



KAUNO TECHNOLOGIJOS UNIVERSITETAS
Informatikos fakultetas

T120B165 Saityno taikomųjų programų projektavimas

Projekto „Nenupirkto maisto svetainė“ ataskaita

Dėstytojas:
Tomas Blažauskas

Studentė:
Ieva Liktaravičiūtė, IFF-9/4

KAUNAS, 2022

Turinys

1.	Sprendžiamo uždavinio aprašymas	3
1.1.	Sistemos paskirtis	3
1.2.	Funkciniai reikalavimai	3
1.3.	Sistemos architektūra	4
2.	API sąsajos funkcijos	5
2.1.	API užklauskos	5
2.1.1.	Autorizacija	5
2.1.2.	Restoranai	6
3.	API realizacija.....	9
3.1.	Naudotojo sąsaja.....	9
4.	Išvados	24
	Paveikslėlių sąrašas	25
	Lentelių sąrašas	26

1. Sprendžiamo uždavinio aprašymas

1.1. Sistemos paskirtis

Projekto tikslas – suteikti galimybę maisto gamintojams parduoti nenupirktą maistą.

Veikimo principas – kuriamą sistemą sudaro dvi dalys: internetinė aplikacija, kuria naudosis maisto gamintojai, norintys parduoti nenupirktą maistą, vartotojai, siekiantys įsigyti maistą, administratorius bei aplikacijų programavimo sąsaja.

Maisto gamintojas, norėdamas naudotis nenupirkto maisto svetaine „Likučiai“, turės prisiregistruoti prie internetinės svetainės ir galės padaryti įrašą apie nepasiektą, nenupirktą, pagal klaidingą užsakymą pagamintą maisto prekę ar patiekalą. Vartotojas, prisiregistravęs prie sistemos, matys visus restoranų pasiūlymus. Vartotojas, radęs jį dominantį pasiūlymą, atliks pirkimą, įrašas bus pašalintas iš sąrašo (kol kas sistema nepalaiko pirkimo funkcijos). Tada vartotojas galės nukeliavęs į įrašę nurodytą adresą atsiimti maistą. Administratorius galės šalinti įrašus, pridėti, bei redaguoti patiekalus.

1.2. Funkciniai reikalavimai

Neregistruotas sistemos naudotojas galės:

1. Peržiūrėti platformos pagrindinį puslapį.
2. Prisiregistruoti prie internetinės aplikacijos.

Registruotas sistemos naudotojas galės:

1. Atsijungti nuo internetinės aplikacijos.
2. Prisijungti prie svetainės.
3. Sukurti naują įrašą.
4. Peržiūrėti įkeltus įrašus.
5. Atlikti pirkimą;

Administratorius galės:

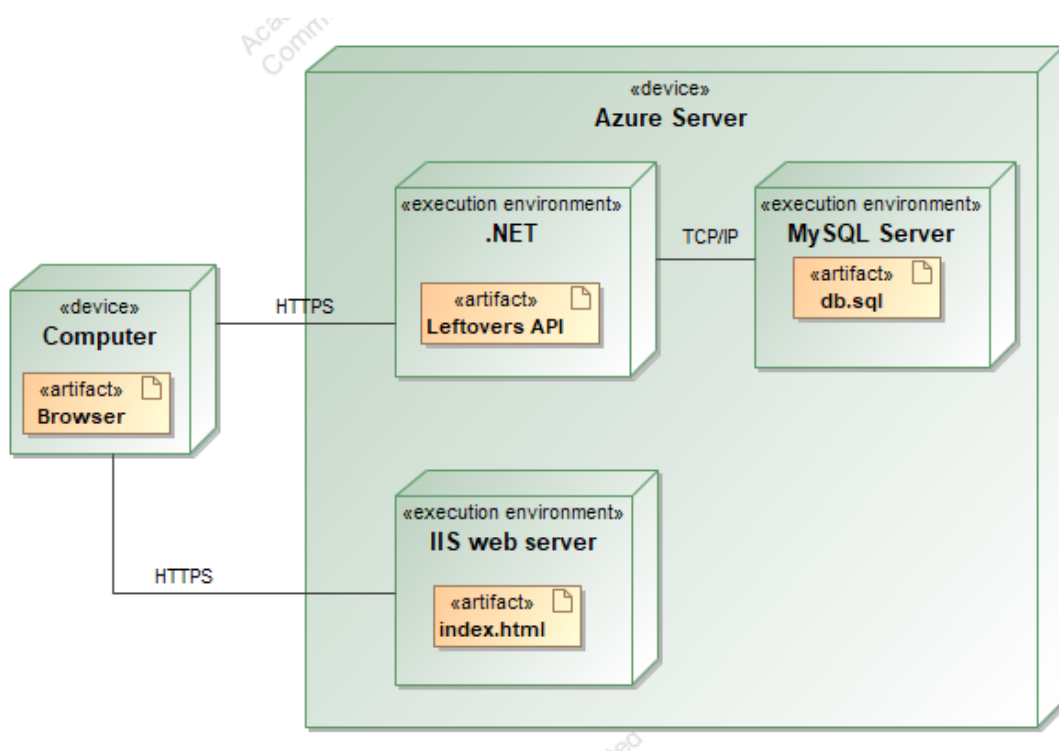
1. Šalinti įrašus.
2. Šalinti vartotojų paskyras.

