



VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
KOMPIUTERIJOS KATEDRA

Baigiamasis bakalauro darbas

Maitinimo įstaigų valdymo sistema

Atliko:

Konstantinas Arefjevas

parašas

Vadovas:

dr. Severinas Zubė

Vilnius
2018

Turinys

Sutartinis terminų žodynas	3
Santrauka	4
Summary	5
Ivydas	6
1. Panašių sistemų analizė	7
1.1. Meniu	7
1.2. Restoranai	7
1.3. Kur pavalgyti	7
1.4. Open table	8
1.5. Book a table	8
1.6. Apibendrinimas	8
1.7. Funkciniai sistemos reikalavimai	9
2. Projektavimas	10
2.1. Techniniai sprendimai	10
2.2. Sistemos veikimo architektūra	10
2.3. Sistemos aktoriai	11
2.4. Panaudos atvejų diagrama	12
2.5. Duomenų bazės architektūra	13
3. Realizavimas	19
3.1. Valdikliai	19
3.2. Registracija	21
3.3. Maisto ir gėrimų valgiaraštis	21
3.4. Staliukų išsidėstymo schemas	23
3.5. Staliukų rezervavimas	26
3.6. Maitinimo įstaigų valdymas	28
Išvados ir rekomendacijos	29

Sutartinis terminų žodynas

Prezentacinė dalis - galutiniam vartotojui matoma aplikacijos dalis (angl. front-end)

Loginė dalis - galutiniam vartotojui nematoma aplikacijos dalis (angl. back-end). Joje vyksta pagrindiniai aplikacijos procesai, vykdomi atitinkami algoritmai rezultatams gauti. Gauti rezultatai atvaizduojami prezentacinėje dalyje.

MVC - Modelio, valdiklio ir rodinio technologija naudojama programavimo karkasuose (angl. Model - View - Controller).

Modelis - atsakingas už sistemos sąveiką su duomenų baze.

Valdiklis - atlieka esminius loginius veiksmus ir grąžina duomenis į rodinį.

Rodinys - atsakingas už duomenų atvaizdavimą.

Santrauka

Šio darbo tikslas - sukurti maitinimo įstaigų valdymo sistemą. Šiam tikslui įgyvendinti buvo atlikta panašių egzistuojančių sistemų analizė, kuri padėjo suformuluoti pagrindinius sistemos reikalavimus ir uždavinius.

Maitinimo įstaigų valdymo sistema skirta dviejų tipų naudotojams: maitinimo paslaugų teikėjams ir jų klientams. Maitinimo įstaigų vadovai arba jų atstovai gali užsiregistruoti savo įmone šioje sistemoje - pateikti informaciją apie maitinimo įstaigą, savo siūlomo maisto valgiaraštį, nubrėžti staliukų išsidėstymo schemą. Antras vartotojų tipas - klientai, gali matyti užsiregistravusias maitinimo įstaigas, jas filtruoti. Pasirinkus norimą įstaigą galima pamatyti jos meniu ir atlikti užsakymą, rezervuoti staliuką.

Summary

Catering Management System

The main goals of this bachelor work are to create catering management system, perform similar systems analysis and apply strongest properties in current system.

Catering management system aimed for two main user types: catering services providers and their customers. Catering providers could register their company in current system thus provide necessary information, create food menu and tables position schema. Second user type – customers, could navigate through registered catering systems, filter them and find suitable one. By choosing the desired company, users can see food menu and create orders from it and book a table relying on tables schema.

Ivadas

Internetinės sistemos plačiai naudojamos įvairios paskirties duomenų saugojimui ir apdorojimui bei naudotojų grupių priėjimui prie tų duomenų organizuoti. Dažniausiai tokios sistemos yra prieinamos ne vieno, bet didelio kiekio naudotojų. Naudotojai dažniausiai būna kelių lygių, nuo lygių priklauso naudotojų funkcijos, bei galimybės sistemoje. Tokio tipo sistemose visada būna aukščiausio lygio naudotojai, kurie turi visas galimas funkcijas ir gali pilnai kontroliuoti ir administruoti sistemą ir jos naudotojus. Kiti naudotojų lygiai yra įgyvendinami atsižvelgus į sistemos paskirtį ir poreikius.

Šio darbo tikslas - sukurti laisvai prieinamą maitinimo įstaigų valdymo sistemą, kurios pagalba galima greitai ir patogiai rasti išsamią informaciją apie maitinimo įstaigas, peržiūrėti įstaigų siūlomą maisto meniu ir staliukų išsidėstymo schemą, daryti užsakymus, bei rezervacijas. Sistemos paskirtis - padėti maitinimo įstaigoms ir jų klientams greičiau surasti vienas kitą.

Siekiant išsiaiškinti tokios sistemos aktualumą, buvo atlikta panašaus pobūdžio egzistuojančių sistemų analizė Lietuvos ir užsienio rinkose (žr. 1 skyrių). Atlikus analizę buvo iškelti kriterijai, pagal kuriuos sistemos buvo įvertintos. Įvertinus sistemas buvo iškelti reikalavimai naujai sistemai, bei parengtas planas sistemos realizavimui (žr. 2 skyrių).

Darbo rezultatas yra sukurta maitinimo įstaigų valdymo sistema (žr. 3 skyrių). Ši sistema turi tris vartotojų lygius: paprastas vartotojas - klientas, maitinimo įstaigos atstovas, ir administratorius. Maitinimo įstaigos atstovas gali užregistruoti savo įstaigą sistemoje, pateikti pagrindinę informaciją, įsivesti meniu, nusibraižyti staliukų schemą, gauti užsakymus iš klientų, bei juos apdoroti. Klientas gali ieškoti tarp maitinimo įstaigų, peržiūrėti meniu, bei staliukų schemą, daryti užsakymus ir rezervacijas. Administratorius turi daugiausiai teisių sistemoje: jis gali matyti, redaguoti ir kitaip tvarkyti visus vartotojų tipus ir jų saugomus duomenis sistemoje.

Sistemos realizavimui buvo naudojamas *Laravel* [5] karkasas ir *PHP* programavimo kalba. Duomenų bazės kūrimui buvo naudojamas *MySQL* reliacinės duomenų bazės varikliukas.

1. Panašių sistemų analizė

Dabartinėje Lietuvos rinkoje egzistuoja kelios panašaus pobūdžio sistemos, tačiau užsienio rinkoje panašios sistemos yra labiau pažengusios ir turi daugiau funkcionalumo. Prieš pradėdant projektuoti tokio pobūdžio sistemą yra būtina išanalizuoti panašias sistemas, išsiaiškinti jų privalumus ir trūkumus. Remiantis gautomis išvadomis galima pradėti projektuoti kitą sistemą kaip alternatyvą.

1.1. Meniu

www.meniu.lt [6] - tarp lietuviškų tinklalapių, tai - viena geriausių tokio pobūdžio sistemų.

Privalumai: turi labai didelę Lietuvos maitinimo įstaigų duomenų bazę. Įstaigos išskirstytos pagal kategorijas: restoranai, kavinės, picerijos, vyninės, alinės, greitas maistas, maistas į namus. Įstaigas galima filtruoti pagal virtuvės tipą (europinė, kiniečių, azijos, lietuvių ir kt.), ir pagal gaminių patiekalų tipą (kepsniai, picos, kebabai, žuvis ir kt.). Yra rodomas kainos lygis penkių balų skalėje. Pasirinkus kurią nors įstaigą parodoma visa pagrindinė informacija, nuotraukos, galimi atsiskaitymo būdai. Yra galimybė palikti atsiliepimą ir įvertinimą.

Trūkumai: Staliuko rezervacija yra galima tik nedidelei daliai rodomų įstaigų ir ji vykdoma per kitą sistemą, t. y. norint rezervuoti staliuką naudotojas yra nukreipiamas į atitinkamą sistemą. Nėra galimybės iš anksto užsisakyti norimų maisto gaminių ar gėrimų, valgiaraštis pateikiamas PDF formatu ir tik kai kurių įstaigų.

1.2. Restoranai

www.restoranai.lt [9] - lietuviška maitinimo įstaigų paieškos sistema, kurioje yra tik aukštesnio lygio restoranai. Sistemoje yra ribotas funkcionalumas: nėra galimybės pažiūrėti valgiaraščio. Norint rezervuoti staliuką, yra naudojama trečios šalies sistema - *table.in*, kuri yra mokama ir daugumai restoranų nėra aktyvuota.

Privalumai: Maitinimo įstaigų filtravimas pagal virtuvės tipą (prancūzų, europinė, vokiečių, čekų ir kt.). Yra galimybė palikti įvertinimus ir komentarus apie maitinimo įstaigas.

Trūkumai: *table.in* sistemos aktyvavimas restoranams yra mokamas, todėl daugumoje restoranų nėra galimybės matyti staliukų schemas ir remiantis ją rezervuoti staliuką. Taip pat nėra galimybės matyti maisto meniu.

1.3. Kur pavalgyti

www.kurpavalgyti.lt [4] - lietuviška maitinimo įstaigų paieškos sistema, turinti seną (apie 2010m.) įstaigų duomenų bazę, todėl naujų įstaigų surasti neįmanoma. Svetainės turinys yra pasenęs.

1.4. Open table

www.opentable.com [7]- tarptautinė staliukų rezervacijos sistema. Joje yra aukšto lygio restoranai, esantys didžiausiuose pasaulio miestuose. Išsirinkus restoraną, galima matyti įstaigos pagrindinę informaciją, kainos lygį, meniu, yra galimybė rezervuoti staliuką. Staliukų schemos pažiūrėti nėra galimybės. Restoranams sistema yra mokama.

