



Kauno technologijos universitetas

Informatikos fakultetas

T120B165 Saityno taikomųjų programų projektavimas

Projekto „Viešbučiai“ ataskaita

Daniel Artūr Siebert IFF-9/9

Autorius

doc. prakt. Petras Tamošiūnas
dėst. val. Simonas Baltulionis

Vertintojai

Kaunas, 2022

Turiny

1. Sprendžiamo uždavinio aprašymas	3
1.1. Sistemos paskirtis	3
1.2. Funkciniai reikalavimai	3
2. Sistemos architektūra.....	4

1. Sprendžiamo uždavinio aprašymas

1.1. Sistemos paskirtis

Projekto tikslas – sukurti informacinę sistemą, kuri padėtų valdyti viešbučius.

Veikimo principas – informacinę sistemą sudarys naudotojo sąsaja, kuri bus pasiekama per internetinę aplikaciją, ir kuria naudosis visi sistemos naudotojai bei aplikacijų programavimo sąsaja (angl. trump. *API*).

Sistemos administratorius galės sukurti ir administruoti viešbučius, jų aukštus ir kiekvieno aukšto kambarius, juos redaguoti, peržiūrėti bei šalinti. Sistemos naudotojas galės peržiūrėti, pridėti, šalinti, redaguoti kambarius ir aukštus savo viešbučiams. Neprisijungęs naudotojas galės peržiūrėti viešbučius, prisijungti bei prisiregistruoti. Neregistruotas informacinės sistemos naudotojas galės prisiregistruoti ir peržiūrėti viešbučių sąrašą.

1.2. Funkciniai reikalavimai

Neregistruotas informacinės sistemos naudotojas galės:

1. Peržiūrėti prisijungimo puslapį;
2. Peržiūrėti registracijos puslapį;
3. Prisijungti prie sistemos;
4. Užsiregistruoti į sistemą;
5. Peržiūrėti viešbučių sąrašą;

Registruotas informacinės sistemos naudotojas galės:

1. Prisijungti prie sistemos;
2. Peržiūrėti visus viešbučius
3. Redaguoti savo sukurtus viešbučius, jų aukštus, kambarius.
4. Šalinti savo sukurtus viešbučius, jų aukštus, kambarius.
5. Atsijungti nuo sistemos.

Administratorius galės:

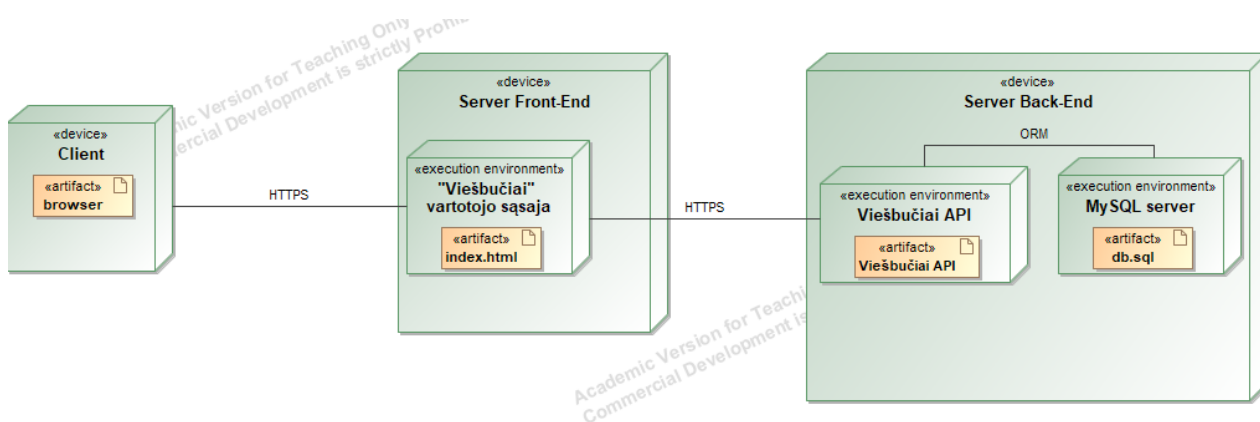
1. Peržiūrėti sistemos naudotojus;
2. Sukurti, ištrinti, redaguoti, peržiūrėti kambarį;
3. Sukurti, ištrinti, redaguoti, peržiūrėti aukštą;
4. Sukurti, ištrinti, redaguoti, peržiūrėti viešbutį.
5. Atsijungti nuo sistemos