1.3. Sistemos architektūra

Sistemos sudedamosios dalys:

- Kliento pusė (ang. Front-End) – naudojant React.js;
- Serverio pusė (angl. Back-End) – naudojant C# .NET. Duomenų bazė – MySQL.

1 pav. pavaizduota kuriamos sistemos diegimo diagrama. Sistemos talpinimui yra naudojamas Azure serveris. Kiekviena sistemos dalis yra diegiama tame pačiame serveryje. Internetinė aplikacija yra pasiekama per HTTPS protokolą. Šios sistemos veikimui yra reikalingas Leftovers API, kuris pasiekiamas per aplikacijų programavimo sąsają. Pats Leftovers API vykdo duomenų mainus su duomenų baze - tam naudojama TCP/IP sąsaja.



pav. 1 Nenupirkto maisto sistemos diegimo diagrama

2. API sąsajos funkcijos

Pirmas žingsnis kuriant vartoto sąsają buvo jos planavimas. Buvo nuspręsta pasirinkti paprastą dizainą, kai duomenys pateikiami sąrašu, tarp langų naviguojama pagal pasirinktų taikomosios srities objektų hierarchiją (restoranų tinklas <- restoranas <- patiekalas). Trečio laboratorinio darbo metu realizavau šį planą pasinaudojant „React.js“.

Duomenų saugojimui, po sėkmingo sistemos įkėlimo į saityną, buvo pradėta naudoti „Azure“ duomenų saugykla.




Sąsajos funkcijoms kviesti bei testuoti buvo pasirinkta naudoti „Postman“ aplikaciją. Jos pagalba galėjau ištestuoti visų užklausų korektiškumą ir įsitikinti jų teisingu veikimu.

2.1. API užklausos




Igyvendintos API užklausos autentifikacija bei autorizacija naudojant JWT technologinį sprendimą.

2.1.1. Autorizacija

Lentelė 1 (Post) Register



POST	Register
URL (teisingas)	https://leftoversapi.azurewebsites.net/api/register
Užklausos pvz. (JSON):	Atsako pvz. (JSON):
Kai vartotojas egzistuoja: <pre>{ "UserName": "Naujas", "Email": "naujas@gmail.com", "Password": "Naujas123!"}</pre>	 400 Bad Request 11.41 s 528 B 1 User already exists
Kai vartotojas neegzistuoja: <pre>{ "UserName": "TikraiNaujas", "Email": "naujas@gmail.com", "Password": "Naujas123!"}</pre>	 201 Created 1305 ms 348 B { "id": "a7f00992-8692-4a63-ae5a-681002af0b11", "userName": "TikraiNaujas", "email": "naujas@gmail.com"}
Kai neteisingai užpildyti duomenys: <pre>{ "UserName": "Nesilaikoformato", "Email": "naujas@gmail.com", "Password": "nesilaikoformato"}</pre>	 400 Bad Request 167 ms 208 B 1 Could not create a user
Kai URL (neteisingas) <pre>{ "UserName": "Teisingas", "Email": "naujas@gmail.com", "Password": "Teisingas!1"}</pre>	 404 Not Found 442 ms 133 B 1

