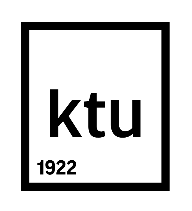
****

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**IFORMATIKOS FAKULTETAS**

**T120B165 Saityno taikomųjų programų projektavimas**

**Internetinė mažmeninė prekyba**

**Atliko:** IFK-0 gr. stud. Augustas Druceika

**Dėstytojai:** Tomas Blažauskas

Tomas Lukošius

**KAUNAS, 2022**

Turinys

[1. Sprendžiamo uždavinio aprašymas 3](#_Toc114684046)

[1. Sistemos paskirtis 3](#_Toc114684047)

[2. Funkciniai reikalavimai 3](#_Toc114684048)

[Neregistruotas vartotojas gales: 3](#_Toc114684049)

[2. Sistemos architektūra 3](#_Toc114684050)

[1. Sistemos sudedamos dalys: 3](#_Toc114684051)

[2. Objektų ryšis 3](#_Toc114684053)

# Sprendžiamo uždavinio aprašymas

## Sistemos paskirtis

Projekto tikslas – sukurti svetainę kurioje gamintojai ar mažmeninę prekyba užsiimančios įmonės ar asmenys galėtų pardavinėti savo produkcija kitoms įmonėms ar juridiniams asmenims.

Veikimo principas – pačia kuriamą platformą sudaro dvi dalys internetine aplikacija kurią galės naudoti pardavėjai, pirkėjai ir administracija bei aplikacijos programavimo sąsaja (angl. API)

Juridiniai asmenys kurie norės nusipirkti prekių turės prisiregistruoti ir suvesti savo duomenis kai prisiregistruos jis matys prekių sąrašą kuriame gali būti nuo žaliavų iki pagamintų produktų. Gamintojai ar mažmenine prekyba užsiimančios imonės kurios norės pardavinėti savo produktus turės susisiekti su administratoriais kurie suteiks jiems prisijungimą, o kai jie gaus savo prisijungimus galės skelbti savo parduodamų produktų sąrašą.

Prekes bus galima atsirinkti pagal kategorijas ir subkategorijas.

## Funkciniai reikalavimai

## Neregistruotas vartotojas gales:

* 1. Prisiregistruoti ar prisijungti.  
     **Registruotas pirkėjas galės:**
  2. Atsijungti nuo internetines aplikacijos
  3. Prisijungti prie aplikacijos
  4. Įdėti prekes į pirkinių krepšelį
  5. Koreguoti pirkinių krepšelį
  6. Pasirinkti norima kieki prekių
  7. Pašalinti prekes iš pirkinių krepšelio
  8. Nusipirkti prekes bankiniu pervedimu

**Gamintojai ar mažmeninę prekyba užsiimančios įmonės gales:**

1. Atsijungti nuo internetines aplikacijos
2. Prisijungti prie aplikacijos
3. Įdėti prekes į pardavimų sąrašą
4. Redaguoti parduodamas prekes
5. Pašalinti prekes iš pardavimo sąrašo
6. Matyti užsakymus
7. Patvirtinti užsakymus  
   **Administratorius gales:**
   1. Užregistruoti gamintojus ar įmones kurios nori atlikti prekyba

## Sistemos architektūra

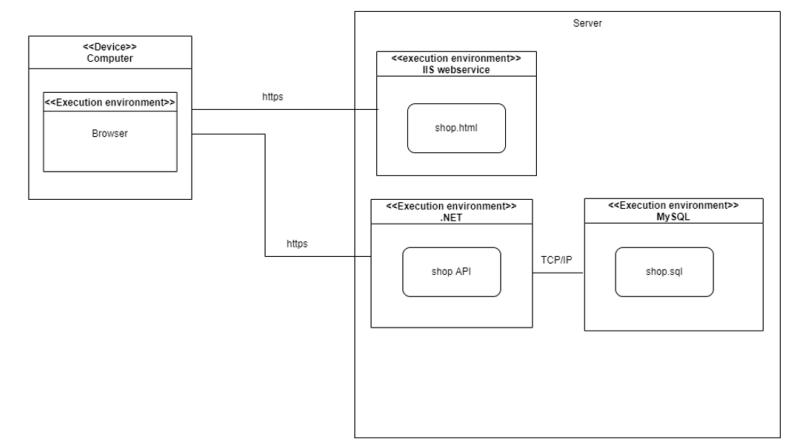
### Sistemos sudedamos dalys:

