SSMS 18*.

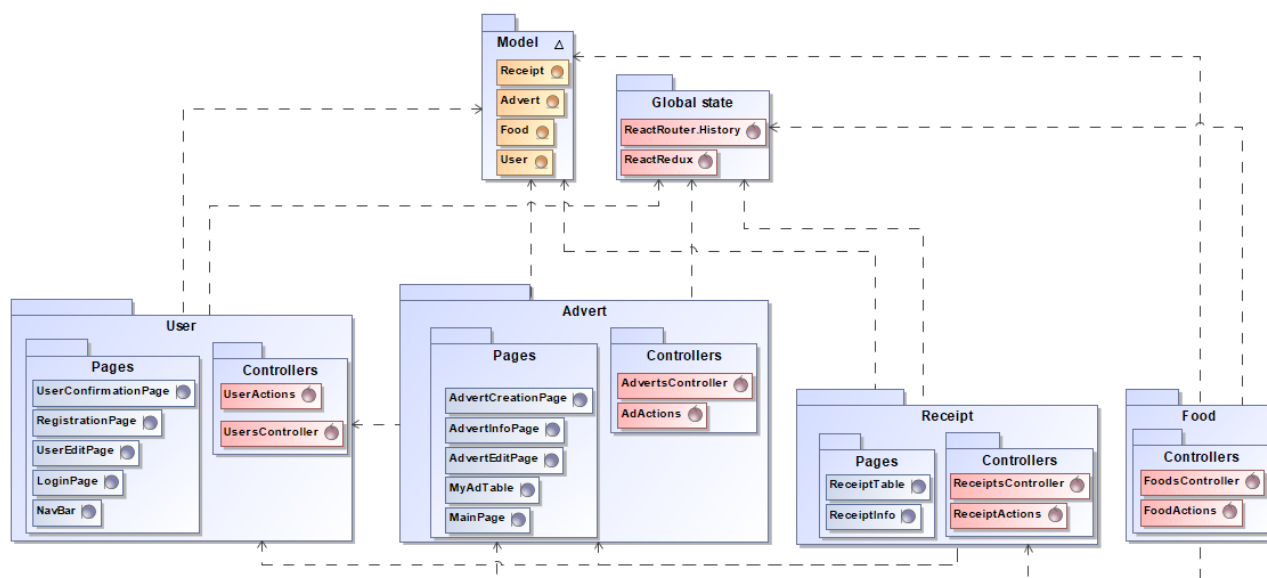
2.3. Sistemos projektas

2.3.1. Statinis sistemos vaizdas

Sistemos statiniam vaizdai paaiškinti buvo pasirinktos paketų, diegimo ir klasių diagramos.

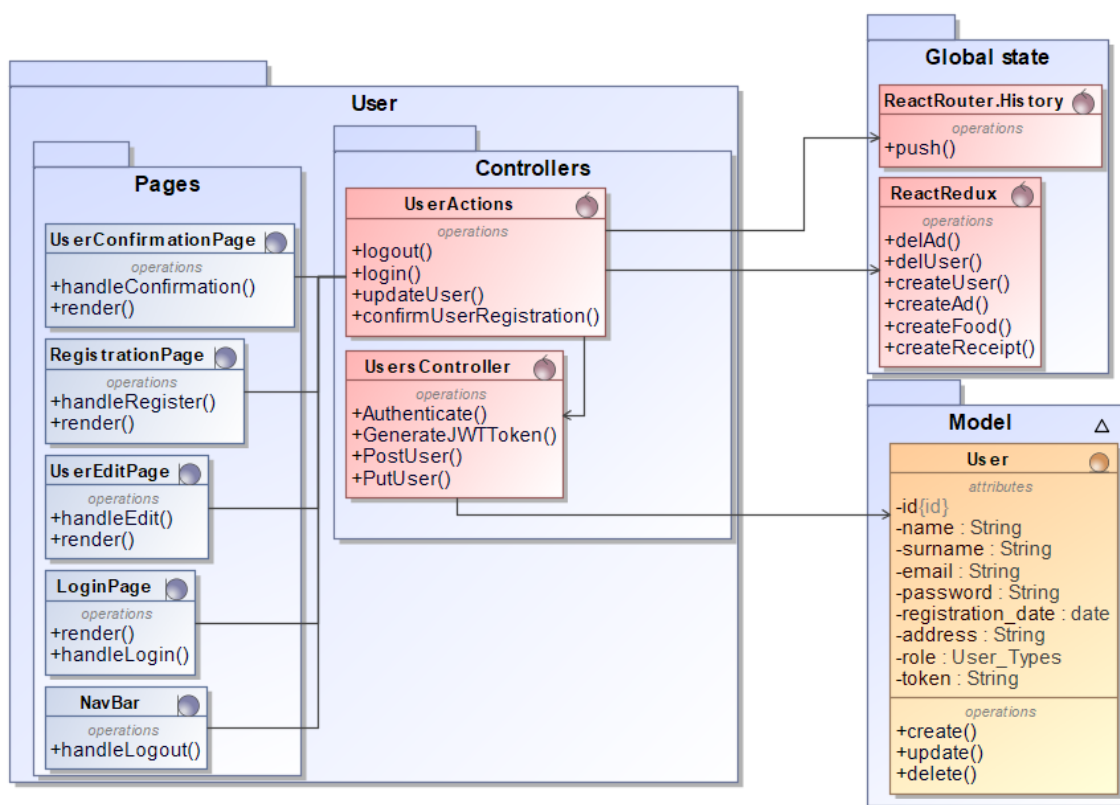


20 pav. Diegimo diagrama.



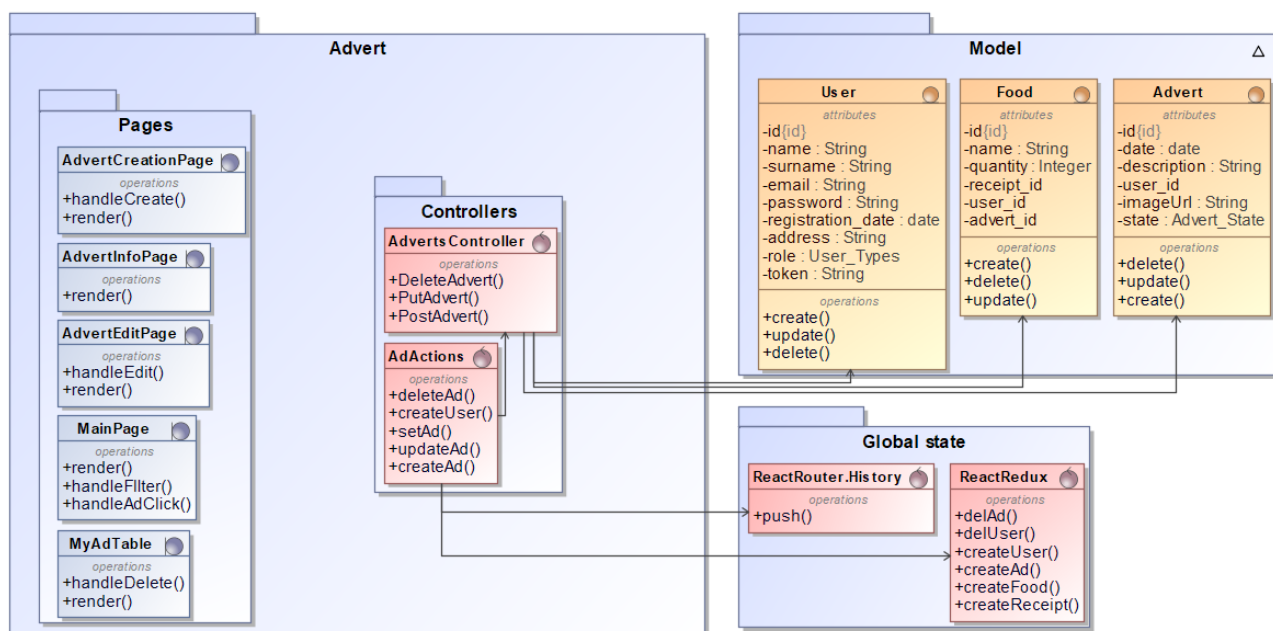
21 pav. Paketų diagrama.