1.5. Book a table

www.bookatable.co.uk [1]- Jungtinės Karalystės rinkai skirta staliukų rezervacijos sistema. Yra galimybė rezervuoti staliuką. Meniu galima pažiūrėti tik atsisiuntus PDF formatu. Sistemoje yra didelė restoranų duomenų bazė. Restoranams sistema yra mokama.

1.6. Apibendrinimas

Atlikus panašių sistemų analizę buvo išskirti kriterijai, pagal kuriuos sistemos buvo vertinamos:

- Paieška - galimybė atlikti maitinimo įstaigų paiešką ir filtravimą
- Meniu - galimybė susikurti arba įkelti maisto meniu
- Užsakymas - galimybė užsisakyti patiekalą iš meniu
- Staliukų schema - galimybė nusibraižyti/susikurti staliukų išsidėstymo schemą
- Rezervavimas - galimybė rezervuoti staliuką
- Kainos lygis - ar rodomas kainos lygis
- Registracija - ar visas funkcionalumas yra prieinamas be registracijos
- Nemokama - ar naujos įstaigos registracija yra nemokama
- Visos įstaigos - ar skirta visoms įstaigoms, ar tik aukšto lygio restoranams
- Dizainas - prisitaikantis ir mobiliems įrenginiams tinkantis dizainas

1 lentelė. Maitinimo įstaigų kriterijų palyginimas

Kriterijus	Menu.lt	Restoranai.lt	Kurpavalgyti.lt	Opentable.com	Bookatable.co.uk
Paieška	+	+	+	+	+
Menu	+	-	-	+	+
Užsakymas	-	-	-	-	-
Staliukų schema	-	-	-	-	-
Rezervavimas	+	+	-	+	+
Kainos lygis	+	+	-	+	+
Registracija	-	-	-	+	+
Nemokama	+	+	+	+	+
Visos įstaigos	+	-	+	-	-
Dizainas	+	+	-	+	+

Žvelgiant į panašių sistemų įvertinimus (žr. 1 lentelę), galima matyti, kad Lietuvos rinkoje daugiausiai privalumų turi *Menu.lt*, o mažiausiai *Kurpavalgyti.lt*. Užsienio rinkoje *Opentable.com* ir *Bookatable.co.uk* yra labai panašios sistemos savo funkcionalumu, skirtumas tik tas, kad *Opentable.com* yra orientuota į didžiausius pasaulio miestus, o *Bookatable.co.uk* yra skirta tik Jungtinės Karalystės rinkai. Iš įvertinimų taip pat galima matyti, kad visose sistemose nėra galimybės užsisakyti iš valgiaraščio, ir matyti staliukų išsidėstymo schemas. Lietuviškos sistemos staliukų rezervacijai naudoja kitą išorinę sistemą *table.in*, kuri papildomai kainuoja restoranams, todėl tik maža dalis restoranų turi aktyvią šios sistemos paskyrą.

1.7. Funkciniai sistemos reikalavimai

Atsižvelgus į egzistuojančių sistemų privalumus bei trūkumus, naujai planuojamai kurti sistemai buvo išskirti šie funkciniai reikalavimai:

1. Vartotojų registracija ir prisijungimas.
2. Maitinimo įstaigos kūrimas ir modifikavimas.
3. Maitinimo įstaigų paieška ir filtravimas.
4. Valgiaraščio kūrimas ir modifikavimas.
5. Užsakymų kūrimas ir modifikavimas.
6. Staliukų schemas kūrimas ir modifikavimas.
7. Staliukų rezervacija.

2. Projektavimas

Prieš pradėdant kurti sistemą, reikia atlikti išsamų projektavimo procesą, išsikelti tikslus, išsirinkti kūrimo įrankius, susiplanuoti pagrindinį funkcionalumą. Taigi, šiame skyriuje apžvelgiami sistemos techniniai sprendimai, reikalavimai, pagrindinis funkcionalumas bei kūrimo įrankiai.

2.1. Techniniai sprendimai

Sistemos kūrimui nuspręsta naudoti PHP programavimo kalbą. Toks sprendimas buvo padarytas dėl gausaus karkasų ir mokomosios medžiagos pasirinkimo, taip pat dėl patikimumo ir suderinamumo su kitom technologijom.

Karkasas (angl. framework) buvo pasirinktas *Laravel* [5], kuris veikia pagal MVC (Model-View-Controller) technologiją. Šis karkasas skirtas sistemos loginės (angl. back-end) ir prezentacinės dalies (angl. front-end) kūrimui, naudojant PHP programavimo kalbą. Šis karkasas pasižymi plačia dokumentacija, kurios dėka galima greitai ir efektyviai išmokyti šio karkaso ypatumų. *Laravel* karkasas yra gana populiarus ir turi didelę bendruomenę, todėl nesunkiai galima rasti daug pavyzdžių ir tam tikrų problemų sprendimų. Taigi, šio karkaso pasirinkimą lėmė jo paprastumas naudojime, saugumas, ir tai, kad jau anksčiau buvau susipažinęs su MVC technologija, todėl lengviau buvo pereiti prie *Laravel* karkaso.

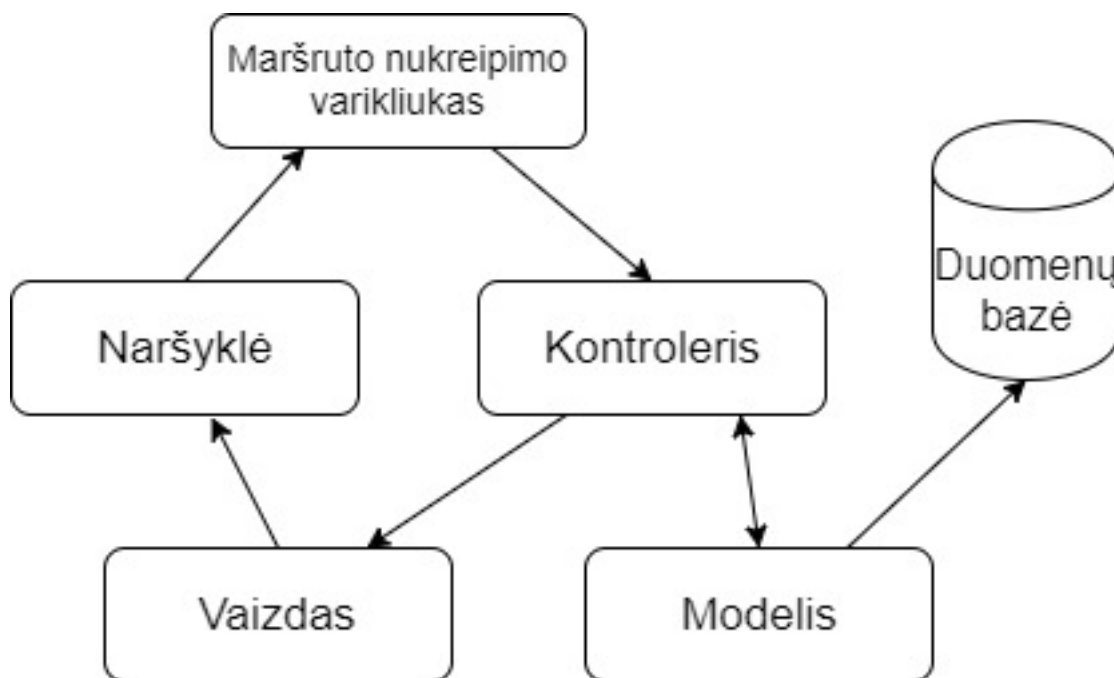
Sistemos prezentacinės dalies kūrimui buvo pasirinktas *Bootstrap* ir *Pure CSS* karkasai. Jie reikalingi tam, kad sukurti prie skirtingų įrenginių ekranų dydžių prisitaikantį dizainą. Taip pat, šie karkasai suteikia papildomų dizaino elementų - juose yra jau sukurti mygtukai, formų ir meniu elementai.

Siekiant suteikti sistemai interaktyvumo buvo naudojama *Java Script* programavimo kalba ir jos biblioteka *jQuery* [3]. Ši biblioteka suteikia sistemai *Ajax* funkcionalumą, kurio pagalba galima pilnai neperkraunant puslapio keisti jo turinį, t.y. puslapis yra perkraunamas tik dalinai. Taip pat šios bibliotekos pagalba galima suteikti papildomo dinamiškumo sistemai.

Duomenų bazės įgyvendinimui buvo pasirinktas MySQL įrankis dėl jo gero suderinamumo su PHP *Laravel* karkasu, paprastumo naudojime, patikimumo, ir dėl turimos patirties dirbant su SQL duomenų bazėmis.

2.2. Sistemos veikimo architektūra

Sistema veikia pagal *Laravel* karkaso MVC technologiją. 1 pav. pavaizduota diagrama parodo kokie komponentai sudaro MVC ir kaip jie sąveikauja tarpusavyje.



1 pav. MVC veikimo schema ir komponentai

- Naršyklė - tai bet kuri interneto naršyklė, kurioje atvaizduojamas sistemos turinys matomas galutiniam naudotojui.
- Maršruto nukreipimo varikliukas - atsako už užklausos nukreipimą į reikiamą kontrolierį.
- Kontrolieris - apdoroja užklausas, paruošia duomenis atvaizdavimui.
- Modelis - atsako už tam tikrą duomenų bazės dalį, sąveikauja su ją.
- Vaizdas - atsako už duomenų atvaizdavimą naudotojui.
- Duomenų bazė - MySQL duomenų bazė, kurioje saugomi duomenys.