2. Sistemos architektūra

Sistemos sudedamosios dalys:

- Kliento pusė (angl. *Front-end*) – bus realizuojama naudojant *React* karkasą. Nuoroda į repozitoriją: <https://github.com/KEKW6969/SaitynuL3>.
- Serverio pusė (angl. *Back-end*) – bus realizuojama naudojant ASP.NET Core. Duomenų bazė MSSQL. Nuoroda į repozitoriją: <https://github.com/KEKW6969/SaitynaiL1>.

1 pav. pavaizduota kuriamos informacinės sistemos diegimo diagrama. Sistemos talpinimui bus naudojami 2 serveriai. Kliento pusės programinis kodas yra talpinamas viename serveryje, o serverio pusės kodas yra talpinamas kitame serveryje. Šie serveriai tarpusavyje komunikuos per *HTTPS* protokolą. Serverio pusės kodas komunikuos su duomenų baze per *ORM* sąsają. Internetinė aplikacija yra pasiekiami per *HTTPS* protokolą.



1 pav. Sistemos "Viešbučiai" diegimo diagrama

3. API Specifikacija

Atsakymo formatas	JSON
Norma ribota?	Neribota

3.1. Viešbučio API metodai

API metodas	Hotels GetMany (GET)
Paskirtis	Gauti visus viešbučius ir jų informaciją
Kelias iki metodo (angl. route)	/api/hotels
Užklausos struktūra	-
Užklausos „header“ dalis	-
Atsakymo struktūra	<pre>[{ "id": 2, "name": "7 Bendrabutisss", "address": "Studentų 71, Kaunas 51393", "phoneNumber": "+370 (696) 73 277", "userId": "826bacbc-7496-4501-8e59-d89d8ae4f0e8" }, { "id": 3, "name": "7 Bendrabutis", "address": "Studentų 71, Kaunas 51393", "phoneNumber": "+370 (696) 73 277", "userId": "826bacbc-7496-4501-8e59-d89d8ae4f0e8" }]</pre>
Atsakymo kodas	200 (OK)
Galimi atsakymo kodai	404 (Not found) – įvedus neteisingą kelią

API metodas	Hotels Get (GET)
Paskirtis	Gauti viešbučio informaciją
Kelias iki metodo (angl. route)	/api/hotels/{hotelId}

Užklauso struktūra	-
Užklauso „header“ dalis	-
Atsakymo struktūra	<pre>{ "id": 2, "name": "7 Bendrabutis", "address": "Studentų 71, Kaunas 51393", "phoneNumber": "+370 (696) 73 277", "userId": "826bacbc-7496-4501-8e59-d89d8ae4f0e8" }</pre>
Atsakymo kodas	200 (OK)
Galimi atsakymo kodai	404 (Not found) – įvedus neteisingą kelią arba jei nėra tokio hotelId

API metodas	Hotels Create (POST)
Paskirtis	Sukurti naują viešbutį
Kelias iki metodo (angl. route)	/api/hotels
Užklauso struktūra	<pre>{ "Name": "9 Bendrabutis", "Address": "Studentų 71, Kaunas 51393", "PhoneNumber": "+370 (696) 73 277" }</pre>
Užklauso „header“ dalis	Authorization: Bearer {token}
Atsakymo struktūra	<pre>{ "id": 20, "name": "9 Bendrabutis", "address": "Studentų 71, Kaunas 51393", "phoneNumber": "+370 (696) 73 277", "userId": "826bacbc-7496-4501-8e59-d89d8ae4f0e8" }</pre>
Atsakymo kodas	201 (Created)
Galimi atsakymo kodai	400 (Bad Request) – neįvestas vienas iš laukų 404 (Not found) – įvedus neteisingą kelią 403 – nenustatytas arba nevalidus token.