Sistemos sąryšius, bei naudojamus komponentus, puikiai atvaizduoja paketų diagrama (21 pav.).

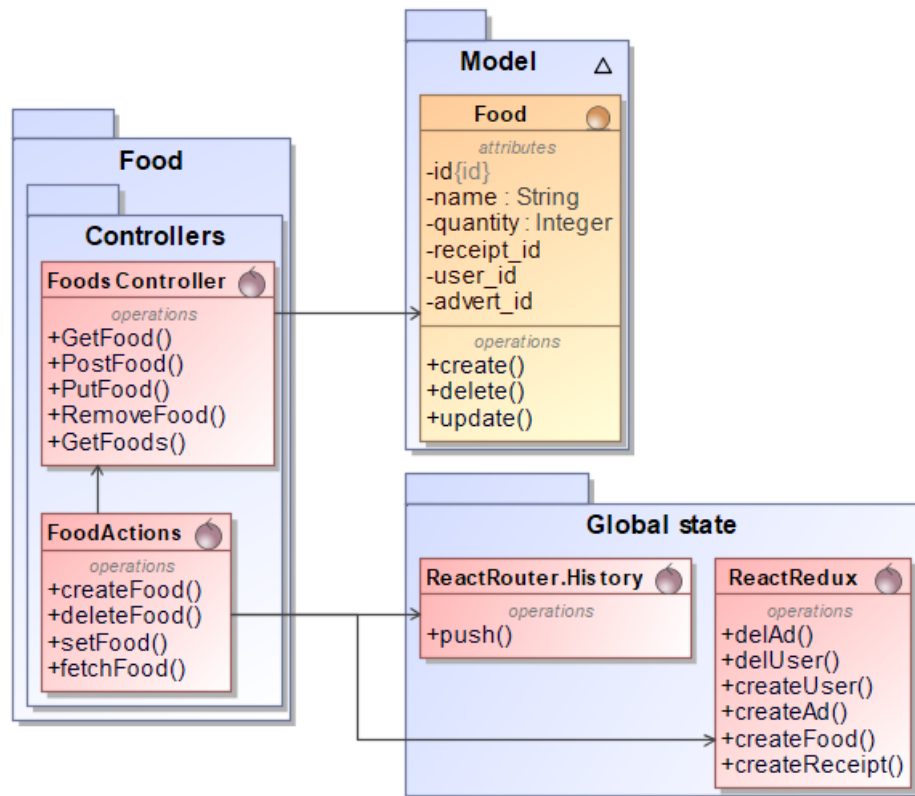


22 pav. Vartotojo klasės diagrama.

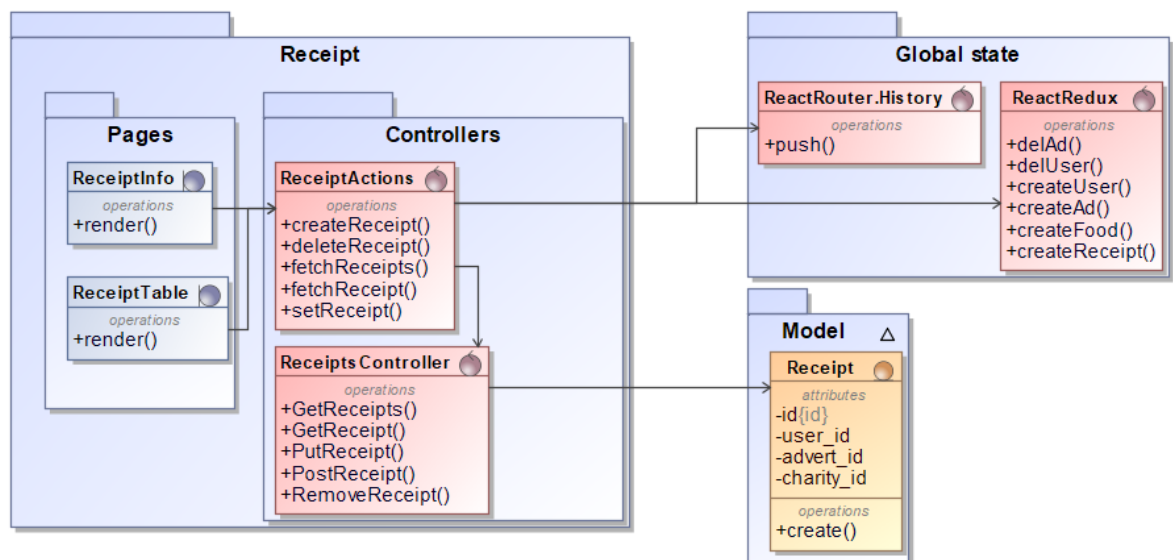
Visi kliento pusėje esantys veiksmai, pirmiausia pasiekia vartotojo valdiklį, kuris po to, priklausomai nuo scenarijaus, „bendruoja“ su tam skirtais komponentais.



23 pav. Skelbimo klasės diagrama.



24 pav. Maisto klasės diagrama.



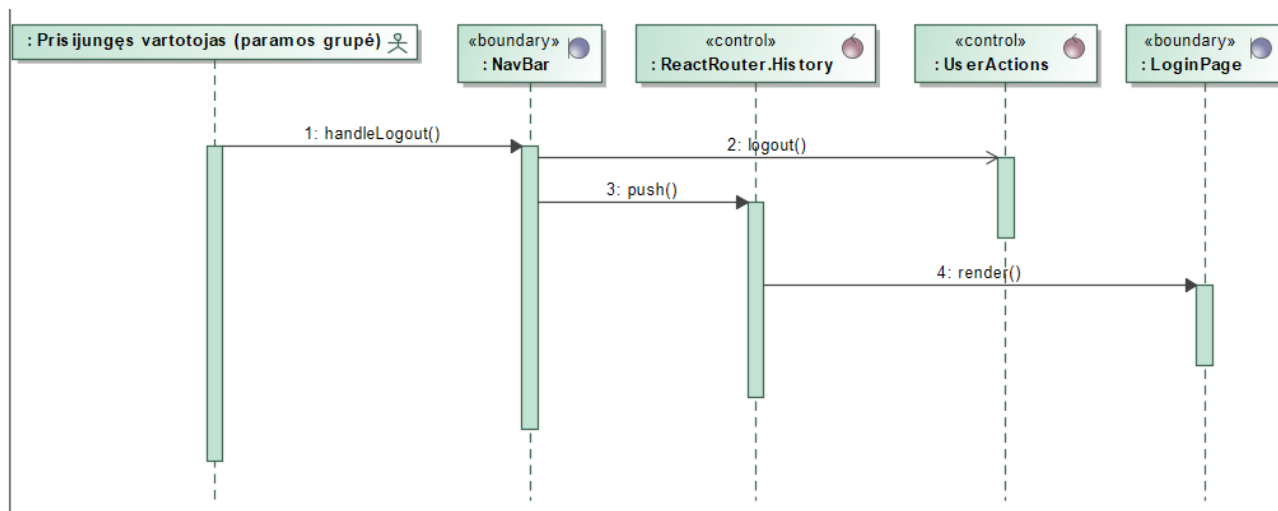
25 pav. Išrašo klasės diagrama.