Lentelė 2 (POST) Login

POST	Login
URL (teisingas)	https://leftoversapi.azurewebsites.net/api/register
Užklauso pvz. (JSON):	Atsako pvz. (JSON):
Kai vartotojas egzistuoja:	 200 OK 902 ms 735 B <pre>"accessToken": "eyJhbGciOiJIUzI1NiI¹ eyJodHRwOi8vc2NoZW1hcy54bWxzY2F¹ W1lIjoiYWRTaW4iLCJqdGkiOiIzNjY5¹ J1c2VySWQiOiJiYW2MzQxZi02NGY0L¹ oZW1hcy5taWNYb3NvZnQuY29tL3dzLz¹ dGF1cmFudFVzZXIiLCJlbGZG1pbiJdLCJ¹ iJUCnVzdGVkQ2xpZW50In0.eyJhbGciOi¹</pre>
Kai vartotojas neegzistuoja arba neteisingai įvesti duomenys:	 400 Bad Request 409 ms 217 B <pre>User name or password is invalid</pre>
Kai URL (neteisingas)	https://leftoversapi.azurewebsites.net/api/nerlogin
<pre>{ "UserName": "admin", "Password": "VerySafePassword1!" }</pre>	 404 Not Found 442 ms 133 B <pre>1</pre>



2.1.2. Restoranai

Lentelė 3 (GET) Get All Restaurants



GET	Get All Restaurants
URL (teisingas)	https://leftoversapi.azurewebsites.net/api/chain/2/restaurant
Užklauso pvz. (JSON):	Atsako pvz. (JSON):
Kai URL teisingas:	 200 OK 875 ms 537 B <pre>"id": 3, "name": "Pizza Express1", "description": "G. Vorutos g. 15, Kaunas" "id": 4, "name": "Pizza Express2", "description": "T. Totoro g. 19, Kaunas"</pre>
Kai URL neteisingas:	 404 Not Found 79 ms 133 B

https://leftoversapi.azurewebsites.net/api/chain/2/restaurants	
--	--

Lentelė 4 (GET) Get Restaurant





GET	Get Restaurant
URL (teisingas)	https://leftoversapi.azurewebsites.net/api/chain/1/restaurant/1
Užklauso pvz. (JSON):	Atsako pvz. (JSON):
Kai URL teisingas: https://leftoversapi.azurewebsites.net/api/chain/1/restaurant/1	 200 OK 663 ms 294 B <pre>{ "id": 1, "name": "CanCan1", "description": "G. Britmo g. 63, Kaunas" }</pre>
Kai URL neteisingas: https://leftoversapi.azurewebsites.net/api/chain/1/restaurant/50	 404 Not Found 161 ms 216 B Restaurant with id '50' not found

Lentelė 5 (POST) Create Restaurant



POST	Create Restaurant
URL (teisingas)	https://leftoversapi.azurewebsites.net/api/chain/4/restaurant
Užklauso pvz. (JSON):	Atsako pvz. (JSON):
Kai vartotojas neturi teisių, bet užklausa teisinga: <pre>{ "name": "Grill London3", "description": "T. Krevės g. 14, Kaunas" }</pre>	 401 Unauthorized 425 ms 248 B
Kai vartotojas turi teises, bet užklausa neteisinga: <pre>{ "name": 1, "description": "T. Krevės g. 14, Kaunas" }</pre>	 400 Bad Request 66 ms 620 B <pre>{ "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1", "title": "One or more validation errors occurred.", "status": 400, "traceId": "00-a09fe9d73488c22edeb9036794e04743-56fd2e0a07437811-00", "errors": { "restaurantDto": ["The restaurantDto field is required."] } }</pre>

	<pre> "\$.name": ["The JSON value could not be converted to Leftovers.Data.Dtos.Restaurants.CreateRestaurantDto. Path: \$.name LineNumber: 1 BytePositionInLine: 13."] } } </pre>
Kai vartotojas turi teises ir užklausa teisinga:	 201 Created 458 ms 298 B
<pre> { "name": "Grill London 5", "description": "T. Krevės g. 5, Kaunas" } </pre>	<pre> { "id": 24, "name": "Grill London 5", "description": "T. Krevės g. 5, Kaunas" } </pre>
Kai URL (neteisingas)	https://leftoversapi.azurewebsites.net/api/neregiste
https://leftoversapi.azurewebsites.net/api/chain/4/restaurants	 404 Not Found 442 ms 133 B 1