API metodas	Hotels Update (PUT)
Paskirtis	Atnaujinti viešbučio duomenis
Kelias iki metodo (angl. route)	/api/hotels/{hotelId}
Užklausos struktūra	<pre>{ "Name": "Magnus Hotel Kaunasss", "Address": "Vytauto Ave 25 Lithuani", "PhoneNumber": "(8-37) 214750" }</pre>
Užklausos „header“ dalis	Authorization: Bearer {token}
Atsakymo struktūra	<pre>{ "id": 20, "name": "Magnus Hotel Kaunasss", "address": "Vytauto Ave 25 Lithuani", "phoneNumber": "(8-37) 214750", "userId": "826bacbc-7496-4501-8e59-d89d8ae4f0e8" }</pre>
Atsakymo kodas	200 (OK)
Galimi atsakymo kodai	<p>400 (Bad Request) – neįvestas vienas iš laukų</p> <p>404 (Not found) – įvedus neteisingą kelią arba jei nėra tokio hotelId</p> <p>403 – nenustatytas arba nevalidus token.</p>

API metodas	Hotels Delete (DELETE)
Paskirtis	Ištrinti viešbutį
Kelias iki metodo (angl. route)	/api/hotels/{hotelId}
Užklausos struktūra	-
Užklausos „header“ dalis	Authorization: Bearer {token}
Atsakymo struktūra	-
Atsakymo kodas	204 (No Content)
Galimi atsakymo kodai	<p>404 (Not found) – įvedus neteisingą kelią arba jei nėra tokio hotelId</p> <p>403 – nenustatytas arba nevalidus token.</p>

3.2. Aukšto API metodai

API metodas	Floors GetMany (GET)
Paskirtis	Gauti visus viešbučio aukštus ir jų informaciją
Kelias iki metodo (angl. route)	/api/hotels/{hotelId}/floors
Užklaustos struktūra	-
Užklaustos „header“ dalis	Authorization: Bearer {token}
Atsakymo struktūra	<pre>[{ "id": 1, "number": 1, "hotel": { "id": 2, "name": "7 Bendrabutisss", "address": "Studentų 71, Kaunas 51393", "phoneNumber": "+370 (696) 73 277", "userId": "826bacbc-7496-4501-8e59-d89d8ae4f0e8", "user": null } }, { "id": 2, "number": 2, "hotel": { "id": 2, "name": "7 Bendrabutisss", "address": "Studentų 71, Kaunas 51393", "phoneNumber": "+370 (696) 73 277", "userId": "826bacbc-7496-4501-8e59-d89d8ae4f0e8", "user": null } }]</pre>
Atsakymo kodas	200 (OK)
Galimi atsakymo kodai	404 (Not found) – įvedus neteisingą kelią ar jei nėra tokio hotelId 403 – nenustatytas arba nevalidus token.

API metodas	Floors Get (GET)
Paskirtis	Gauti viešbučio aukšto informaciją
Kelias iki metodo (angl. route)	/api/hotels/{hotelId}/floors/{floorId}
Užklausos struktūra	-
Užklausos „header“ dalis	Authorization: Bearer {token}
Atsakymo struktūra	<pre>{ "id": 5, "number": 5, "hotel": { "id": 2, "name": "7 Bendrabutisss", "address": "Studentų 71, Kaunas 51393", "phoneNumber": "+370 (696) 73 277", "userId": "826bacbc-7496-4501-8e59-d89d8ae4f0e8", "user": null } }</pre>
Atsakymo kodas	200 (OK)
Galimi atsakymo kodai	<p>404 (Not found) – įvedus neteisingą kelią ar jei nėra tokio hotelId ar floorId</p> <p>403 – nenustatytas arba nevalidus token.</p>

API metodas	Floors Create (POST)
Paskirtis	Sukurti naują aukštą
Kelias iki metodo (angl. route)	/api/hotels/{hotelId}/floors
Užklausos struktūra	<pre>{ "Number": "1" }</pre>
Užklausos „header“ dalis	Authorization: Bearer {token}
Atsakymo struktūra	<pre>{ "id": 19, "number": 1, "hotel": { "id": 2, "name": "7 Bendrabutisss", "address": "Studentų 71, Kaunas 51393", </pre>