Iš viršuje pateiktų diagramų (22 pav. - 25 pav.), aiškiai matyti, kokie komponentai bendrauja su kokiais valdikliais, bei kokius metodus turi savo viduje.

2.3.2. Dinaminis sistemos vaizdas

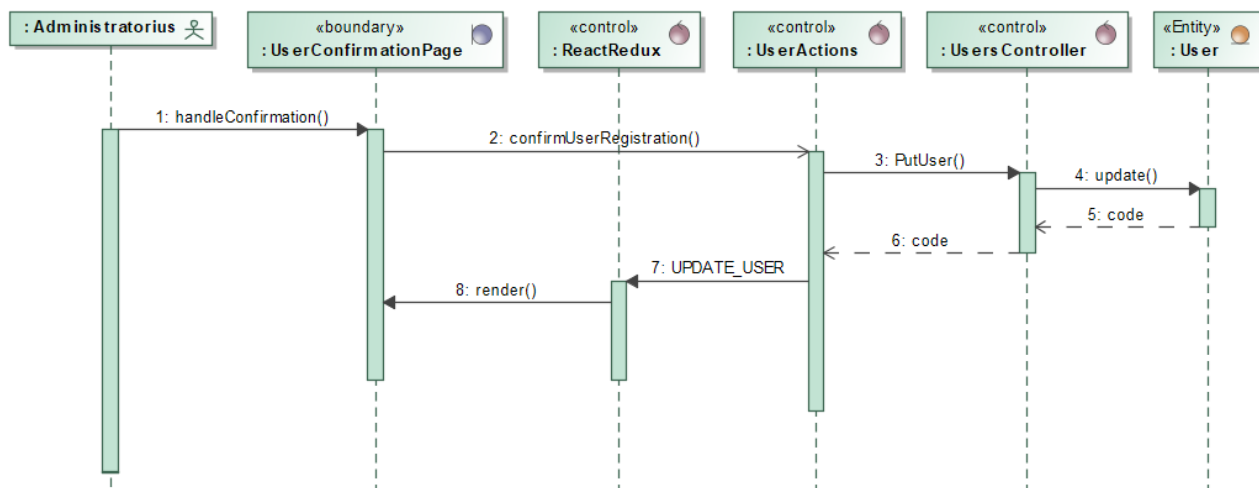
Dinaminiam sistemos vaizdo paaiškinimui, buvo pasirinktos sekų diagramos. Šiuo būdu siekiama kuo aiškiau parodyti naudojamus metodus bei klasių tarpusavio ryšius. Kadangi funkcijų veikimai

buvo pavaizduoti ir aprašyti veiklos diagramomis ankstesniame skyriuje, sekų diagramų abstrakcijos lygis bus gana aukštas – bus rodomi pagrindiniai metodai ir sąryšiai.



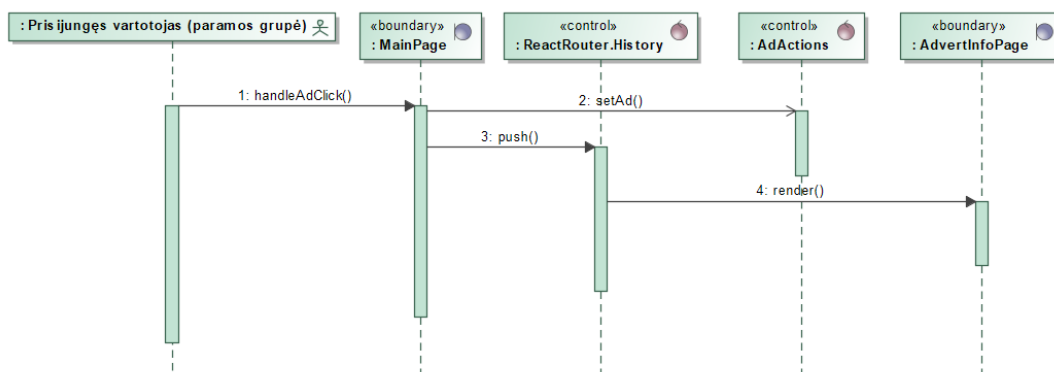
26 pav. Sekų diagrama (Atsijungti).

Prisijungęs vartotojas, baigęs veiklą sistemoje, gali atsijungti (26 pav.), o tai padaryti galima paspaudus atsijungimo mygtuką navigacijos juostoje. Vartotojas pirmiausia siunčia metodą navigacijos juostai, kuri iššaukia atsijungimo funkciją iš vartotojų veiksmų valdiklio, kadangi veiksmas yra asinchroninis, kitas valdiklis, kuris yra atsakingas už vaikščiojimą tarp puslapių, panaudoja *push()* metodą vartotojui perkelti į prisijungimo langą.



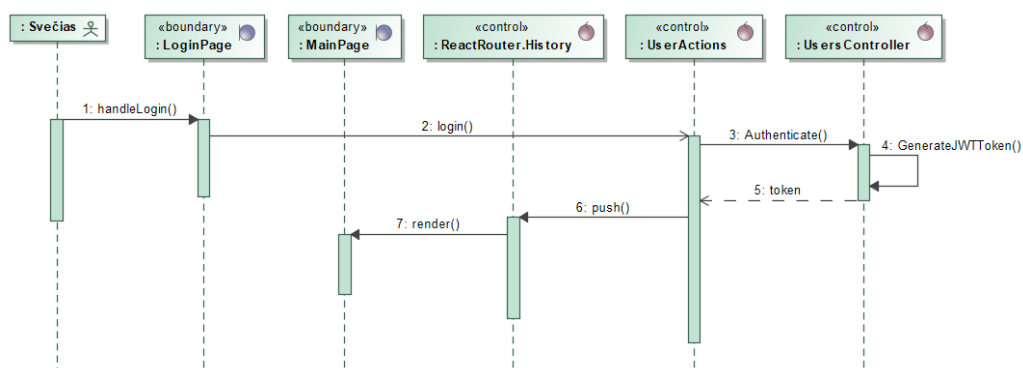
27 pav. Sekų diagrama (patvirtinti vartotojo registraciją).

Vartotojo registracijai patvirtinti (27 pav.) yra naudojami 3 valdikliai – Redux, kuris yra skirtas globaliai projekto informacijai laikyti, vartotojų veiksmų valdiklis, bei vartotojų valdiklis, esantis serveryje. Vartotojui paspaudus patvirtinimo mygtuką, suveikia funkcija, kuri duotam langui įsako naudoti metodą iš vartotojų veiksmų valdiklio. Šis valdiklis kreipiasi į serverį, kuriame pakeičia pasirinkto vartotojo *Confirmed* lauką į 1. Tai reiškia, kad vartotojas jau gali prisijungti prie sistemos. Atnaujinus šią informaciją, grąžinamas kodas, pagal kurį kliento dalyje pakeičiamas patvirtintas vartotojas. Veiksmo pabaigoje, perkraunamas vartotojų registracijos patvirtinimo langas.