* Kliento puse (angl. Fron-End) – React.js
* Serverio puse (angl. Back-End) – naudojant c#. Duomenų baze MySQL

### Objektų ryšis

Pardavėjas (gamintojas ar mažmeninę prekyba užsiimančios įmonės) gali turėti daug prekių, tos prekės įeina į užsakovų sąskaitas kuriose gali būti daug prekių ir kiekviena sąskaita priklauso tik vienam užsakovui.

### Diegimo diagrama



## API specifikacija

Pateikiama svarbiausia informacija sėkmingui užklausos įvykdymui. Kelias į metodą bus parašyta tik pabaiga, nes domenas gali skirtis.

1. Parduotuvių API metodai

|  |  |
| --- | --- |
| API metodas | Get |
| Paskirtis | Gauti vieną parduotuvę pagal jos id |
| Kelias iki metodo | /api/sellers/{id} |
| Užklausos struktūra | - |
| Header dalis | - |
| Atsakymo struktūra | {          "id": 15,          "name": "vilniaus kava",          "city": "vilnius442224",          "address": "kauno g",          "userId": "899e4325-4967-493d-ad93-caf5027313b4"      }, |
| Atsakymas | 200 (kai yra pardavėjas), 404(kai nerandamas) |
| Galimos klaidos | 400(kai API kelias yra blogas) |

Lentelė 1

|  |  |
| --- | --- |
| API metodas | Get many |
| Paskirtis | Gauti visas parduotuves |
| Kelias iki metodo | /api/sellers |
| Užklausos struktūra | - |
| Header dalis | - |
| Atsakymo struktūra | {          "id": 15,          "name": "vilniaus kava",          "city": "vilnius442224",          "address": "kauno g",          "userId": "899e4325-4967-493d-ad93-caf5027313b4"      }, |
| Atsakymas | 200 |
| Galimos klaidos | - |

Lentelė 2

|  |  |
| --- | --- |
| API metodas | Create |
| Paskirtis | Sukurti nauja pardavėja |
| Kelias iki metodo | /api/sellers |
| Užklausos struktūra | {       "City": "vilnius442224",       "Name": "vilniaus kava",       "address": "kauno g"    } |
| Header dalis | Authorization: Bearer(token) |
| Atsakymo struktūra | {      "id": 33,      "name": "vilniaus kava",      "city": "vilnius442224",      "address": "kauno g",      "userId": "6bff1618-9c8e-4445-85c3-b9ca345d4795"  } |
| Atsakymas | 201(kai sukuriama) |
| Galimos klaidos | 401(kai vartotojas nėra prisijungęs), 403(kai vartotojas nėra pardavėjas) |

Lentelė 3

|  |  |
| --- | --- |
| API metodas | Update |
| Paskirtis | Atnaujinti esamą |
| Kelias iki metodo | /api/sellers/{id} |
| Užklausos struktūra | {       "City": "vilnius442224",       "Name": "vilniaus kava",       "address": "kauno g"    } |
| Header dalis | Authorization: Bearer(token) |
| Atsakymo struktūra | {      "id": 33,      "name": "vilniaus kava",      "city": "vilnius442224",      "address": "kauno g",      "userId": "6bff1618-9c8e-4445-85c3-b9ca345d4795"  } |
| Atsakymas | 200(kai atnaujinama) |
| Galimos klaidos | 401(kai vartotojas nėra prisijungęs), 403(kai vartotojas nėra pardavėjas) |

Lentelė 4

|  |  |
| --- | --- |
| API metodas | Delete |
| Paskirtis | Ištrinti pardavėją pagal id |
| Kelias iki metodo | /api/sellers/{id} |
| Užklausos struktūra | - |
| Header dalis | Authorization: Bearer(token) |
| Atsakymo struktūra | - |
| Atsakymas | 204(kai sėkmingai ištrinimą) |
| Galimos klaidos | 401(kai vartotojas nėra prisijungęs), 403(kai vartotojas nėra pardavėjas) |