	<pre> "phoneNumber": "+370 (696) 73 277", "userId": "826bacbc-7496-4501-8e59-d89d8ae4f0e8", "user": null } } </pre>
Atsakymo kodas	201 (Created)
Galimi atsakymo kodai	<p>400 (Bad Request) – neįvestas laukas</p> <p>404 (Not found) – įvedus neteisingą kelią ar jei nėra tokio hotelId</p> <p>403 – nenustatytas arba nevalidus token.</p>

API metodas	Floors Update (PUT)
Paskirtis	Atnaujinti aukšto duomenis
Kelias iki metodo (angl. route)	/api/hotels/{hotelId}/floors/{floorId}
Užklausos struktūra	<pre> { "Number": "222" } </pre>
Užklausos „header“ dalis	Authorization: Bearer {token}
Atsakymo struktūra	<pre> { "id": 19, "number": 222, "hotel": { "id": 2, "name": "7 Bendrabutisss", "address": "Studentų 71, Kaunas 51393", "phoneNumber": "+370 (696) 73 277", "userId": "826bacbc-7496-4501-8e59-d89d8ae4f0e8", "user": null } } </pre>
Atsakymo kodas	200 (OK)
Galimi atsakymo kodai	<p>400 (Bad Request) – neįvestas laukas</p> <p>404 (Not found) – įvedus neteisingą kelią arba jei nėra tokio hotelId (parodoma žinutė „Hotel with id ‘...‘ not found.“)</p> <p>403 – nenustatytas arba nevalidus token.</p>

API metodas	Floors Delete (DELETE)
Paskirtis	Ištrinti aukštą
Kelias iki metodo (angl. route)	/api/hotels/{hotelId}/floors/{floorId}
Užklausos struktūra	-
Užklausos „header“ dalis	Authorization: Bearer {token}
Atsakymo struktūra	-
Atsakymo kodas	204 (No Content)
Galimi atsakymo kodai	404 (Not found) – įvedus neteisingą kelią arba jei nėra tokio hotelId ar floorId 403 – nenustatytas arba nevalidus token.

3.3. Kambario API metodai

API metodas	Rooms GetMany (GET)
Paskirtis	Gauti visus viešbučio aukšto kambarius ir jų informaciją
Kelias iki metodo (angl. route)	/api/hotels/{hotelId}/floors/{floorId}/rooms
Užklausos struktūra	-
Užklausos „header“ dalis	Authorization: Bearer {token}
Atsakymo struktūra	<pre>[{ "id": 7, "number": 519, "price": 169.69, "description": "Kambarys su numeriu 519", "floor": { "id": 19, "number": 222, "hotel": { "id": 2, "name": "7 Bendrabutisss", "address": "Studentų 71, Kaunas 51393", "phoneNumber": "+370 (696) 73 277",</pre>

	<pre> "userId": "826bacbc-7496-4501-8e59-d89d8ae4f0e8", "user": null } }, { "id": 8, "number": 519, "price": 169.69, "description": "Kambarys su numeriu 519", "floor": { "id": 19, "number": 222, "hotel": { "id": 2, "name": "7 Bendrabutisss", "address": "Studentų 71, Kaunas 51393", "phoneNumber": "+370 (696) 73 277", "userId": "826bacbc-7496-4501-8e59-d89d8ae4f0e8", "user": null } } }] </pre>
Atsakymo kodas	200 (OK)
Galimi atsakymo kodai	<p>404 (Not found) – įvedus neteisingą kelią ar jei nėra tokio hotelId ar floorId</p> <p>403 – nenustatytas arba nevalidus token.</p>

API metodas	Rooms Get (GET)
Paskirtis	Gauti viešbučio aukšto kambario informaciją
Kelias iki metodo (angl. route)	/api/hotels/{hotelId}/floors/{floorId}/rooms/{roomId}
Užklauskos struktūra	-
Užklauskos „header“ dalis	Authorization: Bearer {token}
Atsakymo struktūra	<pre> { "id": 7, "number": 519, "price": 169.69, "description": "Kambarys su numeriu 519", "floor": { </pre>

	<pre> "id": 19, "number": 222, "hotel": { "id": 2, "name": "7 Bendrabutisss", "address": "Studentų 71, Kaunas 51393", "phoneNumber": "+370 (696) 73 277", "userId": "826bacbc-7496-4501-8e59-d89d8ae4f0e8", "user": null } } } </pre>
Atsakymo kodas	200 (OK)
Galimi atsakymo kodai	<p>404 (Not found) – įvedus neteisingą kelią ar jei nėra tokio hotelId ar floorId</p> <p>403 – nenustatytas arba nevalidus token.</p>