Sistemos naudotojui atlikus tam tikrą veiksmą naršyklėje yra nusiunčiama užklausa į maršruto nukreipimo varikliuką, kuris nukreipia užklausa į reikiamą kontrolierį. Kontrolieris apdoroja užklausa, jeigu reikia kreipiasi į modelį, modelis savo ruožtu kreipiasi į duomenų bazę atlikdamas jos pakeitimus. Toliau kontrolieris gauna duomenis iš duomenų bazės, paruošia juos, ir išsiunčia į vaizdo komponentą. Vaizdas paruošia duomenis atvaizdavimui, sukuria vaizdą ir išsiunčia į naršyklę, kuri atvaizduoja turinį galutiniam naudotojui.

2.3. Sistemos aktoriai

Sistemoje yra keturi naudotojų tipai (aktoriai): neprisijungęs naudotojas, klientas, restorano atstovas ir administratorius. Šių keturių naudotojų tipų teisės sistemoje skiriasi.

Neprisijungęs naudotojas gali peržiūrėti visas įstaigas, atlikti jų paiešką ir filtravimą. Išsirinkus vieną įstaigą, gali peržiūrėti jos pateikiamą informaciją, valgiaraštį ir staliukų schemą.

Klientas - tai užsiregistravęs ir prisijungęs naudotojas, turintis daugiau teisių sistemoje. Jis turi visas neprisijungusio naudotojo teises ir papildomai dar gali daryti užsakymus iš valgiaraščio ir

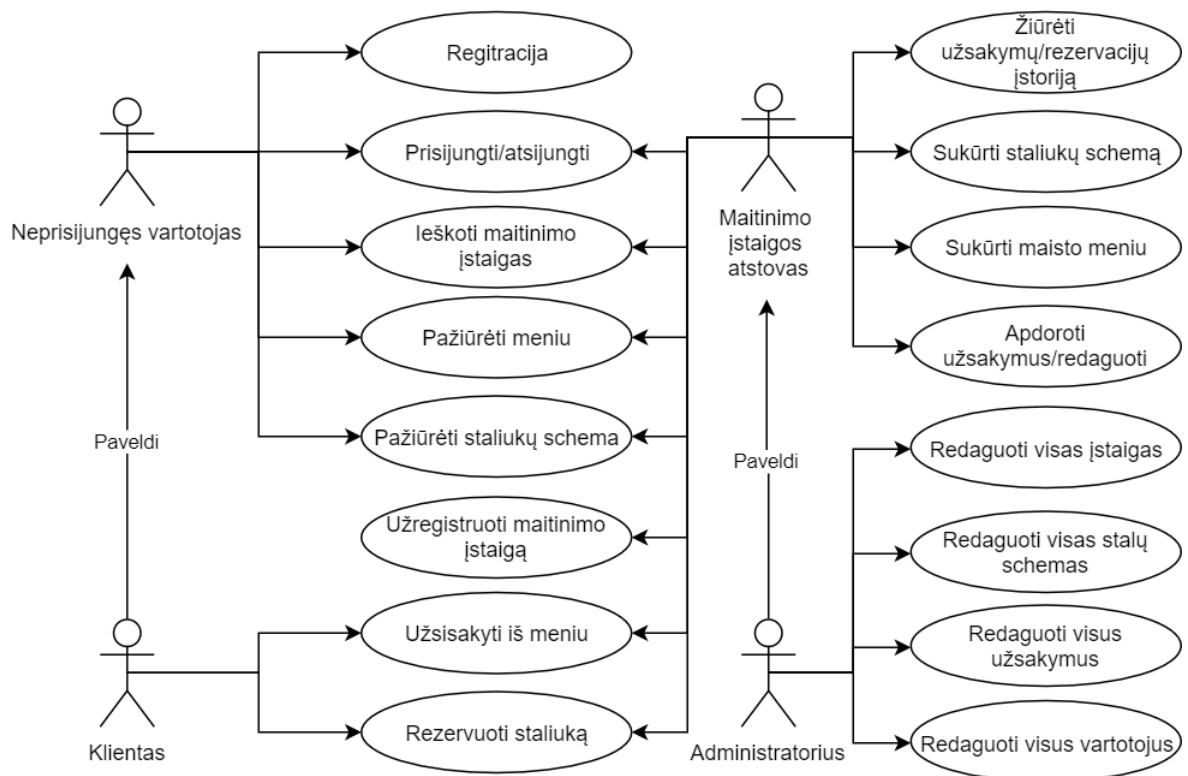
rezervuoti staliuką.

Restorano atstovas - tai yra prisijungęs naudotojas, kuris gali užregistruoti savo restoraną, pateikti pagrindinę informaciją apie jį, kontaktinius duomenis, darbo laiką, įkelti nuotraukų, sukurti staliukų išsidėstymo schemą, sukurti valgiaraštį. Taip pat atstovas gali matyti klientų pateiktų užsakymų ir stalų rezervacijų sąrašą ir juos apdoroti.

Administratorius turi daugiausiai teisių, jis turi visas prieš tai minėtų naudotojų teises bei papildomai gali matyti visus klientų užsakymus, keisti visų įstaigų informaciją, keisti kitų naudotojų informaciją ir teises.

2.4. Panaudos atvejų diagrama

Panaudos atvejų diagramoje (žr. 2 pav.) pavaizduota kokius veiksmus gali atlikti kiekvienas naudotojų tipas.



2 pav. Panaudos atvejų diagrama

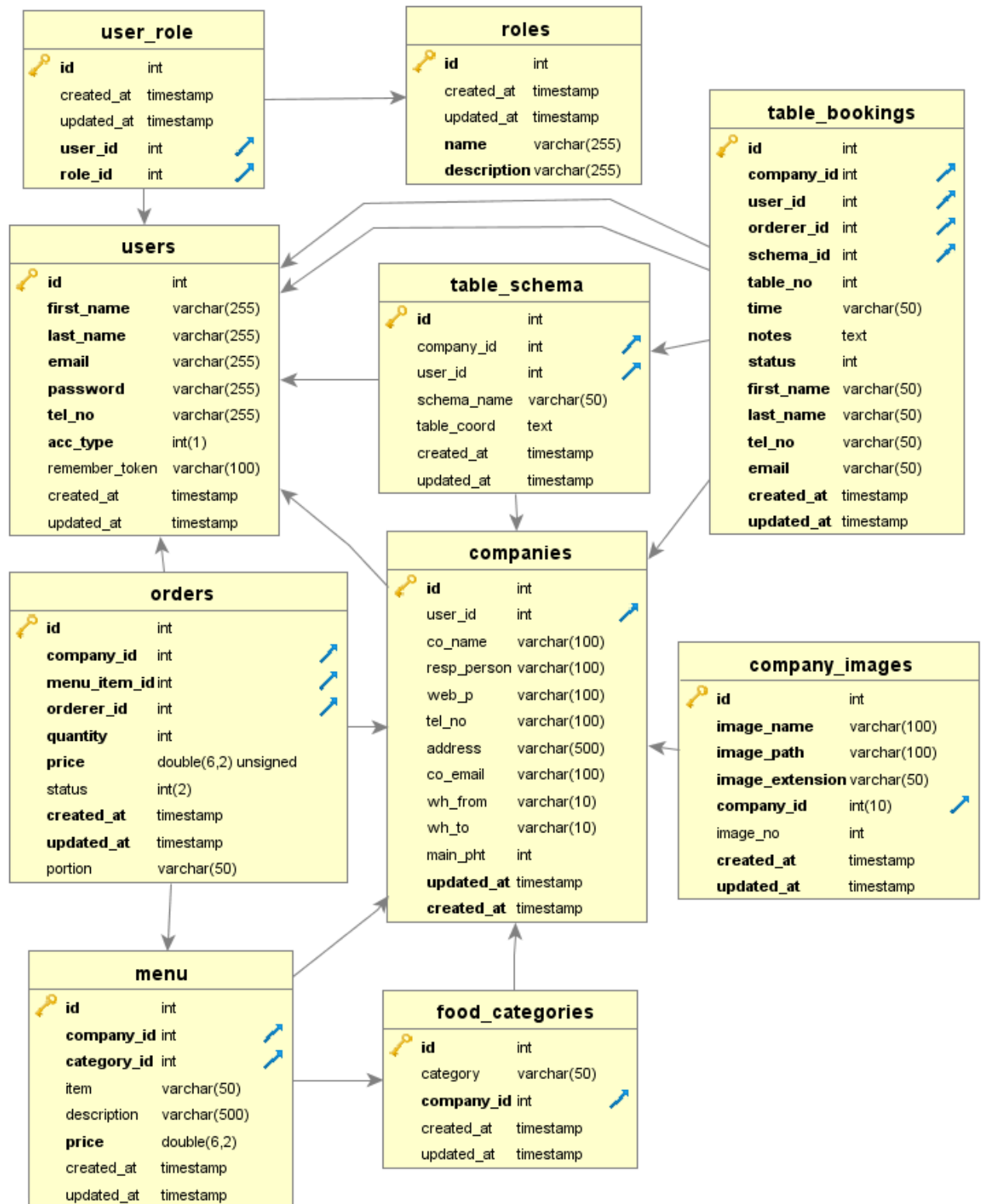
- Registracija - registruotis gali neprisijungęs naudotojas, registracijos metu galima pasirinkti vieną iš dviejų paskyros tipų - *klientas* arba *maitinimo įstaigos atstovas*. Registracijos metu reikės pateikti asmeninius duomenis tokius kaip vardas, pavardė, elektroninio pašto adresas.
- Maitinimo įstaigos sukūrimas - sukurti maitinimo įstaigą gali užsiregistravę ir prisijungę maitinimo įstaigos atstovai. Yra galimybė sukurti neribotą, maitinimo įstaigų ir jos padalinių, skaičių.