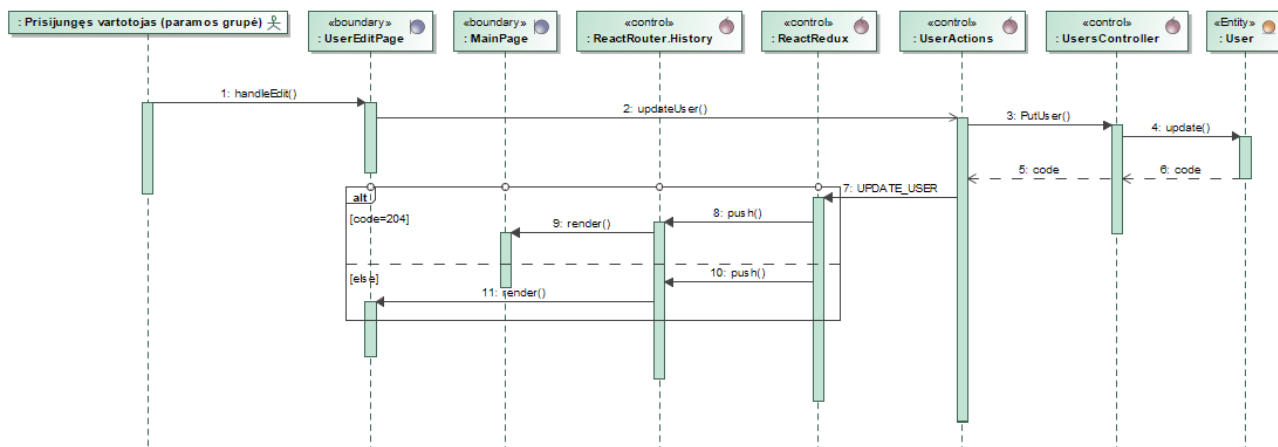
28 pav. Sekų diagrama (Peržiūrėti skelbimą).

Skelbimo peržiūrėjimo atvejis (28 pav.) nelabai skiriasi nuo atsijungimo atvejo, kadangi veikimo principas tas pats, tiesiog skiriasi metodai, bei valdikliai, kadangi yra dirbama nebe su vartotojo veiksmiais, o su skelbimų.



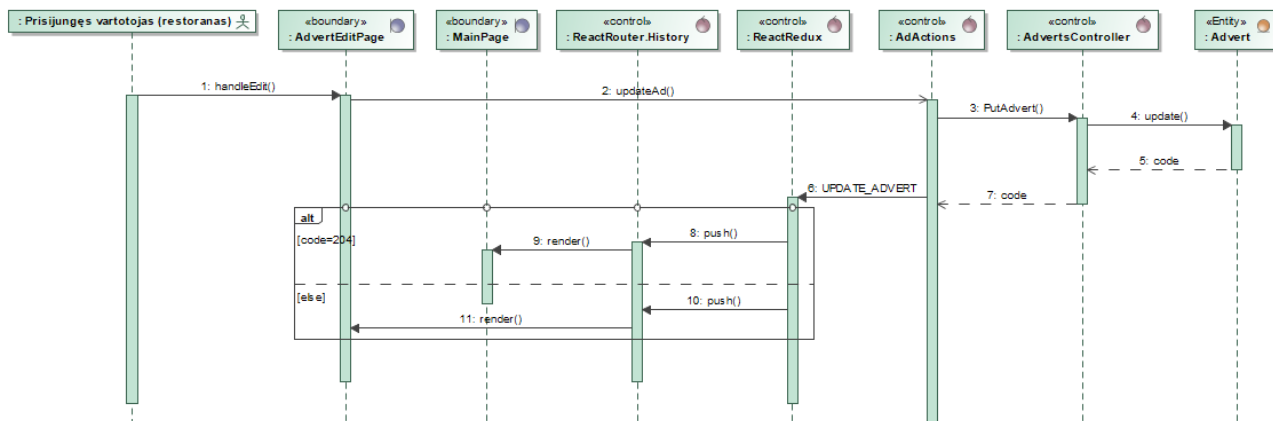
29 pav. Sekų diagrama (Prisijungimas).

Prisijungimui (29 pav.) svečias suvedęs teisingus duomenis ir paspaudęs prisijungti mygtuką, inicijuoja funkcijų seką, į kurią įeina vartotojo veiksmų valdiklis, serverio pusės vartotojų valdiklis, kuris sugeneruoja JWT tokeną ir jį grąžina, kad prisijungęs vartotojas būtų autentifikuotas visą jo sesijos laiką. JWT tokeno struktūrą sudaro: antraštė, krovinytis ir parašas (7). Po prisijungimo vartotojas nusiunčiamas į skelbimų puslapį.



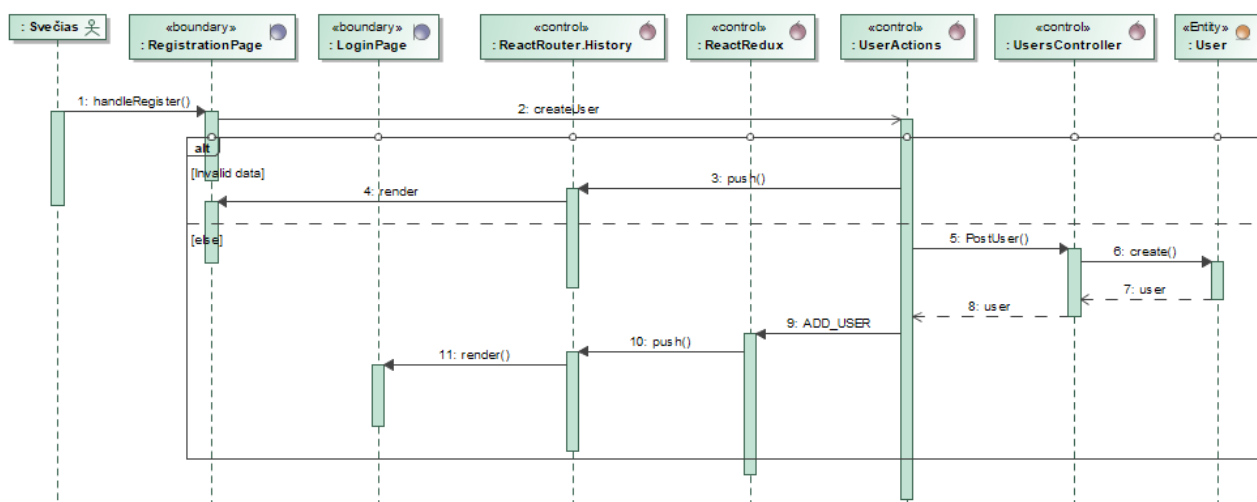
30 pav. Sekų diagrama (Paskyros redagavimas).

Paskyros redagavimas (30 pav.) veikia panašiu principu, kaip ir vartotojo patvirtinimas, tačiau šįkart iš serverio grąžinamas HTTP kodas 204 yra svarbus, kadangi pagal jį bus sprendžiama ar atnaujinti vartotojo informaciją globaliame valdiklyje ar ne. Jeigu kodas yra tinkamas, vartotojo paskyra yra redaguojama ir vartotojas grąžinamas į pagrindinį puslapį, jeigu kodas kitoks – išvedamas klaidos pranešimas ir vartotojas paliekamas redagavimo puslapyje.



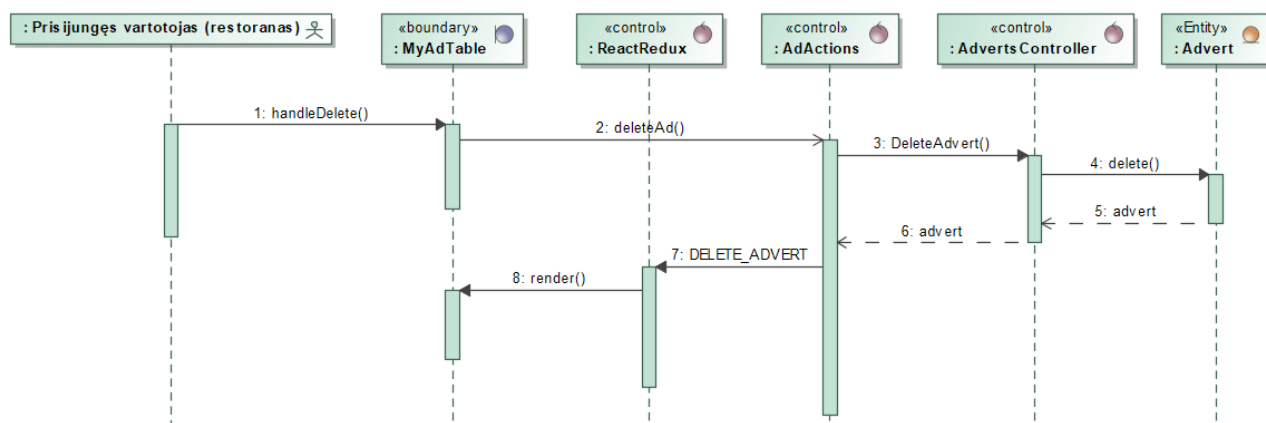
31 pav. Sekų diagrama (Skelbimo redagavimas).