API metodas	Rooms Create (POST)
Paskirtis	Sukurti naują kambarį
Kelias iki metodo (angl. route)	/api/hotels/{hotelId}/floors/{floorId}/rooms
Užklausos struktūra	<pre> { "Number": 519, "Price": 169.69, "Description": "Kambarys su numeriu 519" } </pre>
Užklausos „header“ dalis	Authorization: Bearer {token}
Atsakymo struktūra	<pre> { "id": 8, "number": 519, "price": 169.69, "description": "Kambarys su numeriu 519", "floor": { "id": 19, "number": 222, "hotel": { "id": 2, "name": "7 Bendrabutisss", "address": "Studentų 71, Kaunas 51393", "phoneNumber": "+370 (696) 73 277", "userId": "826bacbc-7496-4501-8e59-d89d8ae4f0e8", </pre>

	<pre> "user": null } } } </pre>
Atsakymo kodas	201 (Created)
Galimi atsakymo kodai	<p>400 (Bad Request) – neįvestas vienas iš laukų</p> <p>404 (Not found) – įvedus neteisingą kelią ar jei nėra tokio hotelId ar floorId</p> <p>403 – nenustatytas arba nevalidus token.</p>

API metodas	Rooms Update (PUT)
Paskirtis	Atnaujinti kambario duomenis
Kelias iki metodo (angl. route)	/api/hotels/{hotelId}/floors/{floorId}/rooms/{roomId}
Užklausos struktūra	<pre> { "Number": 5, "Price": 169.69, "Description": "Kambarys su numeriu 5" } </pre>
Užklausos „header“ dalis	Authorization: Bearer {token}
Atsakymo struktūra	<pre> { "id": 7, "number": 5, "price": 169.69, "description": "Kambarys su numeriu 5", "floor": { "id": 19, "number": 222, "hotel": { "id": 2, "name": "7 Bendrabutisss", "address": "Studentų 71, Kaunas 51393", "phoneNumber": "+370 (696) 73 277", "userId": "826bacbc-7496-4501-8e59-d89d8ae4f0e8", "user": null } } } </pre>

Atsakymo kodas	200 (OK)
Galimi atsakymo kodai	<p>400 (Bad Request) – neįvestas vienas iš laukų</p> <p>404 (Not found) – įvedus neteisingą kelią arba jei nėra tokio hotelId ar floorId</p> <p>403 – nenustatytas arba nevalidus token.</p>

API metodas	Rooms Delete (DELETE)
Paskirtis	Ištrinti kambarį
Kelias iki metodo (angl. route)	/api/hotels/{hotelId}/floors/{floorId}/rooms/{roomsId}
Užklausos struktūra	-
Užklausos „header“ dalis	Authorization: Bearer {token}
Atsakymo struktūra	-
Atsakymo kodas	204 (No Content)
Galimi atsakymo kodai	<p>404 (Not found) – įvedus neteisingą kelią arba jei nėra tokio hotelId ar floorId</p> <p>403 – nenustatytas arba nevalidus token.</p>

4. Sistemos naudotojo sąsaja

Pagrindinis sistemos puslapis (2 pav.), kuriame yra mygtukai sugrįžti į viešbučių sąrašą ir atsijungimui nuo sistemos. Puslapyje taip pat yra pateikiamas pilnas viešbučių sąrašas, viešbučio sukūrimui skirtas mygtukas. Galima paspausti prie viešbučių esančius redagavimo (EDIT) ir ištrynimo (DELETE) mygtukus. Yra galimybė peržiūrėti viešbučių aukštus.