- Ieškoti maitinimo įstaigos - sistemoje visiems naudotojų tipams leidžiama ieškoti ir rūšiuoti maitinimo įstaigas pagal pavadinimą, darbo valandas, ir kitus kriterijus.
- Pažiūrėti valgiaraštį - visiems naudotojų tipams leidžiama funkcija - siūlomo maisto menių peržiūrėjimas.
- Pažiūrėti staliukų schemą - visiems naudotojų tipams leidžiama funkcija - įstaigos staliukų išsidėstymo schemos peržiūra.
- Užsisakyti iš valgiaraščio - tik prisijungusiems naudotojams leidžiama funkcija, leidžianti užsisakyti patiekalus iš valgiaraščio.
- Rezervuoti staliuką - tik prisijungusiems naudotojams leidžiama funkcija, leidžianti rezervuoti staliuką remiantis staliukų schema.
- Žiūrėti užsakymų/rezervacijų istoriją - restorano atstovui leidžiama žiūrėti savo įstaigos užsakymų/rezervacijų istoriją. Administratoriui leidžiama žiūrėti visų įstaigų užsakymų/rezervacijų istoriją.
- Sukurti staliukų schemą - įstaigos atstovui ir administratoriui leidžiama funkcija nusibrėžti staliukų schemą.
- Sukurti maisto valgiaraštį - įstaigos atstovui ir administratoriui leidžiama funkcija susikurti maisto valgiaraštį. Kuriant naują valgiaraščio įrašą yra prašoma pateikti patiekalo pavadinimą, kainą, maisto kategoriją ir aprašymą taip pat galima pasirinktinai pridėti nuorodą į paruošimo receptą.
- Apdoroti užsakymus/redaguoti - įstaigos atstovas ir administratorius gali matyti, redaguoti ir apdoroti klientų užsakymus ir rezervacijas, gali keisti jų statusus.
- Redaguoti visas įstaigas - tik administratoriui leidžiama funkcija, kuri suteikia galimybę redaguoti visų įstaigų duomenis.
- Redaguoti visus užsakymus - tik administratoriui leidžiama funkcija, kuri suteikia galimybę keisti visų užsakymų duomenis ir statusą.
- Redaguoti visus naudotojus - tik administratoriui leidžiama funkcija, kuri suteikia galimybę keisti visų naudotojų duomenis ir teises.

2.5. Duomenų bazės architektūra

Kaip ir buvo minėta anksčiau, duomenų bazės realizavimui pasirinktas *MySQL* įrankis. Duomenų bazėje saugomas visi sistemos duomenys. *Laravel* karkase yra nemažai įrankių ir pagalbininkų darbui su duomenų bazėmis palengvinimui, vienas iš jų - *Eloquent ORM*, kurio pagalba galima sąveikauti su duomenų baze nerašant didelių SQL užklausų. Šis pagalbininkas suteikia paprastesnę sintaksę operacijoms su duomenų baze atlikti (duomenų nuskaitymas, įterpimas, redagavimas, trynimasis). Kitas dažnai naudojamas pagalbininkas yra *Migrations* - tai tarsi duomenų bazės struktūros versijavimo įrankis, kuris praverčia dirbant komandoje, kai reikia sinchronizuoti duomenų bazės struktūrą.

3 pav. pavaizduota duomenų bazės schema. Kiekvienoje lentelėje yra laukai *created_at* ir *updated_at*. Šie laukai sukuriami automatiškai naudojant *Laravel* pagalbininką *Migrations*. Šie laukai užsipildo automatiškai sukuriant arba atnaujinant objektą.



3 pav. Duomenų bazės schema

Lentelėje *users* (žr. 2 lentelę) saugomi duomenys apie naudotojus. Įrašas *acc_type* parodo kokio tipo yra naudotojas: paprastas naudotojas ar maitinimo įstaiga. Įrašas *remember_token* skirtas *Atsiminti mane* (angl. *Remember me*) funkcionalumui įgyvendinti. Šiame įrašė saugomas sesijos raktas, pagal kurį į sistemą galima prisijungti be slaptažodžio įvedimo. Šis raktas galios tol, kol nebus atsijungta iš sistemos.

2 lentelė. Vartotojų lentelė (*users*)

Stulpelis	Duomenų tipas	Paaiškinimas
id	int(10)unsigned	Pirminis raktas
first_name	varchar(255)	Vartotojo vardas
last_name	varchar(255)	Vartotojo pavardė
email	varchar(255)	El. pašto adresas
acc_type	int(1)	Vartotojo tipas. 0 -> klientas, 1 -> įstaiga
remember_token	varchar(100)	Sesijos žymė

Lentelėje *roles* (žr. 3 lentelę) saugomos naudotojų rolės: administratorius, maitinimo įstaiga arba klientas.

3 lentelė. Rolių lentelė (*roles*)

Stulpelis	Duomenų tipas	Paaiškinimas
id	int(10)unsigned	Pirminis raktas
name	varchar(255)	Rolės pavadinimas
description	varchar(255)	Rolės aprašymas

Lentelė *user_role* (žr. 4 lentelę) sujungia lenteles *roles* ir *users*, joje saugomi duomenys apie tai, kokias roles turi naudotojai.

4 lentelė. Vartotojus ir roles sujungianti lentelė (*user_role*)

Stulpelis	Duomenų tipas	Paaiškinimas
id	int(10)unsigned	Pirminis raktas
user_id	int(10)unsigned	Išorinis raktas -> users.id
role_id	int(10)unsigned	Išorinis raktas -> roles.id

Lentelėje *companies* (žr. 5 lentelę) saugoma informacija apie maitinimo įstaigą. Lauke *user_id* saugoma maitinimo įstaigą sukūrusio naudotojo ID, *co_name* - įstaigos pavadinimas, *resp_person* - įstaigos atstovo vardas ir pavardė, *web_p* - įstaigos interneto tinklalapis, *tel_no* - telefono numeris, *address* - įstaigos adresas, *wh_from* - darbo laikas nuo, *wh_to* - darbo laikas iki, *main_pht* - pagrindinės nuotraukos indeksas.

5 lentelė. Maitinimo įstaigų lentelė (*companies*)

Stulpelis	Duomenų tipas	Paaiškinimas
id	int(10)	Pirminis raktas
user_id	int(10)unsigned	Sukūrusio naudotojo ID. Išorinis raktas -> users.id
resp_person	varchar(100)	Atstovo vardas ir/ar pavardė
web_p	varchar(100)	Internetinės svetainės adresas
tel_no	varchar(100)	Telefono numeris
address	varchar(500)	Įstaigos fizinis adresas
wh_from	varchar(10)	Darbo valandos (nuo)
wh_to	varchar(10)	Darbo valandos (iki)
main_pht	int(11)	Išorinis raktas -> company_images.id

Lentelėje *table_schema* (žr. 6 lentelę) saugomi duomenys apie staliukų schemas. Lauke *company_id* saugomas įstaigos ID, kuriai priklauso atitinkama schema, *user_id* - schemą sukūrusio naudotojo ID, *schema_name* - schemas pavadinimas, *table_coord* - staliukų išsidėstymo koordinatės.

6 lentelė. Staliukų schemas (*table_schema*)

Stulpelis	Duomenų tipas	Paaiškinimas
id	int(11)	Pirminis raktas
company_id	int(11)	Išorinis raktas -> companies.id
user_id	int(11)	Išorinis raktas -> users.id
schema_name	varchar(50)	Schemas pavadinimas
table_coord	text	Staliukų išsidėstymo koordinatės

Lentelėje *company_images* (žr. 7 lentelę) saugomi duomenys apie įmonių nuotraukas. Lauke *image_path* saugomas kelias iki katalogo, kuriame saugomos nuotraukos, *image_extension* - nuotraukos išplėtimas (pvz.: jpg, jpeg, png), *company_id* - išorinis raktas, parodantis kuriai įstaigai priklauso nuotrauka, *image_no* - nuotraukos numeris.

7 lentelė. Nuotraukos (*company_images*)

Stulpelis	Duomenų tipas	Paaiškinimas
id	int(11)	Pirminis raktas
image_path	varchar(100)	Kelias iki nuotraukos direktorijos
image_extension	varchar(50)	Nuotraukos išplėtimas
company_id	int(10)	Išorinis raktas -> companies.id
image_no	int(11)	Nuotraukos numeris

Lentelėje *food_categories* (žr. 8 lentelę) saugomi maisto kategorijų duomenys. Laukas *company_id* parodo kokiai įstaigai priklauso maisto kategoriją, lauke *category* saugomas pačios kategorijos pavadinimas.

8 lentelė. Maisto kategorijos (*food_categories*)

Stulpelis	Duomenų tipas	Paiškinimas
id	int(11)	Pirminis raktas
company_id	int(11)	Išorinis raktas -> companies.id
category	varchar(50)	Maisto kategorijos pavadinimas

Lentelė *menu* (žr. 9 lentelę) saugo duomenis apie maitinimo įstaigų maisto ir gėrimų valgiaraščius. Šioje lentelėje talpinami visi valgiaraščio objektai. Laukas *company_id* parodo kokias įstaigas priklauso valgiaraščio objektas, *category_id* parodo kokias maisto kategorijai priklauso valgiaraščio objektas, lauke *item* saugomas objekto pavadinimas, *description* - objekto aprašymas.

9 lentelė. Maisto valgiaraštis (*menu*)

Stulpelis	Duomenų tipas	Paiškinimas
id	int(11)	Pirminis raktas
company_id	int(11)	Išorinis raktas -> companies.id
category_id	int(11)	Išorinis raktas -> categories.id
item	varchar(50)	Menu objekto pavadinimas
description	varchar(500)	Menu objekto aprašymas

Lentelėje *orders* (žr. 10 lentelę) saugomi klientų užsakymų duomenys. Laukas *company_id* parodo kuriai įstaigai priklauso užsakymas, *menu_item_id* parodo koks valgiaraščio objektas yra užsakyme, lauke *orderer_id* saugomas naudotojo ID, kuris atliko užsakymą, *quantity* - užsakymo objektų kiekis, *price* - užsakymo galutinė kaina, *status* - užsakymo būseną t.y. *aktyvus* arba *atliktas*.