Lentelė 6 (PUT) Update Restaurant

PUT	Update Restaurant
URL (teisingas)	https://leftoversapi.azurewebsites.net/api/chain/4/restaurant/1
Užklauskos pvz. (JSON):	Atsako pvz. (JSON):
Kai vartotojas neturi teisių, ir užklausa teisinga:	 200 OK 96 ms 304 B
<pre> { "Name" : "Pizza Express1", "Description" : "T. Tumano naujinta g. 122A" } </pre>	<pre> { "id": 1, "name": "Pizza Express1", "description": "T. Tumano naujinta g. 122A" } </pre>
Kai vartotojas turi teises, bet URL neteisingas:	 404 Not Found 85 ms 215 B
https://leftoversapi.azurewebsites.net/api/chain/1/restaurant/6	Restaurant with id '6' not found
Kai vartotojas turi teises ir užklausa teisinga:	 201 Created 458 ms 298 B
<pre> { "name": "Grill London 5", "description": "T. Krevės g. 5, Kaunas" } </pre>	<pre> { "id": 24, "name": "Grill London 5", "description": "T. Krevės g. 5, Kaunas" } </pre>
Kai URL neteisingas:	https://leftoversapi.azurewebsites.net/api/neregiste
https://leftoversapi.azurewebsites.net/api/chain/1/restaurants/6	 404 Not Found 442 ms 133 B

Lentelė 7 (DELETE) Delete Restaurant

DELETE	Delete Restaurant
URL (teisingas)	https://leftoversapi.azurewebsites.net/api/register
Užklauso pvz. (JSON):	Atsako pvz. (JSON):
Kai vartotojas turi teises ir URL teisingas: https://leftoversapi.azurewebsites.net/api/chain/4/restaurant/22	 204 No Content 211 ms 115 B
Kai URL neteisingas: https://leftoversapi.azurewebsites.net/api/chain/1/restaurant/7	https://leftoversapi.azurewebsites.net/api/register  404 Not Found 442 ms 133 B Restaurant with id '7' not found


3. API realizacija

Galiausiai buvo atlikta naudotojo sąsaja pasinaudojant „React.js“. Įgyvendinti šie punktai:

- Sukurti naudotojo sąsajos sprendimą realizuotam REST API
- Responsive layout
- Header, Content (ne Footer) srityse stilius turėtų būti skirtingas (specifiškas). Srityse turėtų būti bent po keletą skirtingų elementų. Analogiški elementai skirtingose srityse turėtų įgauti skirtingą išvaizdą.
- Realizuoti informacijos įvedimo sąsają, bent su keletu skirtingų tipų input elementų.
- Images prisitaikymas
- Panaudoti transitions arba animacijas, pagyvinančius naudotojo sąsają (UI)
- Padarytas responsive meniu
- Panaudoti vektorines ikonas (webfont, svg).
- Panaudotas kitoks šriftas
- Panaudoti modalinį langą, kuriame pateikiama aktuali/prasminga informacija.
- Parinkti tarpusavyje besiderinančias spalvas.
- UI elementai negali išsimėtyti, išdėstymas turi paklusti tinkamo horizontalioms ir vertikaloms linijoms.
- Elementų matomumas ir pasiekiamumas neturi būti apsunkintas.
- Įvedimo formos turi būti aiškios ir nuoseklios.
- Naudotojo sąsajos grafinis dizainas turi būti vientisas, atitikti projekto užduotį
- Projekto kodas turi būti laikomas Git saugykloje (github, bitbucket, gitlab ar kt.). Dokumentacija - projektui sukurtame wiki arba projekto kodo Git saugykloje (.readme).