Viešbučiai

Hotels

Logout

CREATE HOTEL

Hotel ID	Name	Address	Phone number	Edit	Delete	Floors
2	7 Bendrabutiss	Studentų 71, Kaunas 51393	+370 (696) 73 277	EDIT	DELETE	FLOORS
3	7 Bendrabutis	Studentų 71, Kaunas 51393	+370 (696) 73 277	EDIT	DELETE	FLOORS
5	9 Bendrabutis	Studentų 71, Kaunas 51393	+370 (696) 73 277	EDIT	DELETE	FLOORS
6	9 Bendrabutis	Studentų 71, Kaunas 51393	+370 (696) 73 277	EDIT	DELETE	FLOORS
7	9 Bendrabutis	Studentų 71, Kaunas 51393	+370 (696) 73 277	EDIT	DELETE	FLOORS
9	9 Bendrabutis	Studentų 71, Kaunas 51393	+370 (696) 73 277	EDIT	DELETE	FLOORS
10	Magnus Hotel Kaunase	Vytauto Ave 25 Lithuania	(8-37) 214750	EDIT	DELETE	FLOORS
11	9 Bendrabutis	Studentų 71, Kaunas 51393	+370 (696) 73 277	EDIT	DELETE	FLOORS

Rows per page: 100

1-8 of 8

Copyright ©2022.

2 pav. Pagrindinis sistemos puslapis

Aukštų sąrašo langas (3 pav.), jame galime sukurti, redaguoti ir ištrinti pasirinkto viešbučio aukštus. Galime iš šio lango pereiti į pasirinkto aukšto kambarius.

Viešbučiai

Hotels

Logout

CREATE FLOOR

Floor ID	Floor's Number	Edit	Delete	Rooms
1	1	EDIT	DELETE	ROOMS
2	2	EDIT	DELETE	ROOMS
3	3	EDIT	DELETE	ROOMS
4	4	EDIT	DELETE	ROOMS
5	5	EDIT	DELETE	ROOMS
19	222	EDIT	DELETE	ROOMS

Rows per page: 100

1-6 of 6

Copyright ©2022.

3 pav. Aukštų sąrašo langas

Viešbučio aukšto kambarių sąrašo puslapis (4 pav.), kuriame galime sukurti, redaguoti ir ištrinti kambarį.

Viešbučiai

Hotels

Logout

CREATE ROOM

Room ID	Room Number	Room's price	Room's description	Edit	Delete
7	5	169.69	Kambarys su nume...	EDIT	DELETE
8	519	169.69	Kambarys su nume...	EDIT	DELETE

1 row selected

Rows per page: 100 ▾1-2 of 2<>

Copyright ©2022.

4 pav. Viešbučio aukšto kambarių sąrašo puslapis

Viešbučio sukūrimo lange (5 pav.) galime įvesti viešbučio informaciją.

Create a hotel

Name

Address

Phone number

CREATE

5 pav. Viešbučio sukūrimo langas

Viešbučio redagavimo lange (6 pav.) matome esamą viešbučio informaciją ir galime ją redaguoti.

Edit hotel

Name

9 Bendrabutis

Address

Studentų 71, Kaunas 51393

Phone number

+370 (696) 73 277

[EDIT](#)

6 pav. Viešbučio redagavimo langas

Viešbučio ištrynimo patvirtinimo lange (7 pav.) matome viešbučio, kurį norime ištrinti, informaciją.

Delete hotel

Are you sure you want to delete?

Name

9 Bendrabutis

Address

Studentų 71, Kaunas 51393

Phone number

+370 (696) 73 277

[DELETE](#)

7 pav. Viešbučio ištrynimo patvirtinimo langas

Aukšto sukūrimo lange (8 pav.) galime pasirinkti arba įvesti aukšto numerį

Create a floor

Floor's number

CREATE

8 pav. Aukšto sukūrimo langas

Aukšto redagavimo lange (8 pav.) matome esamą aukšto informaciją, galime pasirinkti arba įvesti aukšto numerį.

Edit floor

Floor's number

5

EDIT

9 pav. Aukšto redagavimo langas

Aukšto ištrynimo patvirtinimo lange (10 pav.) matome esamą aukšto informaciją.

Delete floor

Floor's number

5

DELETE

10 pav. Aukšto ištrynimo patvirtinimo langas

Kambario sukūrimo lange (11 pav.) galime įvesti ar pasirinkti kambario numerį ir kainą, įrašyti aprašymą.