10 lentelė. Užsakymai (*orders*)

Stulpelis	Duomenų tipas	Paiškinimas
id	int(11)	Pirminis raktas
company_id	int(11)	Išorinis raktas -> companies.id
menu_item_id	int(11)	Išorinis raktas -> menu.id
orderer_id	int(11)	Išorinis raktas -> users.id
quantity	int(11)	Užsakymo objekto kiekis
price	double(6,2)	Užsakymo kaina
status	int(2)	Užsakymo statusas

Lentelėje *table_bookings* (žr. 11 lentelę) saugomi staliukų rezervacijų duomenis. Laukas *company_id* parodo kuriai įstaigai priklauso rezervacija. *user_id* parodo kuriam naudotojui priklauso įstaigą. *orderer_id* parodo kuris naudotojas atliko užsakymą, *schema_id* parodo kurioje schemoje buvo atlikta rezervacija, *table_no* - rezervuoto staliuko numeris, *time* - kuriam laikui klientas planuoja ateiti, *notes* - kliento pastabos, *status* - rezervacijos statusas (aktyvus arba atliktas), *first_name* - užsakovo vardas, *last_name* - pavardė, *tel_no* - telefono numeris, *email* - elektroninis paštas. Kadangi naudotojas yra prisijungęs laukai *first_name*, *last_name*, *tel_no*, *email* yra automatiškai užpildomi rezervacijos formoje, tačiau jeigu klientas pageidauja rezervaciją atlikti kito asmens vardu - šie duomenis išsaugomi rezervacijos lentelėje.

11 lentelė. Staliukų rezervacijos (*table_bookings*)

Stulpelis	Duomenų tipas	Paaiškinimas
id	int(11)	Pirminis raktas
company_id	int(11)	Išorinis raktas -> companies.id
user_id	int(11)	Išorinis raktas -> users.id
orderer_id	int(11)	Išorinis raktas -> users.id
schema_id	int(11)	Išorinis raktas -> table_schema.id
table_no	int(11)	Staliuko numeris
time	varchar(50)	Rezervacijos laikas
notes	text	Pastabos
status	int(1)	Rezervacijos statusas
first_name	varchar(50)	Vardas
last_name	varchar(50)	Pavardė
tel_no	varchar(50)	Telefono nr.
email	varchar(50)	El. paštas

3. Realizavimas

Kaip ir buvo minėta projektavimo skyriuje, sistema veikia MVC (Model-View-Controller) principu. Šiam veikimo principui įgyvendinti reikalingi valdikliai (angl. controller), rodiniai (angl. views) ir modeliai. Valdikliuose realizuojami pagrindiniai sistemos algoritmai ir duomenų apdorojimas, jie yra tarsi vidurinė sistemos ašis. Valdikliai kreipiasi į modelius kai reikalinga sąveika su duomenų baze. Rodiniai - tai duomenų atvaizdavimo šablonas, valdikliai perduoda paruoštus duomenis į rodinius, kurie ir atvaizduoja informaciją galutiniam vartotojui.

3.1. Valdikliai

Sistemoje yra keturi pagrindiniai valdikliai (žr. 4 pav.). Valdikliuose yra aprašomos funkcijos (metodai), kurias jis gali atlikti. Kiekvienai funkcijai, jeigu jos galutinė užduotis yra atvaizduoti duomenis, dažniausiai yra atskirai sukurtas rodinys, kuris atvaizduoja būtent tos funkcijos grąžinamus duomenis.

AuthController	MainController	FoodMenuCtrl	TablesCtrl
<code>postAdminAssignRoles()</code> <code>getSignUpPage()</code> <code>getSignInPage()</code> <code>postSignUp(\$request)</code> <code>postSignIn(\$request)</code> <code>getLogout()</code>	<code>showCompanies(\$request)</code> <code>showCompany(\$id)</code> <code>getEditCompany(\$id)</code> <code>updateCompany(\$id, \$request)</code> <code>deleteCompany(\$id)</code> <code>showProfile(\$id)</code> <code>updateProfile(\$id, \$request)</code> <code>showAllMyCompanies()</code> <code>getEditMyCompany(\$id)</code> <code>getCreateCompany()</code> <code>postCreateCompany(\$request)</code> <code>deletePhoto(\$id)</code> <code>chooseCmpMng(\$cmp_id)</code>	<code>showFoodMenu(\$company_id)</code> <code>createOrder(\$request)</code> <code>showOrders()</code> <code>showOrdersUser()</code> <code>showAllOrdersAdmin()</code> <code>showCreateFoodCat()</code> <code>createFoodCategory(\$request)</code> <code>foodCatMngmnt()</code> <code>createMenuEntry_show()</code> <code>createMenuEntry(\$request)</code> <code>showMenuManagement()</code> <code>editMenuItem(\$id)</code> <code>update_menu_item(\$request)</code> <code>deleteMenuItem(\$id)</code> <code>deleteFoodCategory(\$id)</code>	<code>createTableMap()</code> <code>createTableSchema(\$request)</code> <code>getEditSchema(\$id)</code> <code>updateTableSchema(\$request)</code> <code>showAllSchemas()</code> <code>deleteSchema(\$id)</code> <code>showSchema(\$id)</code> <code>showSchemaOrder(\$id)</code> <code>showAllSchemasListOrder(\$id)</code> <code>bookTable(\$request)</code> <code>getTableBookingsCmp()</code> <code>changeBookingStatus(\$request)</code> <code>deleteBooking(\$request)</code>

4 pav. Valdikliai (angl. controllers)

Valdiklis **AuthController** turi šešias funkcijas, kurios skirtos vartotojų prisijungimui prie sistemos organizuoti. Funkcija *postAdminAssignRoles()* - skirta administratoriui, kuriam nustačius vartotojui atitinkamas roles/teises sistemoje ir paspaudus patvirtinimo mygtuką suveikia ši funkcija, kuri duomenų bazėje atlieka atitinkamus pakeitimus. *getSignUpPage()* - užkrauna vartotojui registracijos puslapį. *getSignInPage()* - užkrauna vartotojui prisijungimo prie sistemos puslapį. *postSignUp(\$request)* - užkrauna formoje suvestus registracijos duomenis ir įrašo į duomenų bazę, tokių būdu sukuriamas naujas vartotojas, *\$request* parametras paduodamas tam, kad jo pagalba valdiklis galėtų pasiimti formoje suvestus duomenis. *postSignIn(\$request)* - nuskaito prisijungimo formoje suvestus duomenis, sutikrina juos su turimais duomenų bazėje ir jeigu duomenys atitinka - autorizuoja vartotoją sistemoje (vartotojas tampa prisijungęs prie sistemos). *getLogout()* - nutraukia vartotojo sesiją, vartotojas tampa atsijungęs nuo sistemos.

Valdiklis **MainController** turi 13 funkcijų ir yra skirtas klientų ir maitinimo įstaigų paskyrų tvarkymui, redagavimui, duomenų keitimui. Funkcija *showCompanies(\$request)* - grąžina į rodinį visas sistemoje turimas maitinimo įstaigas. *showCompany(\$id)* - ši funkcija suveikia paspaudus ant vienos iš maitinimo įstaigų, ji parodo tos įstaigos pagrindinę informaciją, funkcija gauna *\$id* parametą - identifikacijos numerį, pagal kurį suranda reikiamą įstaigą duomenų bazėje. *getEditCompany(\$id)* - užkrauna maitinimo įstaigos redagavimo puslapį, funkcija skirta administratoriui kuris gali redaguoti visas įstaigas. *updateCompany(\$id, \$request)* - nuskaito įstaigos redagavimo puslapyje įvestus/pakeistus duomenis ir padaro pakeitimus duomenų bazėje, funkcija skirta administratoriui. *deleteCompany(\$id)* - pašalina maitinimo įstaigą iš sąrašo, funkcija skirta administratoriui. *showProfile(\$id)* - parodo prisijungusiam vartotojui jo profilio informaciją, tame pačiame lange jis gali redaguoti profilį. *updateProfile(\$id, \$request)* - nuskaito įvestus duomenis profilio formoje ir atlieka pakeitimus duomenų bazėje. *showAllMyCompanies()* - užkrauna maitinimo įstaigų valdymo puslapį, *getEditMyCompany(\$id)* - užkrauna maitinimo įstaigos redagavimo puslapį, funkcija skirta įstaigos atstovui. *getCreateCompany()* - užkrauna maitinimo įstaigos kūrimo puslapį, *postCreateCompany(\$request)* - nuskaito maitinimo įstaigos kūrimo formoje įrašytus duomenis ir išsaugo duomenų bazėje. *deletePhoto(\$id)* - pašalina nuotrauką, *chooseCmpMng(\$cmp_id)* - funkcija naudojama kai reikia pasirinkti maitinimo įstaigą kuria norima valdyti sistemoje.