3.1. Naudotojo sąsaja

Prisijungimas bei registracija:

Restoranų tinklai 

Prisijungti/Atsijungti

Registruotis

Sign In


Username:

Password:

Sign In

Need an Account?
[Sign Up](#)

Pav. 1 Prisijungimas

Restoranų tinklai 

Prisijungti/Atsijungti

Registruotis

Register

Username:

Email:

Password:


Confirm Password:

Sign Up

Already registered?
[Sign In](#)

Pav. 2 Registracija

Restoranų tinklų peržiūra, redagavimas, pridėjimas ir modalinis langas:

Restoranų tinklai

Prisijungti/Atsijungti
Registruotis


Restoranų tinklai

Pavadinimas	Aprašymas	Restoranai	Kesiti	Trinti
CanCan	Picos restoranų tinklas	Peržiūrėti restoranus	Keisti	Trinti
Pizza Express	Picos restoranų tinklas	Peržiūrėti restoranus	Keisti	Trinti
Charlie	Restoranų tinklas	Peržiūrėti restoranus	Keisti	Trinti
Grill London	Restoranų tinklas	Peržiūrėti restoranus	Keisti	Trinti
N333	N	Peržiūrėti restoranus	Keisti	Trinti

Pridėti restoranų tinklą

Pav. 3 Restoranų tinklų sąrašas

Restoranų tinklai



Prisijungti/Atsijungti

Registruotis

Keisti CanCan

Restoranų tinklo pavadinimas

CanCan

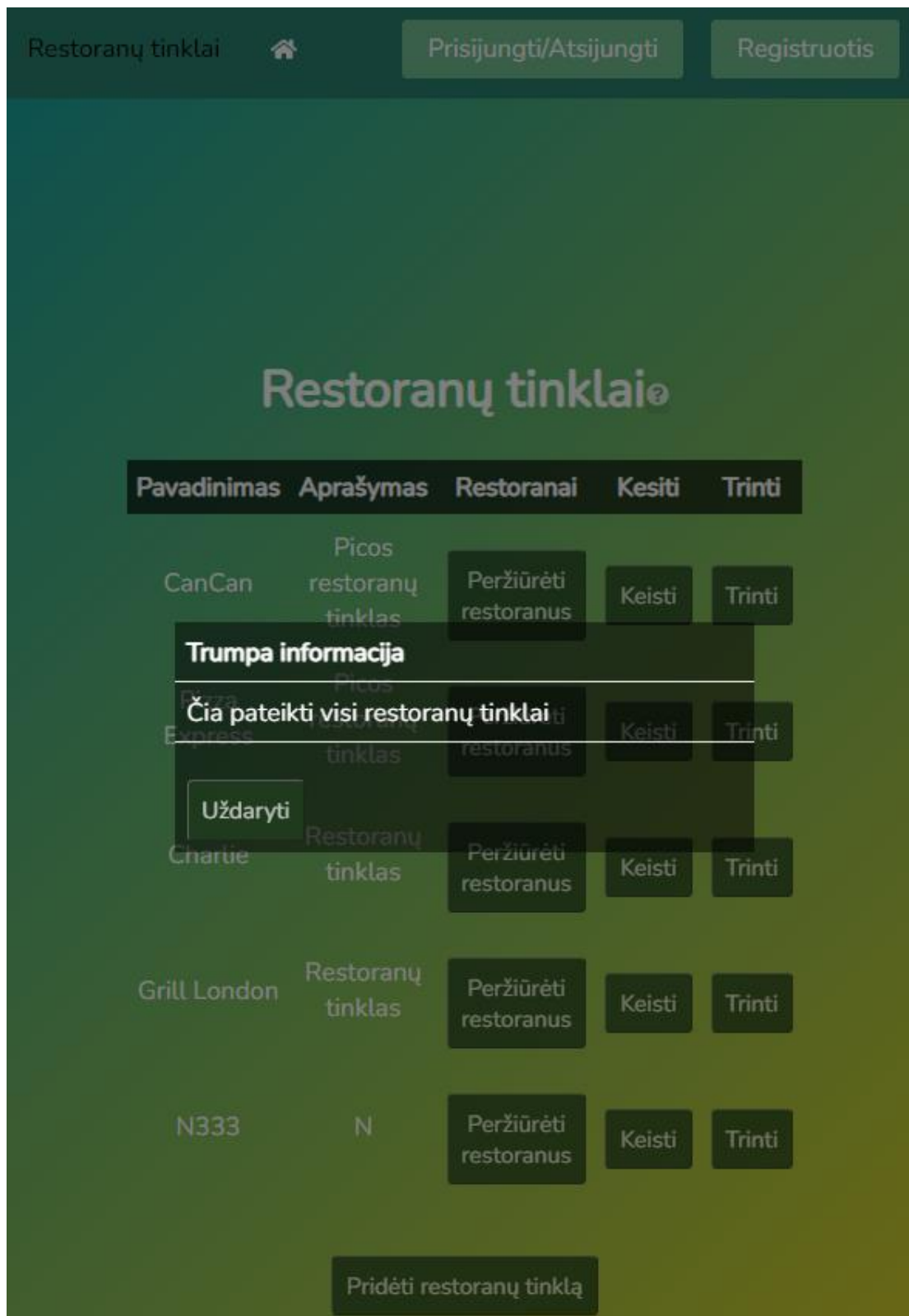
Aprašymas

Picos restoranų tinklas

Keisti

Atgal

Pav. 4 Restorano tinklo redagavimo langas




Pav. 6 Modalinis langas

Restoranų peržiūra, redagavimas ir pridėjimas:



Pav. 7 Pasirinkto tinklo restoranų sąrašas

Restoranų tinklai



Prisijungti/Atsijungti

Registruotis

Redaguoti restoraną

Restorano pavadinimas

Pizza Express1


Aprašymas

G. Britmo g. 63, Kaunas

Pakeisti

Atgal

Pav. 8 Pasirinkto tinklo restorano redagavimo langas

Restoranų tinklai 

Prisijungti/Atsijungti

Registruotis


CanCan Restorano siūlomi patiekalai

ID	Pavadinimas	Kaina	Pirkti	Trinti
1	Havajų pica	8,50 eur	Pirkti	Trinti
2	Firminė pica	7,50 eur	Pirkti	Trinti
3	Salsa pica	7,50 eur	Pirkti	Trinti