Skelbimo redagavimas (31 pav.) savo veikimo principu nesiskiria nuo prieš tai minėto paskyros redagavimo (30 pav.). Skiriasi tik naudojami langai bei atitinkami valdikliai.



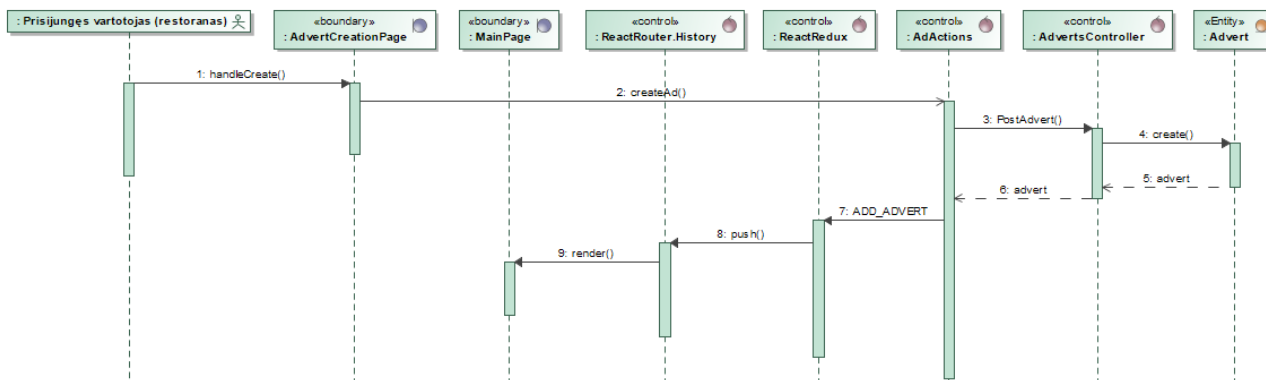
32 pav. Sekų diagrama (Registracija).

Registracijai (32 pav.) visa logika ir metodika panaši į prieš tai minėtus atvejus, tik šįkart pademonstruojamas netinkamų duomenų įvedimas, kuris neleidžia vartotojui susikurti paskyros, kol šie duomenys neištaisomi.



33 pav. Sekų diagrama (Skelbimo šalinimas).

Skelbimo šalinimui, restorano rolę turintis vartotojas turi nueiti į savo skelbimų langą, kuriame yra trynimo mygtukas. Jį paspaudus, inicijuojama visa tvarka, pavaizduota diagramoje (33 pav.). Ištrinus skelbimą serverio pusėje, grąžinamas ištrintas skelbimas, kuris tuomet pašalinamas ir kliento pusėje esančiame globalių kintamųjų valdiklyje *Redux*.



34 pav. Sekų diagrama (Skelbimo kūrimas).

Skelbimo kūrimui (34 pav.) inicijuoti, privaloma būti skelbimo kūrimo puslapyje ir turėti restorano rolę. Visa ideologija tokia pati, kaip registruojant vartotoją.

3. Testavimas

3.1. Testavimo planas

Sistamai testuoti buvo pasitelkti trijų tipų testavimai:

1. Kodo peržiūra
2. Integraciniai testai
3. Rankiniai testai

Pagrindinis analizės tikslas – ištaisyti kuo daugiau klaidų, surasti nenaudojamus kodo fragmentus bei paspartinti projekto veikimą.

3.2. Testavimo kriterijai

Kodo peržiūrai buvo taikyti šie kriterijai:

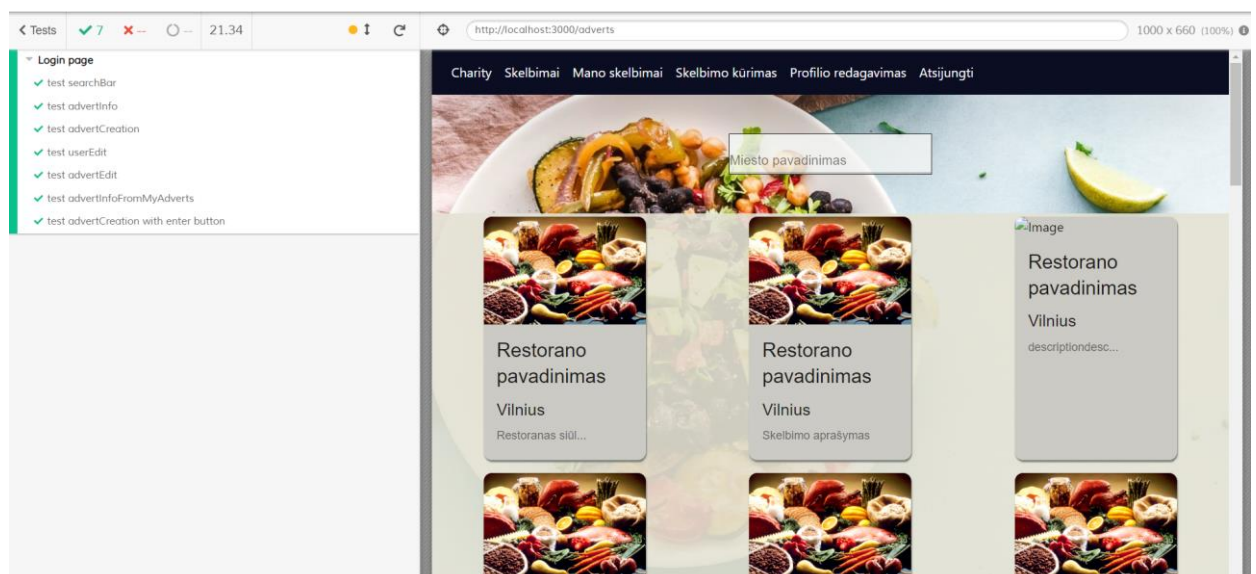
1. Programoje turi likti kuo mažiau nenaudojamų importuotų atributų.
2. Turi nelikti nenaudojamų metodų.
3. Neturi būti užkomentuoto programinio kodo.

Testuojant sistemą integraciniais bei rankiniais testais, buvo taikyti šie kriterijai:

1. Funkcijos turi veikti taip, kaip aprašyta specifikacijoje.
2. Metodai, kurie reikalauja prisijungimo, turi būti nepasiekiami neautentifikuotiems vartotojams.
3. Metodai, kurie reikalauja specifinės rolės, turi būti nepasiekiami kitokias roles turintiems vartotojams.
4. Suvedus klaidingus duomenis, turi būti grąžinamas klaidos kodas.

3.3. Integracinis testavimas

Integraciniams testams buvo pasitelktas *Cypress* – kliento pusėje testuoti skirtas įrankis (35 pav.). Juo naudojantis buvo atlikti tam tikri scenarijai, tokie kaip prisijungimas, skelbimų redagavimas, kūrimas.