Valdiklis **FoodMenuCtrl** turi 15 funkcijų, jis skirtas maitinimo įstaigos maisto valgiaraščio ir užsakymų tvarkymui. *showFoodMenu(\$company_id)* - parodo konkrečios įstaigos maisto meniu. *createOrder(\$request)* - sukuria užsakymą iš valgiaraščio, išsaugo duomenų bazėje. *showOrders()* - maitinimo įstaigos vartotojui parodo jam priskirtus užsakymus. *showOrdersUser()* - klientui parodo jo darytus užsakymus. *showAllOrdersAdmin()* - administratoriui parodo visų įstaigų užsakymus. *showCreateFoodCat()* - užkrauna maisto kategorijos sukūrimo formą. *createFoodCategory(\$request)* - nuskaito kategorijos sukūrimo formoje įvestus duomenis ir išsaugo į duomenų bazę. *foodCatMngmnt()* - užkrauna maisto kategorijų valdymo puslapį. *createMenuEntry_show()* - užkrauna maisto meniu objekto sukūrimo formą. *createMenuEntry(\$request)* - paima iš maisto meniu objekto sukūrimo formos įvestus duomenis ir išsaugo juos į duombazę. *showMenuManagement()* - užkrauna valgiaraščio valdymo puslapį. *editMenuItem(\$id)* - užkrauna meniu objekto redagavimo formą. *update_menu_item(\$request)* - nuskaito valgiaraščio objekto redagavimo formoje įvestus duomenis ir išsaugo duomenų bazėje. *deleteMenuItem(\$id)* - ištrina meniu objektą. *deleteFoodCategory(\$id)* - ištrina maisto kategoriją.

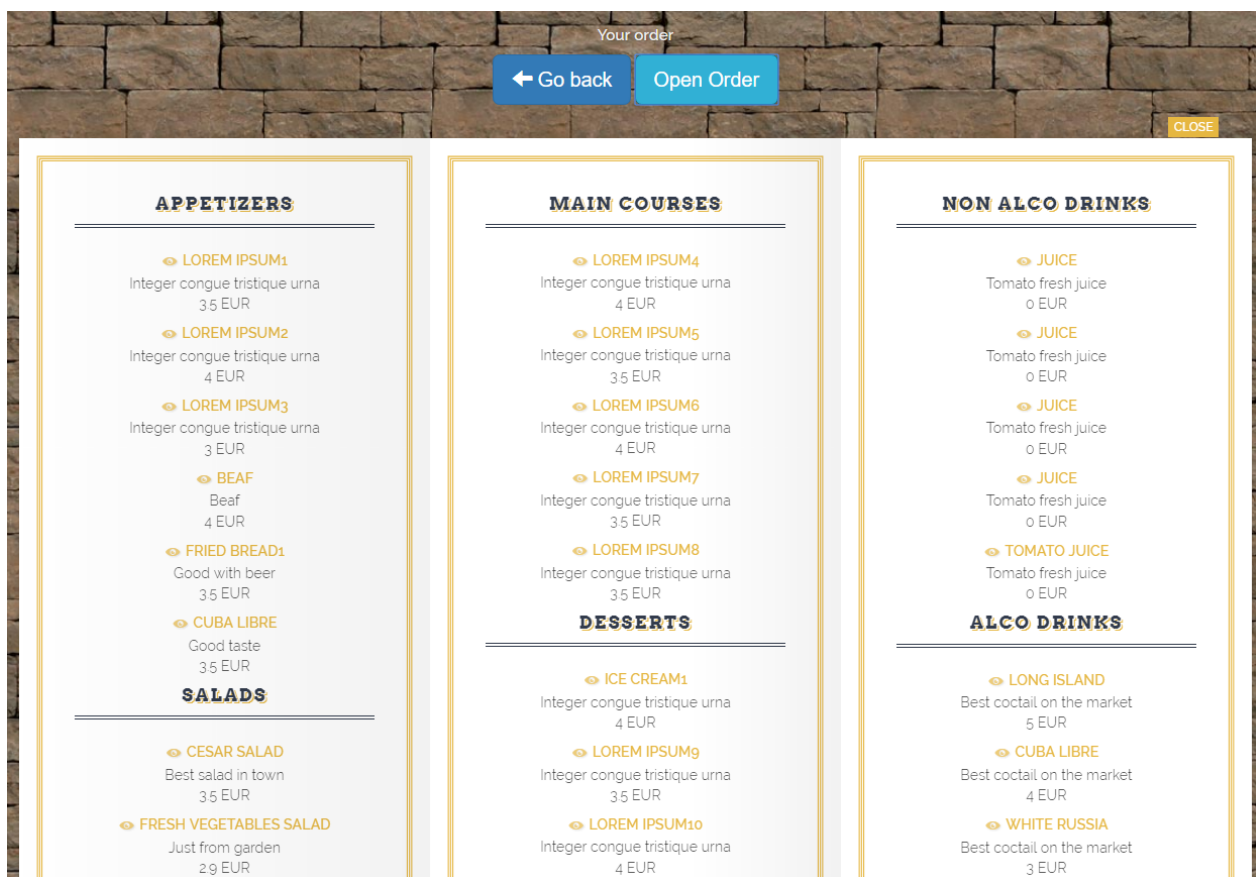
Valdiklis **TablesCtrl** turi 13 funkcijų ir yra skirtas staliukų schemoms valdyti ir apdoroti. Funkcija *createTableMap()* - užkrauna staliukų schemos sukūrimo langą. *createTableSchema* - nuskaito duomenis iš staliukų sukūrimo formos ir išsaugo į duomenų bazę. *getEditSchema(\$id)* - užkrauna schemos redagavimo langą. *updateTableSchema(\$request)* - nuskaito duomenis iš redagavimo lango ir padaro pakeitimus duomenų bazėje. *showAllSchemas()* - užkrauna įstaigos atstovui staliukų schemų valdymo puslapį. *deleteSchema(\$id)* - ištrina schemą. *showSchema(\$id)* - parodo pasirinktą staliukų schemą, be galimybės redaguoti. *showSchemaOrder(\$id)* - parodo staliukų schemą klientui, su galimybe rezervuoti staliuką. *showAllSchemasListOrder(\$id)* - parodo visų maitinimo įstaigos staliukų schemų sąrašą. *bookTable(\$request)* - surenka staliuko rezervavimo formoje įvestus duomenis ir sukuria įrašą duomenų bazėje. *changeBookingStatus(\$request)* - pakeičia rezervacijos statusą iš *Active* į *Completed* ir atvirkščiai. *deleteBooking(\$request)* - pašalina rezervaciją.

3.2. Registracija

Registruotis gali neprisijungęs naudotojas. Registracijos metu įvedami duomenis tokie kaip vardas, pavardė, el. paštas ir kt. Taip pat registracijos metu pasirenkamas naudotojo tipas: *Klientas* arba *Maitinimo įstaiga*, nuo šio pasirinkimo priklauso kokios funkcijos ir galimybės sistemoje bus suteiktos naudotojui.

3.3. Maisto ir gėrimų valgiaraštis

Maisto ir gėrimų valgiaraštį (žr. 5 pav.) pažiūrėti gali visi vartotojai, tačiau daryti užsakymus gali tik prisijungę vartotojai. Valgiaraštis yra pasiekiamas užėjus ant konkrečios maitinimo įstaigos.



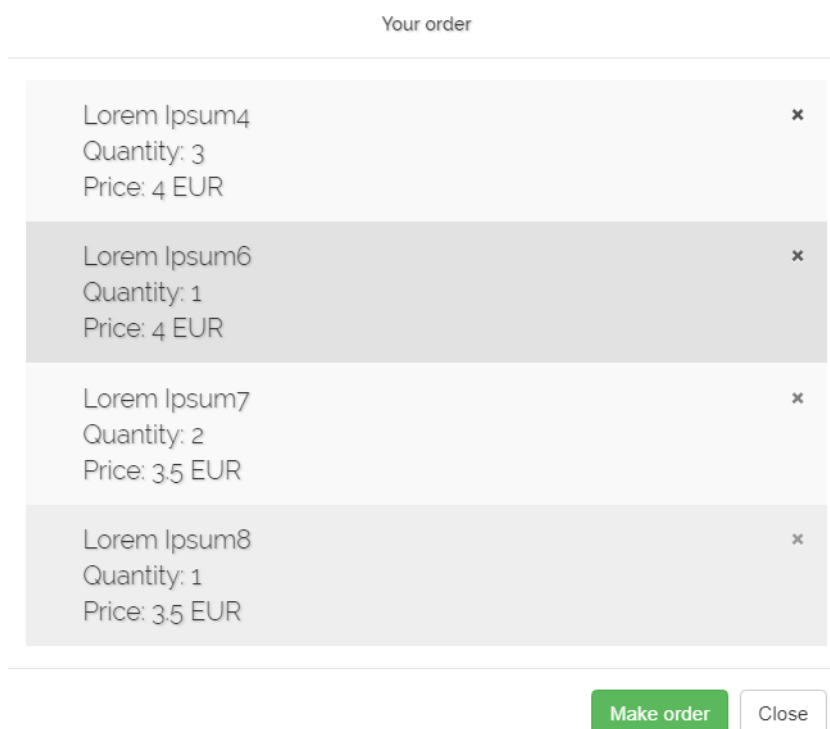
5 pav. Maisto ir gėrimų valgiaraštis

Valgiaraštis yra išskirstytas pagal maisto kategorijas. Viršui rašoma maisto kategorija, po kurios seka valgiaraščio objektai (patiekalai), kurie atitinka tą kategoriją. Užėjus ant konkretaus objekto išskyla papildomas langas (žr. 6 pav.), kuriame parodoma platesnė informaciją apie patiekalą. Tame pačiame lange yra galimybė pridėti pasirinktą patiekalą į užsakymą, tai padaryti galima pasirinkus norimą patiekalo kiekį vienetais ir paspaudus mygtuką *Add to order*.



6 pav. Meniu objekto platesnė informacija

Pridėjus norimus valgiaraščio objektus į užsakymų sąrašą, galima atlikti užsakymą, kuris pasieks maitinimo įstaigos atstovą. Tai galima padaryti paspaudus ant viršuje esančio mygtuko *Open order*. Atsidariusiame lange pasirodys užsakytų patiekalų sąrašas (žr. 7 pav.), jį peržvelgus galima atlikti užsakymą paspaudus mygtuką *Make order*.