Create a room

Number *

Price *

Description *

CREATE

11 pav. Kambario sukūrimo langas

Kambario redagavimo lange (12 pav.) galime pamatyti kambario informaciją, pasirinkti ar įvesti kambario numerį ir kainą, įrašyti naują aprašymą.

Edit room

Rooms's number

5

Room's price

169.69

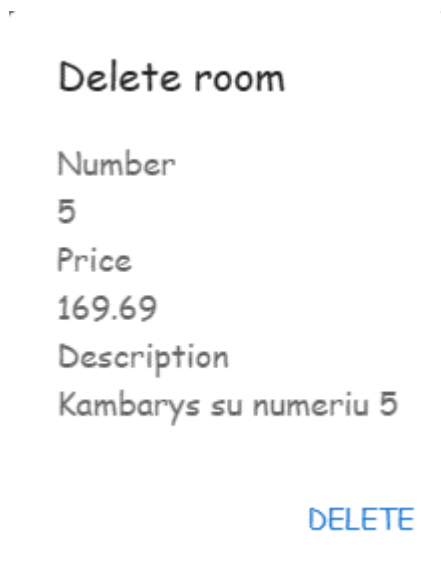
Room's description

Kambarys su numeriu 5

EDIT

12 pav. Kambario redagavimo langas

Kambario ištrynimo patvirtinimo lange (13 pav.) matome kambario, kurį ketiname ištrinti, informaciją.



Delete room

Number
5

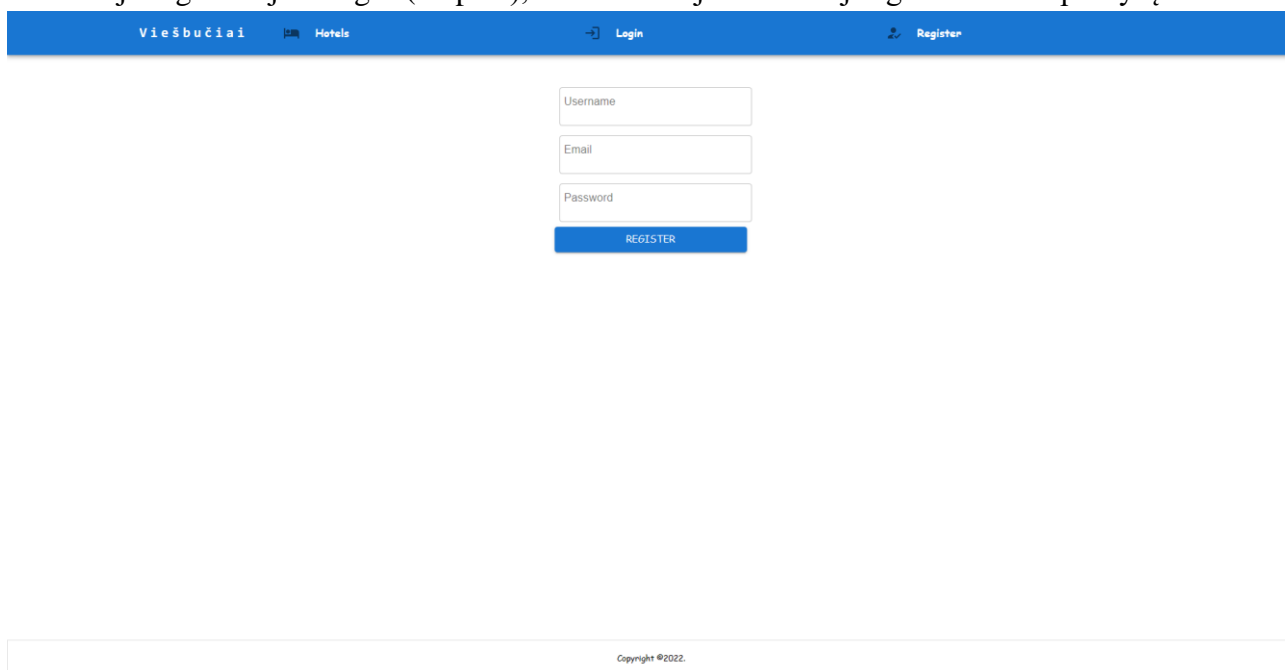
Price
169.69

Description
Kambarys su numeriu 5

DELETE

13 pav. Kambario ištrynimo patvirtinimo langas

Naudotojo registracijos langas (14 pav.), kuriame naujas naudotojas gali susikurti paskyrą.