Sukurti patiekaląAtgal

Pav. 10 Pasirinkto restorano patiekalų sąrašas

Restoranų tinklai



Prisijungti/Atsijungti

Registruotis

Pridėti patiekalą

Restorano pavadinimas

Pavadinimas

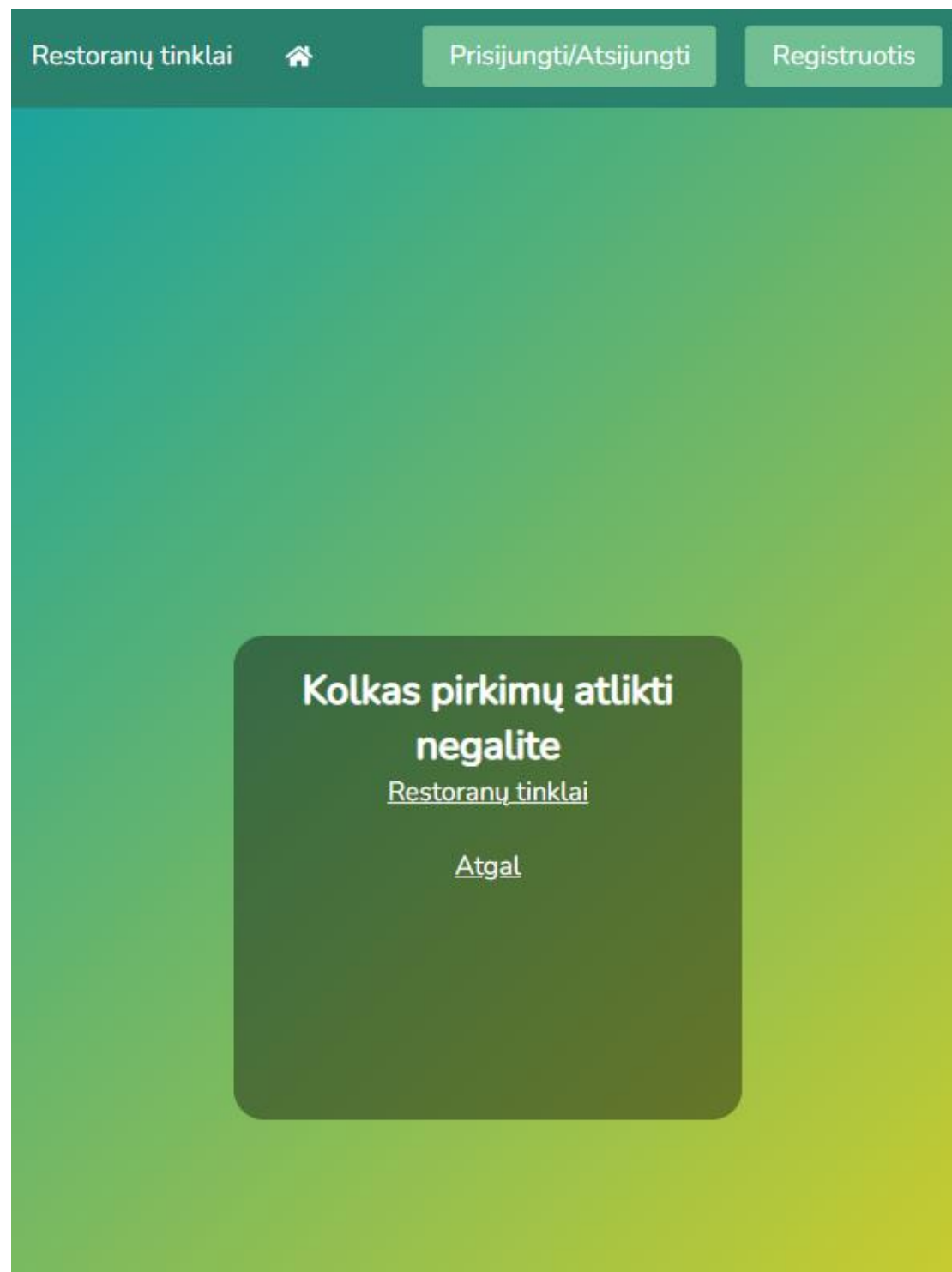
Kaina

Kaina

Pridėti

Atgal

Pav. 11 Pasirinkto restorano patiekalo pridėjimo langas



Pav. 12 Pasirinkto restorano patiekalo pirkimo langas



Pav. 13 "Namų" langas



Pav. 14 Langas, kai yra sumažintas daugiau nei 768 px

4. Išvados

Atliekant projektą buvo įgytos ir pritaikytos žinios apie saityno taikomųjų programų projektavimą. Suprojektuotos ir realizuotos REST principais veikiančios API sąsajos. Pasinaudojant „Postman“ aplikacija ištestuotas užklausų teisingumas. Sėkmingai pritaikyta autorizacija pritaikant JWT technologinius sprendimus. Projektas patalpintas į saityną pasinaudojant „Azure“ debesų technologija. Laikantis nurodymų buvo sukurta naudotojo sąsaja.

Paveikslėlių sąrašas