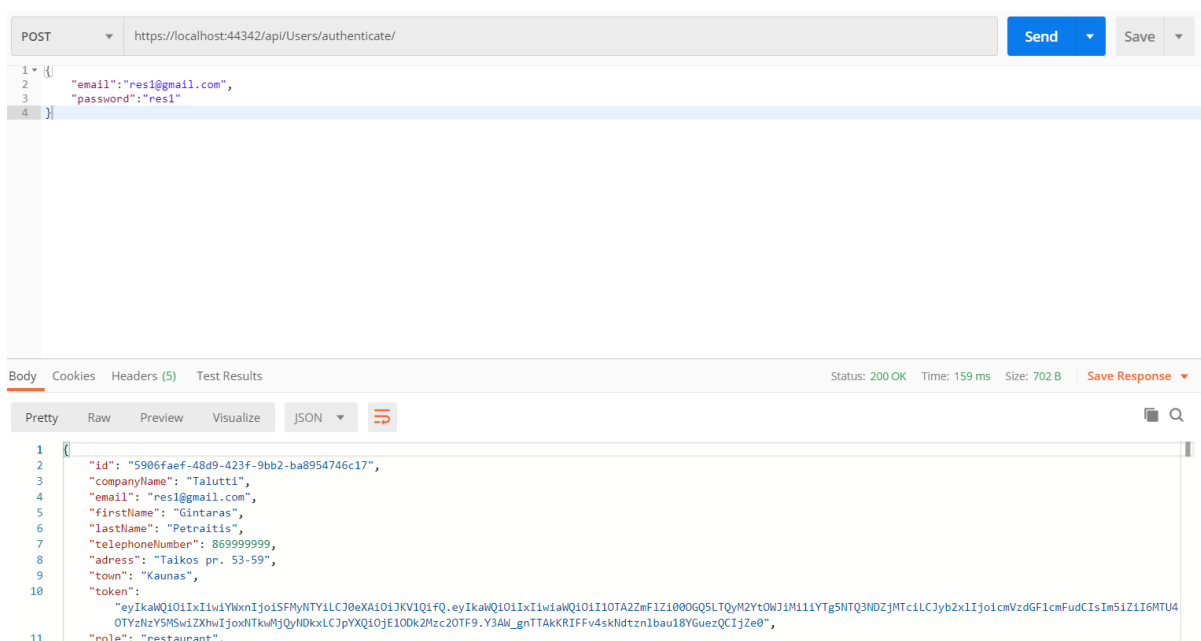


35 pav. Testavimo aplinka.

Testavimo metu išbandyti visi sistemoje esantys kliento-serverio veiksmai. Testavimo scenarijams atlikti buvo susiprojektuojama vartotojo veiksmų seka, kuri po to buvo perverčiama veiksmams, kuriuos naudoja *Cypress*.

3.4. Vartotojo sąsajos testavimas

Atliekant rankinį testavimą, buvo sukurti tam tikri scenarijai, kuriuose paduodami užklauskos duomenys ir laukiamas rezultatas. Kiekvienas scenarijus turi atskirą nesėkmingą scenarijų, kuris taip pat bus testuojamas. Testavimui atlikti bus naudojamas *Postman* įrankis (36 pav.).



36 pav. Postman autentifikacijos scenarijus.

Scenarijų reikšmės ir laukiami rezultatai pavaizduoti lentelėje. Kiekvieno veiksmo metu, turi būti grąžinamas HTTP kodas, kai kuriais variantais sistema grąžina ir objektą. Tokie atvejai atsitinka vykdant trynimo veiksmus.

2 lentelė Autentifikacijos testavimo scenarijus.

Lauko pavadinimas	Reikšmė
Email	<u>res1@gmail.com</u>
Password	res1
Rezultatai	Grąžinamas HTTP 200 kodas reiškia, jog funkcija įvykdyta sėkmingai. Metodas taip pat grąžina vartotojo informaciją, kaip parodyta paveikslėlyje (36 pav.). Taip pat yra sukuriamas JWT žetonas.

3 lentelė GetUser testavimo scenarijus.

Lauko pavadinimas	Reikšmė
Id	5906faef-48d9-423f-9bb2-ba8954746c17
Rezultatai	Grąžintas vartotojo objektas ir HTTP 200 kodas.

4 lentelė PutUser testavimo scenarijus.

Lauko pavadinimas	Reikšmė
Id	d3b0bb56-4b67-416a-9527-768ea78ad6e0
Visi vartotojo laukai	Visa vartotojo informacija
Rezultatai	Atnaujinta vartotojo informacija ir grąžintas HTTP kodas 204.

5 lentelė ConfirmUser testavimo scenarijus.

Lauko pavadinimas	Reikšmė
Id	d3b0bb56-4b67-416a-9527-768ea78ad6e0
Rezultatai	Atnaujinta vartotojo Confirmed vertė ir grąžintas HTTP kodas 204.

6 lentelė DeleteUser testavimo scenarijus

Lauko pavadinimas	Reikšmė
Id	d3b0bb56-4b67-416a-9527-768ea78ad6e0
Rezultatai	Grąžinamas HTTP 200 kodas ir ištrinto vartotojo informacija.

7 lentelė ConfirmOrder testavimo scenarijus.

Lauko pavadinimas	Reikšmė
Id	d3b0bb56-4b67-416a-9527-768ea78ad6e0
Rezultatai	Grąžinamas HTTP 200 kodas ir ištrinto vartotojo informacija.

Taip buvo atliekami visi su užklausomis susiję metodai (iš viso jų yra 23). Visuose testavimuose rezultatai gauti tokie, kokių tikėtasi, kadangi testavimai nuolatos vykdavo programos kūrimo metu. Testavimo metu buvo bandyta daryti veiksmus neprisijungus, tokiais atvejais būdavo gaunamas HTTP 401 kodas, kuris reiškia, kad vartotojas nėra autentifikuotas. Taip pat buvo bandyta labdaros teises turinčiam vartotojui kurti skelbimą, ko pasekoje buvo gaunamas HTTP 403 kodas, kuris reiškia draudimą.

4. Dokumentacija naudotojui

4.1. Apibendrintas sistemos galimybių aprašymas

Sukurta sistema suteikia galimybę restoranams bei labdaros grupėms registruotis, prisijungti, gauti JWT žetoną, su kuriuo vartotojas neatjungiamas iki sesijos pabaigos. Labdaros grupės rolę turintys vartotojai gali peržiūrėti, užsisakyti skelbimus, taip pat patogiai pasirinkti skelbimus pagal miestą, įvedus į paieškos laukelį norimą miesto pavadinimo fragmentą. Jie taip pat gali redaguoti savo paskyros duomenis bei turi lentelę su visų savo užsakymų išrašais. Restoranas gali daryti viską, ką daro labdaros grupė ir dar kurti, trinti bei redaguoti skelbimus.

4.2. Vartotojo vadovas

Dokumentacijoje pateikiami pagrindiniai sistemos naudojimo žingsniai, kuriuos išaiškinus, vartotojai turėtų sklandžiai naudotis svetainės teikiamomis funkcijomis. Pirmiausia vartotojui reikia užsiregistruoti, ten jis suveda prie kiekvieno lauko aprašytas reikšmes. Baigęs duomenų vedimą, vartotoja spaudžia registracijos mygtuką ir laukia, kol administratorius patvirtins arba atmes jo registraciją, galimas telefono skambutis registracijos informacijai patikslinti. Po patvirtinimo, vartotojas turi prisijungti, kad galėtų patekti į pagrindinį sistemos puslapį. Prisijungimui reikalingi registracijoje įvesti el. Pašto pavadinimas ir slaptažodis. Teisingai suvedus duomenis, vartotojas nukeliamas į pagrindinį puslapį. Šiame puslapyje yra langelis skelbimų filtravimui pagal miesto pavadinimą, bei patys skelbimai, kurie yra kortelių formos. Paspaudus ant kortelės yra atverčiama detalesnė skelbimo informacija, jo įdėjusio restorano kontaktiniai duomenys, įdėta maisto arba restorano nuotrauka, bei Google žemėlapis su pažymėta restorano vieta. Restorano teises turintys vartotojai negali užsisakyti kitų skelbimų, tik kurti savo. Navigacijos juostoje, esančioje ekrano viršuje yra siūlomi kiti veiksmai susiję su skelbimais bei vartotojo paskyra.