Viešbučiai Hotels Login Register

Username

Email

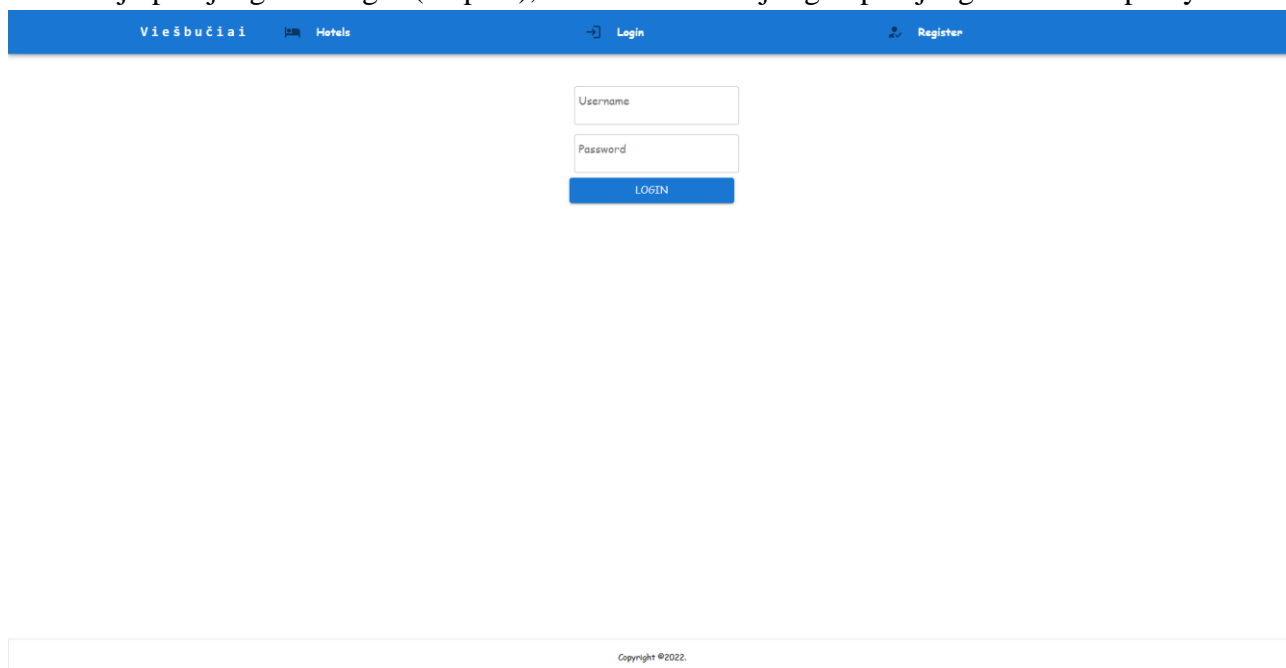
Password

REGISTER

Copyright ©2022

14 pav. Registracijos langas

Naudotojo prisijungimo langas (15 pav.), kuriame naudotojas gali prisijungti su esama paskyra.



The image shows a user login page. At the top, there is a blue navigation bar with the text "Viešbučiai" and "Hotels" on the left, and "Login" and "Register" on the right. Below the navigation bar, there are two input fields: "Username" and "Password". Below the "Password" field is a blue button labeled "LOGIN". At the bottom of the page, there is a thin horizontal line with the text "Copyright ©2022." centered below it.

15 pav. Prisijungimo langas

5. Išvados

- Realizuota sistema, kuri palengvina darbą stebėti viešbučius, aukštus ir kambarius didelėms viešbučio sistemoms.
- Sistemos talpinimui naudojami Azure ir Netlify serveriai
- Sistemą sudaro dvi dalys, serverio ir kliento pusės.
- Kliento pusėje naudojama React biblioteka, o serverio – ASP.NET Core.
- Realizuota REST principais paremta sistema
- Realizuoti 15 sąsajos metodai.
- Realizuota JWT autentifikacija ir autorizacija.
- Sukurta naudotojo sąsaja, leidžiant patogiai naudotis visu REST API.