7 pav. Užsakymo suvestinė

3.4. Staliukų išsidėstymo schemos

Kaip ir buvo minėta pagrindinių reikalavimų (1.7) skyriuje, sistemoje yra galimybė sukurti staliukų išsidėstymo schemą. Ši funkcija skirta maitinimo įstaigų atstovams, jie gali susidėlioti staliukų schemą, kuria remiantis vėliau klientai galės rezervuoti sau norimą staliuką. Schema suteikia papildomą vaizdinę informaciją apie tai, kokia aplinka yra maitinimo įstaigoje. Funkcija pasiekama naudotojo meniu pasirinkus *Table schema*, ant kurios paspaudus atidaromas schemų valdymo puslapis (žr. 8 pav.). Šiame puslapyje yra schemų sąrašas, kuriame taip pat galima kurti naujas schemas, redaguoti arba trinti jau esamas.

Table schemas

Create new table schema

Show 10 entries Search:

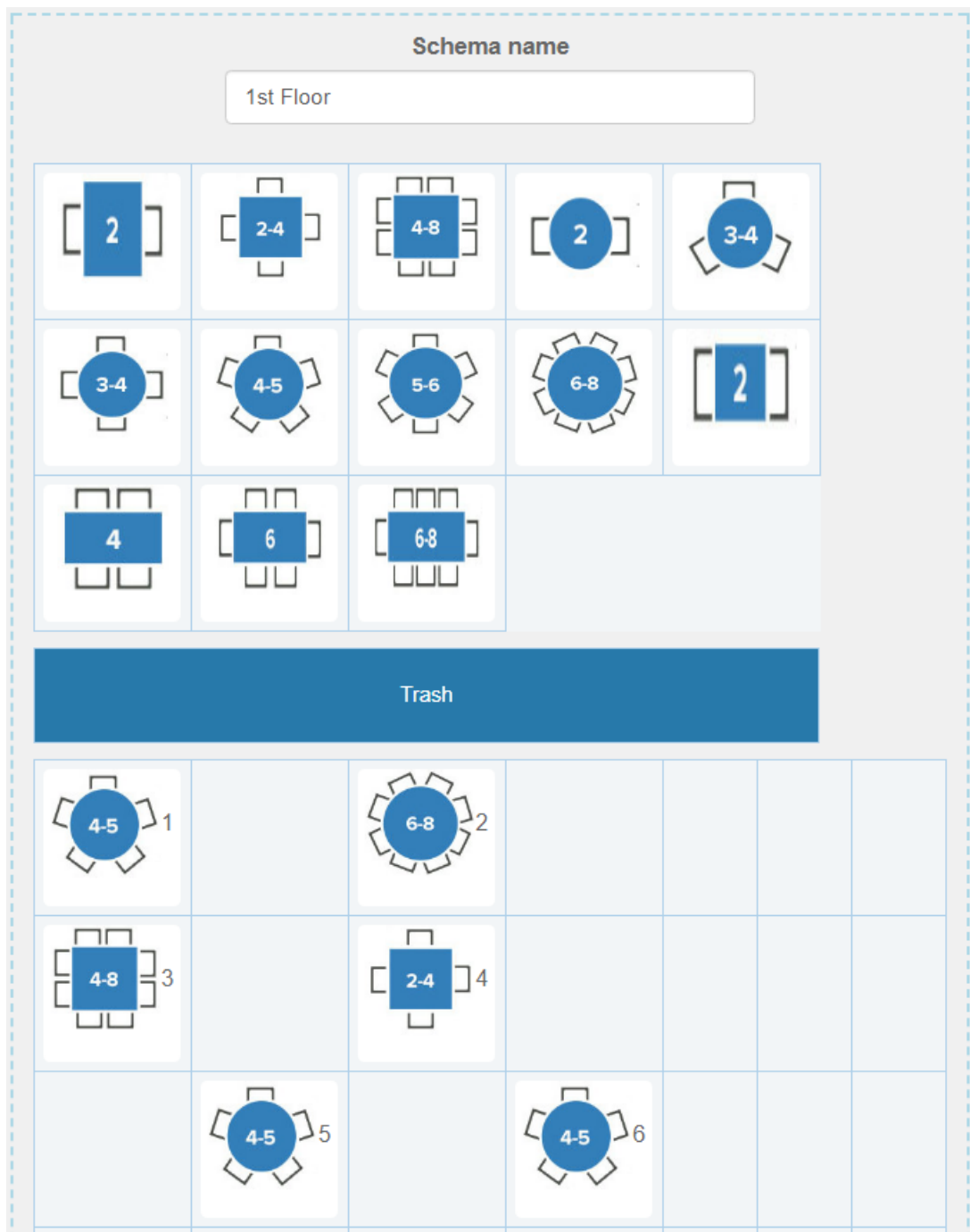
Schema name	Created at	Updated at	Edit	Delete
1st floor	2018-01-11 18:13:06	2018-01-12 00:11:03	Edit	Delete
2nd floor	2017-12-05 23:06:44	2018-01-12 00:11:11	Edit	Delete
2nf floor terasa	2017-12-29 16:35:24	2018-01-12 00:12:06	Edit	Delete
3rd floor	2017-05-19 12:19:55	2018-01-12 00:11:19	Edit	Delete
Basement	2017-12-16 23:00:03	2018-01-12 00:12:14	Edit	Delete
Terasa	2017-12-05 23:06:37	2018-01-12 00:11:41	Edit	Delete

Showing 1 to 6 of 6 entries

Previous 1 Next

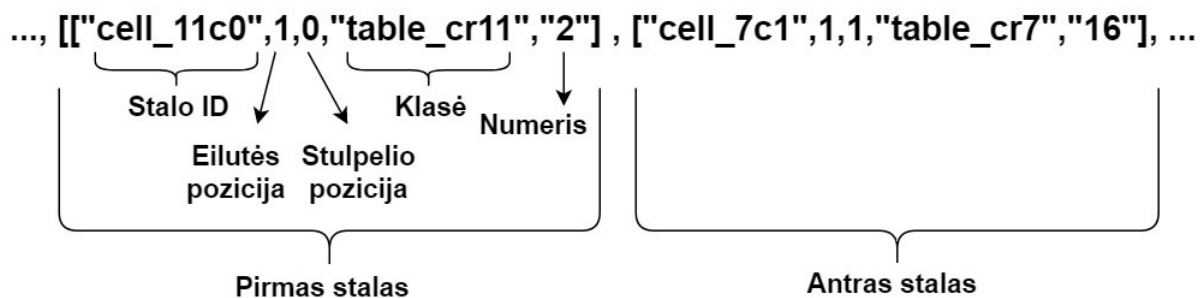
8 pav. Staliukų schemų valdymo puslapis

Norint sukurti naują schemą reikia nuspausti mygtuką *Create new table schema*. Nuspaudus atsidaro schemas kūrimo puslapis (žr. 9 pav.), kurio viršutinėje dalyje yra staliukų tipai. Staliukai yra įvairių formų ir su skirtingu sėdimų vietų skaičiumi. Sėdimos vietos yra nuo 2 iki 8 asmenų, formos yra apskritimo, keturkampio, kvadrato. Prie kiekvieno staliuko pažymėta kiek asmenų prie jo gali patogiai sėdėti. Šiek tiek žemiau staliukų pasirinkimo vietos randasi schemas braižymo (staliukų dėliojimo) vieta. Į ją galima nutempti norimą stalą, iš viršutinės dalies į apatinę.



9 pav. Staliukų schema

Nuspaudus mygtuką *Save schema* yra nuskaitomos staliukų koordinatės ir išsaugomos duomenų bazės lentelėje *table_schema* lauke *table_coord*. Ištrauka iš koordinatžių eilutės pavaizduota 10 pav.



10 pav. Staliukų koordinatų pavyzdys

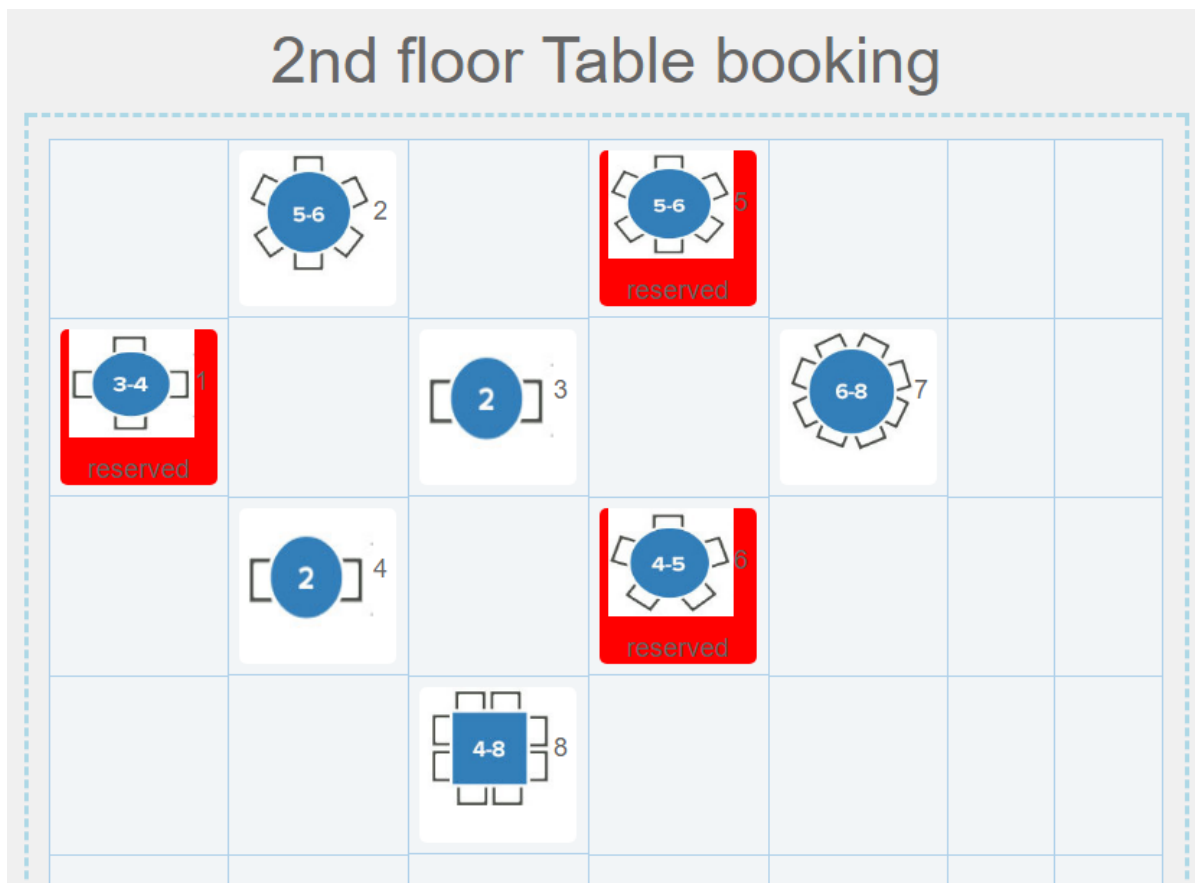
- Stalo ID - unikalus stalo identifikatorius, naudojamas HTML kalboje
- Eilutės pozicija - eilutės pozicija nuo viršaus. Pradedama skaičiuoti nuo 0
- Stulpelio pozicija - stulpelio pozicija iš kairės. Pradedama skaičiuoti nuo 0
- Klasė - stalo klasė, pagal kurią yra nustatomas stalo tipas
- Numeris - stalo numeris, skirtas atskirti stalus vienoje schemoje.