Pav. 1 Prisijungimas.....	10
Pav. 2 Registracija	11
Pav. 3 Restoranų tinklų sąrašas.....	12
Pav. 4 Restorano tinklo redagavimo langas	13
Pav. 5 Restorano tinklo pridėjimo langas	14
Pav. 6 Modalinis langas	15
Pav. 7 Pasirinkto tinklo restoranų sąrašas	16
Pav. 8 Pasirinkto tinklo restorano redagavimo langas	17
Pav. 9 Pasirinkto restoranų tinklo restorano pridėjimo langas	18
Pav. 10 Pasirinkto restorano patiekalų sąrašas	19
Pav. 11 Pasirinkto restorano patiekalo pridėjimo langas	20
Pav. 12 Pasirinkto restorano patiekalo pirkimo langas	21
Pav. 13 "Namų" langas	22
Pav. 14 Langas, kai yra sumažintas daugiau nei 768 px.....	23

Lentelių sąrašas

Lentelė 1 (Post) Register.....	5
Lentelė 2 (POST) Login.....	6
Lentelė 3 (GET) Get All Restaurants.....	6
Lentelė 4 (GET) Get Restaurant	7
Lentelė 5 (POST) Create Restaurant.....	7
Lentelė 6 (PUT) Update Restaurant.....	8
Lentelė 7 (DELETE) Delete Restaurant	9