Charity Prisijungimas Registracija

Registracija

Jmonės pavadinimas

Vadovo vardas

Vadovo pavardė

Telefono numeris

EL paštas

Miestas

Adresas

Slaptažodis

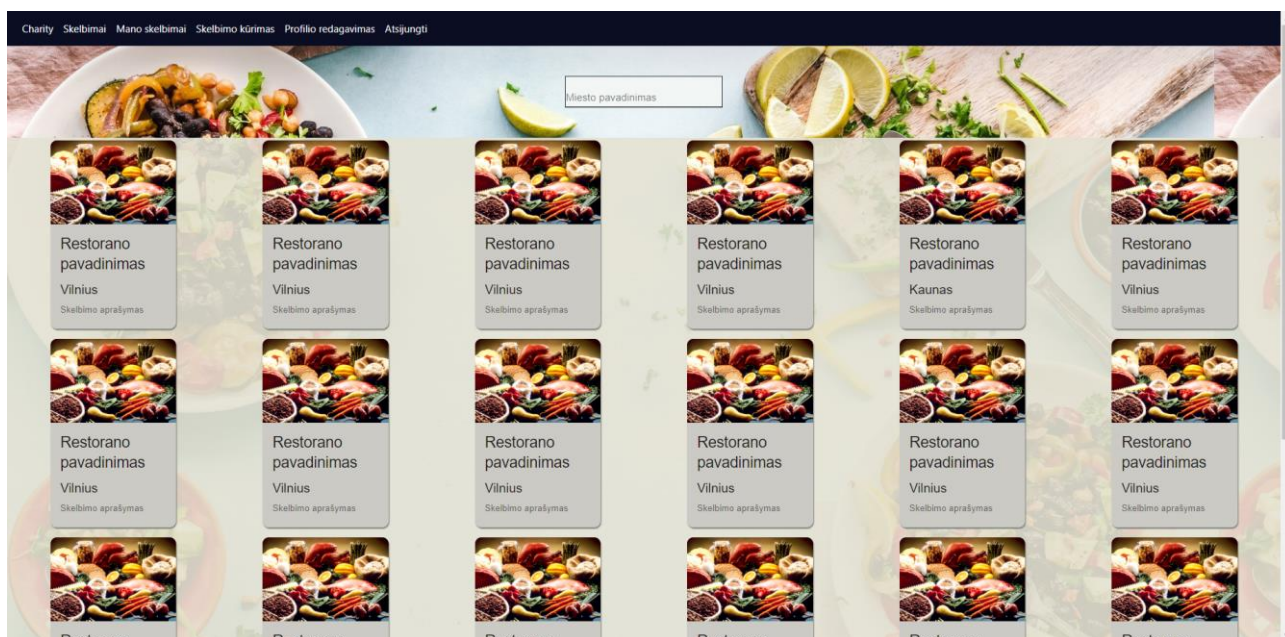
Rolė

37 pav. Registracijos puslapis.

Vartotojo registracijai (37 pav.) reikalinga įvesti kuo galima daugiau laukų, kad administratoriui būtų lengviau įvertinti restorano ar labdaros grupės autentiškumą. Rolės skiltyje reikia įvesti arba restaurant, jeigu vartotojas yra restoranas arba charity jeigu vartotojas yra labdaros grupė.

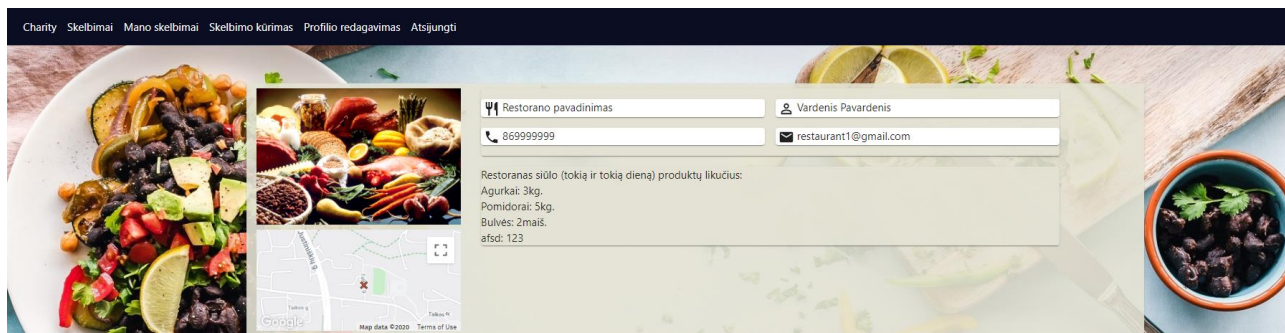
38 pav. Prisijungimo puslapis.

Kaip ir buvo minėta anksčiau, prisijungimui (38 pav.) reikalingas vartotojo el. Paštas bei slaptažodis. Neįvedus bent vieno iš šių laukų, prisijungimo mygtukas bus užblokuotas.



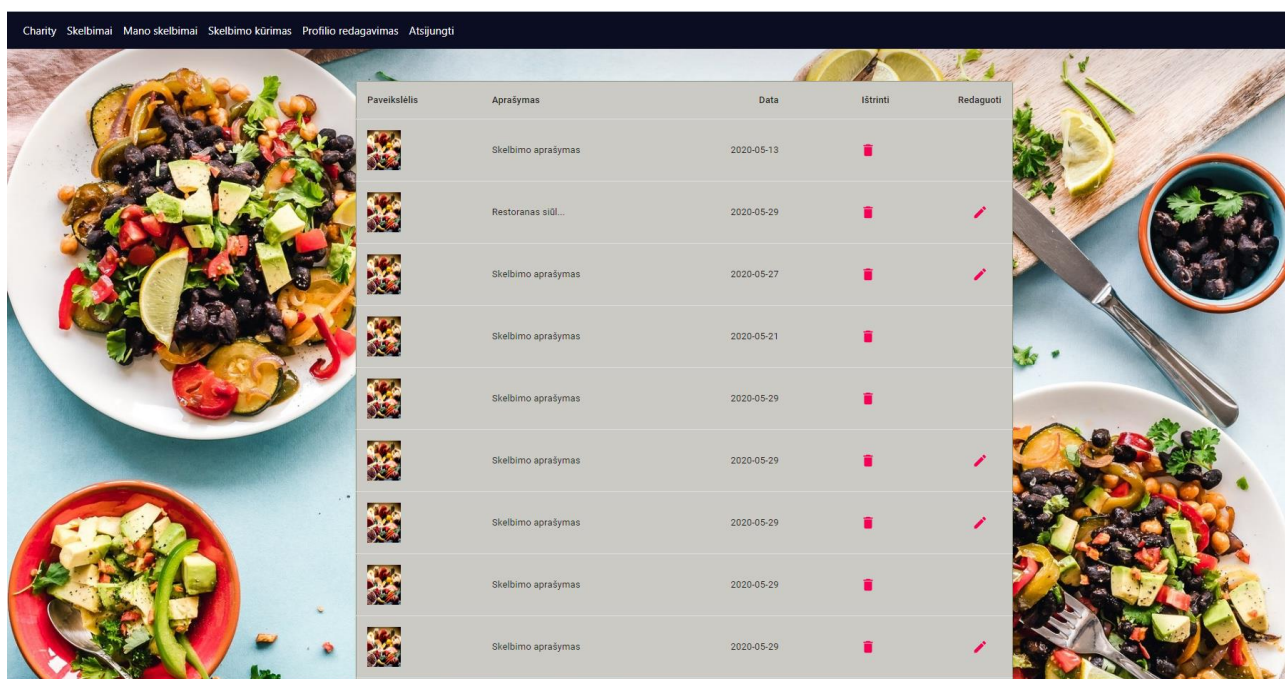
39 pav. Pagrindinis puslapis.