Staliukų schemos funkcionalumo kūrimui buvo naudojamas *Redips JavaScript Drag and Drop* [8] įskiepis. Šis įskiepis buvo naudojamas kaip pagrindas, jame yra sudėtos pagalbinės funkcijos *Tempk ir Paleisk* (angl. *Drag and Drop*) funkcionalumui įgyvendinti tokios kaip elementų pernešimas iš vienos vietos į kitą, elementų pozicijų išsaugojimas į masyvą, elementų pozicijų užkrovimas iš masyvo.

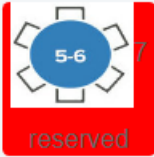
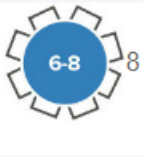

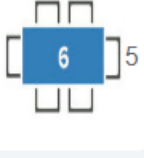
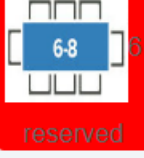
3.5. Staliukų rezervavimas

Prisijungę klientai turi galimybę rezervuoti staliuką, tai padaryti jie gali pasirinkus norimą maitinimo įstaigą, joje paspaudus ant mygtuko *Tables schema* atsiranda visos maitinimo įstaigai priklausančios staliukų schemos. Pasirinkus norimą schemą parodomas jos turinys. Viršutinėje dalyje (žr. 11 pav.) rodomi visi staliukai, rezervuoti staliukai pažymėti raudonu fonu ir užrašu *reserved*, kiti staliukai yra laisvi. Paspaudus ant norimo rezervuoti staliuko jis yra pažymimas mėlynu fonu, toliau reikia įrašyti asmeninius duomenis reikalingus rezervacijai, tokius kaip vardas, pavardė, el. paštas ir kt (žr. 12 pav.). Galiausiai reikia paspausti mygtuką *Book table*. Jei rezervacija sėkminga - parodomas atitinkamas pranešimas.

Rezervacijos atvaizduojamos tik maitinimo įstaigos naudotojui matomame rezervacijų sąrašė, jame galima matyti visą užsakovo pateiktą informaciją, taip pat galima keisti rezervacijos būseną iš *Active* į *Completed*. Kol užsakymo būseną yra *Active* tol staliukas schemoje rodomas kaip rezervuotas, kai būseną yra pakeičiama į *Completed* - staliukas rodomas kaip laisvas.



11 pav. Staliukų rezervavimas (viršutinė dalis)

Choose the table by pressing on it.

Chosen table: 9

Time

12:30

First Name

Vardenis

Last Name

Pavardenis

Tel no

+37067600763

Email

vardenis@gmail.com

Notes

Leave your notes here

[Back](#) [Book table](#)

12 pav. Staliukų rezervavimas (apatinė dalis)

3.6. Maitinimo įstaigų valdymas

Dažnai maitinimo įstaigos turi ne vieną, bet kelis padalinius, kiekvienam padaliniui registruoti po atskirą paskyrą būtų nepatogu. Sistemoje yra galimybė sukurti neribotą skaičių maitinimo įstaigų arba jos padalinių. Tai padaryti galima maitinimo įstaigų valdymo puslapyje (žr. 13 pav.) paspaudus ant mygtuko *Create new company / branch*. Kuriant naują įstaigą arba padalinį reikia nurodyti duomenis tokius kaip pavadinimas, darbo laikas, adresas, įkelti įstaigos nuotrauką ir kt.




Vienu metu sistemoje galima valdyti vieną iš turimų įstaigų/padalinių. Tam, kad pasirinkti norimą valdyti įstaigą/padalinį reikia valdymo puslapyje prie atitinkamos įstaigos paspausti mygtuką *Manage*, tuomet pasirinkta įstaiga tampa aktyvuota. Tokiu būdu galima valdyti tik tos įstaigos staliukų schemas, valgiaraštį bei matyti užsakymus ir rezervacijas. Norint valdyti kitą įstaigą/padalinį reikia vėl atlikti įstaigos pasirinkimo veiksmą įstaigų valdymo puslapyje.

My companies / branches

Create new company / branch

Show 10 entries

Search:

Image	Company	Actions	Address	Responsible	Tel	Delete
	Restoranas 1	<div>Edit</div> <div>Manage</div>	Vilniaus g. 31, Vilnius	Ausra	+37067600769	<div>Delete</div>
	Restoranas 2	<div>Edit</div> <div>Manage</div>	Kauno g. 3, Vilnius	Vardenis	+37067600769	<div>Delete</div>
	Restoranas 3	<div>Edit</div> <div>Manage</div>	Elektrenų g. 20, Skuodas	Pavardene	+37067600767	<div>Delete</div>

13 pav. Įstaigų valdymo puslapis

Išvados ir rekomendacijos

Atlikus egzistuojančių maitinimo įstaigų valdymo sistemų analizę Lietuvos bei užsienio rinkose ir nustačius jų stipriąsias, bei silpnąsias savybes buvo išskirti vertinimo kriterijai, pagal kuriuos sistemos buvo vertinamos. Vėliau pagal šiuos kriterijus buvo išskirti reikalavimai alternatyvios sistemos kūrimui. Lietuvos rinkoje buvo rasti keli stipresni projektai, tačiau ir šie neatitiko visų iškeltų reikalavimų.

Darbo pabaigoje buvo sukurta sistema, skirta maitinimo įstaigoms ir jų klientams. Sistemoje galima ieškoti maitinimo įstaigų, jas filtruoti, daryti užsakymus, bei staliukų rezervacijas. Restorams galima įsivesti valgiaraštį ir nusibrėžti staliukų schemą. Sistema padeda maitinimo įstaigoms ir jų klientams greičiau surasti vienas kitą.

Siekiant sistemą paleisti į rinką, pilnam jos funkcionavimui ir sėkmingam gyvavimui, reiktų atlikti šiuos patobulinimus:

- Integruoti autorizaciją per socialinius tinklus (pvz. Google, Facebook), tai pagreitintų registracijos laiką ir neatbaidytų vartotojų dėl registracijos formų pildymo.
- Užsakymo apmokėjimo įdiegimas. Tokiu būdu vartotojai galėtų iš anksto padaryti užsakymo apmokėjimą naudojantis sistema. Tai sumažintų melagingų užsakymų skaičių.
- SSL sertifikato įdiegimas. Šis sertifikatas suteikia papildomo saugumo komunikuojant su serveriu, kadangi siunčiami duomenys yra šifruojami.
- Sistemos prezentacinės dalies (galutiniam vartotojui matomos) dizaino tobulinimas, kad sistemą kuo daugiau atitiktų UI/UX [2] keliamus principus ir pritrauktų daugiau vartotojų.
- Staliukų braižymo funkcionalumo plėtimas - sienų, langų, kitų objektų žymėjimas schemoje, tai suteiktų klientams daugiau informacijos apie aplinką.

Literatūros sąrašas

- [1] Bookatable. Žiūrėta 2017-12-10
<https://www.bookatable.co.uk>
- [2] Freshconsulting.com straipsnis apie dizaino kūrimo sprendimus. Žiūrėta 2017-01-10
<https://www.freshconsulting.com/7-uiux-principles-fresh-web-experiences/>
- [3] JQuery dokumentacija. Žiūrėta 2017-10-15
<http://api.jquery.com>
- [4] Kur pavalgyti. Žiūrėta 2017-12-10
<http://www.kurpavalgyti.lt>
- [5] Laravel PHP karkaso dokumentacija. Žiūrėta 2017-10-15
<https://laravel.com/docs/5.5>
- [6] Meniu - restoranų gidas. Žiūrėta 2017-12-10
<https://www.meniu.lt>
- [7] Opentable. Žiūrėta 2017-12-10
<https://www.opentable.com>
- [8] Redips Drag and Drop įskiepis. Žiūrėta 2017-12-10
<https://www.redips.net/javascript/drag-and-drop-table-content/>
- [9] Restoranai. Žiūrėta 2017-12-10
<http://www.restoranai.lt>