Kaip matoma iš paveikslėlio (39 pav.), pagrindiniame puslapyje esantis prisijungęs restorano rolę turintis vartotojas, mato visus įdėtus skelbimus, filtravimo lauką, kuriame prašoma įvesti miesto pavadinimo, bei navigacijos juostą, kurioje yra jo kitos galimos funkcijos.



40 pav. Skelbimo puslapis.

Skelbimo detalesnė informacija (40 pav.) išdėstyta tokiu principu: skelbimo kūrėjo kontaktai, aprašymas, maisto produktai. Jungiantis su labdaros grupės paskyra, vartotojas turėtų pasirinkimą užsisakyti šį skelbimą. Vartotojui panorėjus, jis gali pasididinti Google žemėlapyje esantį kairiame apatiniame formos kampe.



41 pav. Vartotojo sukurtų skelbimų puslapis.

Restoranui, norinčiam peržiūrėti, redaguoti ar ištrinti savo sukurtus skelbimus reikia pasirinkti navigacijos juostoje esantį Mano Skelbimai mygtuką, kuris nusiunčia vartotoją į puslapį su visais jo skelbimais, sudėtais į lentelę (41 pav.). Vartotojas, norintis peržiūrėti savo skelbimo detalesnę informaciją, turi spausti ant bet kurios jį dominančio skelbimo eilutės vietos. Skelbimo trynimasis ir regavimas veikia taip pat, tik paspausti reikia ant ikonų, šiuokšliadėžė – trynimui, pieštukas – redagavimui. Kai kurie skelbimai nebeturi pieštuko ikonos nes jie jau yra užsakyti.

Charity Skelbimai Mano skelbimai Skelbimo kūrimas Profilio redagavimas Atsijungti

Skelbimo kūrimas

Data
05/15/2020

Aprašymas
Skelbimo aprašymas

Paveikslėlio nuoroda
https://upload.wikimedia.org/wikipedia/commons/6/6d/Good_Food_Display_-_NCL_Visuals_Online.jpg

Agurkai 2kg. PAŠALINTI

Pomidorai 3kg. PAŠALINTI

PRIDĖTI

SUKURTI ATŠAUKTI

42 pav. Skelbimo kūrimo puslapis.

Skelbimo kūrimui (42 pav.) reikia įvesti dabartinės dienos datą, skelbimo aprašymą, paveikslėlio nuorodą. Norint pridėti daugiau teksto laukų maisto produktams bei jų kiekiams, spaudžiamas pridėti mygtukas. Sukūrus per daug laukų, juos galima pašalinti pasirinkus tam paskirtą mygtuką esantį kiekvienos maisto eilutės gale. Suvedus visus norimus duomenis, vartotojas paspaudžia mygtuką Sukurti. Po sukūrimo vartotojas grąžinamas į pagrindinį puslapį.

Charity Skelbimai Mano skelbimai Skelbimo kūrimas Profilio redagavimas Atsijungti

Profilio redagavimas

Įmonės pavadinimas
Restorano pavadinimas

Vadovo vardas
Vardenis

Vadovo pavardė
Pavardenis

Telefono numeris
869999999

El. pašto adresas
restaurant1@gmail.com

Miestas
Vilnius

Adresas
Taikos pr. 53-59

REDAGUOTI ATŠAUKTI

43 pav. Vartotojo paskyros redagavimo puslapis.

Vartotojo profilio redagavimas (43 pav.) savo forma niekuo nesiskiria nuo registracijos formos, tik nuimami slaptažodžio, bei rolės teksto laukai.

4.3. Diegimo vadovas

Kadangi sistema kol kas padaryta lokaliajoje aplinkoje, viskas ko reikia vartotojui tai aplinkų programos kodui kompiliuoti. Projekto serverinei daliai kurti buvo naudojama *Visual Studio 2019* aplinka, su *Microsoft.EntityFramework Core 3.1.4* versijos karkasu. Duomenų bazei valdyti siūloma naudoti prieš tai minėtą SSMS 18 įrankį. Kliento daliai buvo naudojama *Visual Studio Code* aplinka su npm. Pirmą kartą atsidarius kliento dalį, terminale reikia suvesti *npm install*, kad būtų įdiegti visi reikalingi paketai, naudotis šiuo projektu. Atlikus šiuos žingsnius, lieka paskutinis etapas – paleisti serverio pusę ir kliento pusėje, terminale įrašyti *npm start*. Įvykdžius šiuos žingsnius, atsidaro naršyklės langas su veikiančiu projektu.

Rezultatai ir išvados

Atlikus darbą, buvo prieita prie šių išvadų:

1. Programos kūrimo metu buvo sėkmingai panaudotas vienas svarbiausių *React* bibliotekos siūlomų komponentų - *Redux*, kuris leido visame projekte pasiekti kitus, su tam tikra klase nesusijusius atributus ar metodus. Taip buvo sumažintas pasikartojančio kodo panaudojimas ir išlaikyta taisyklinga programavimo tvarka.
2. Projektavimo metu buvo įsigilinta į *Google* siūlomas *Google Maps* ir *Geocoding API*, kurie yra plačiai naudojami dėl savo paprasto panaudojimo bei daug laiko sutaupančio funkcionalumo.
3. Testavimo planavimo bei jo vykdymo metu, pasinaudojus integracinius, rankinius, bei vienetų testus, nustatyta daug sutrikimų, kurie jų dėka buvo pašalinti.
4. Ateityje numatoma tobulinti sistemos funkcionalumą, pridėdant bendravimo funkciją, restoranų reitingavimo sistemą, atsiliepimų puslapį, įdedant statistikas, kurios demonstruotų kiek ir kokiuose miestuose apytiksliai atiduodama maisto.

Literatūros sąrašas

1. **Google.** Google Maps Documentation. <https://developers.google.com/maps/documentation>. [Tinkle] Google. [Kreiptasi: gegužės 6, 2020.] <https://developers.google.com/maps/gmp-get-started>.
2. **Google.** Google Geocoding. <https://developers.google.com/maps/documentation>. [Tinkle] Google. [Kreiptasi: gegužės 6, 2020.] <https://developers.google.com/maps/documentation/javascript/geocoding>.
3. **Ghanayem, Mark.** Microsoft. <https://docs.microsoft.com/>. [Tinkle] Microsoft, rugsėjo 11, 2019. [Kreiptasi: kovo 23, 2020.] <https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver15>.
4. **Keli, autoriai.** nodejs. nodejs.org. [Tinkle] Kelios kompanijos. [Kreiptasi: kovo 19, 2020.] https://nodejs.org/api/documentation.html#documentation_about_this_documentation.
5. **Anderson, Rick.** Microsoft. <https://docs.microsoft.com/>. [Tinkle] Microsoft, balandžio 17, 2020. [Kreiptasi: balandžio 26, 2020.] <https://docs.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-3.1>.
6. **Banks, Joey.** Figma. [figma.com](https://www.figma.com). [Tinkle] Figma. [Kreiptasi: balandžio 7, 2020.] <https://www.figma.com/best-practices/guide-to-developer-handoff/components-styles-and-documentation/>.
7. **Auth0.** jwt.io. JWT. [Tinkle] Auth0. [Kreiptasi: balandžio 14, 2020.] <https://jwt.io/introduction/>.