Lentelė 5

1. Prekių API metodai

|  |  |
| --- | --- |
| API metodas | Get |
| Paskirtis | Gauti vienos prekės informacija |
| Kelias iki metodo | api/sellers/{sellerId}/items/{itemsId} |
| Užklausos struktūra | - |
| Header dalis | - |
| Atsakymo struktūra | {          "id": 23,          "name": "asaa",          "description": "gelezies blokai 10cmx10cmx10cm",          "price": 10,          "count": 5,          "sellersId": 20      } |
| Atsakymas | 200(kai egzistuoja prekė ar pardavėjas), 404(kai nerandama pardavėjo ar prekės) |
| Galimos klaidos | 400(kai API kelias yra blogas) |

Lentelė 6

|  |  |
| --- | --- |
| API metodas | Get many |
| Paskirtis | Gauti visas prekes iš vieno pardavėjo |
| Kelias iki metodo | api/sellers/{sellerId}/items |
| Užklausos struktūra | - |
| Header dalis | - |
| Atsakymo struktūra | {          "id": 23,          "name": "asaa",          "description": "gelezies blokai 10cmx10cmx10cm",          "price": 10,          "count": 5,          "sellersId": 20      } |
| Atsakymas | 200(pardavėjas), 404(kai nerandama pardavėjo) |
| Galimos klaidos | 400(kai API kelias yra blogas) |

Lentelė 7

|  |  |
| --- | --- |
| API metodas | Create |
| Paskirtis | Sukurti prekę |
| Kelias iki metodo | api/sellers/{sellerId}/items |
| Užklausos struktūra | {      "name": "asaa",      "Description": "gelezies blokai 10cmx10cmx10cm",      "price": "10",      "count": 5  } |
| Header dalis | Authorization: Bearer(token) |
| Atsakymo struktūra | {      "id": 35,      "name": "asaa",      "description": "gelezies blokai 10cmx10cmx10cm",      "price": 10,      "count": 5,      "sellersId": 36  } |
| Atsakymas | 201(kai sukuriama) |
| Galimos klaidos | 401(kai vartotojas nėra prisijungęs), 403(kai vartotojas nėra pardavėjas, arba tos parduotuvės pardavėjas) |

Lentelė 8

|  |  |
| --- | --- |
| API metodas | Update |
| Paskirtis | Atnaujinti prekės informaciją |
| Kelias iki metodo | api/sellers/{sellerId}/items |
| Užklausos struktūra | {      "name": "asaa",      "escription": "gelezies blokai 10cmx10cmx10cm",      "price": "10",      "count": 5  } |
| Header dalis | Authorization: Bearer(token) |
| Atsakymo struktūra | {      "id": 36,      "name": "asaa",      "description": "gelezies blokai 10cmx10cmx10cm",      "price": 10,      "count": 5,      "sellersId": 36  } |
| Atsakymas | 200(kai atnaujinama) |
| Galimos klaidos | 401(kai vartotojas nėra prisijungęs), 403(kai vartotojas nėra pardavėjas) |

Lentelė 9

|  |  |
| --- | --- |
| API metodas | Delete |
| Paskirtis | Ištrinti prekę |
| Kelias iki metodo | api/sellers/{sellerId}/items/{itemsId} |
| Užklausos struktūra | - |
| Header dalis | Authorization: Bearer(token) |
| Atsakymo struktūra | - |
| Atsakymas | 204(kai sėkmingai ištrinimą) |
| Galimos klaidos | 401(kai vartotojas nėra prisijungęs), 403(kai vartotojas nėra pardavėjas ir tai nėra jo prekė) |

Lentelė 10

1. Sąskaitos API metodai

|  |  |
| --- | --- |
| API metodas | Get |
| Paskirtis | Gauti vieną sąskaitą |
| Kelias iki metodo | api/sellers/{sellerId}/items{itemId}/bills/{billsId} |
| Užklausos struktūra | - |
| Header dalis | Authorization: Bearer(token) |
| Atsakymo struktūra | {          "id": 82,          "buyerName": "dasdasd",          "buyerSecondName": "asdasd",          "city": "asdasd",          "address": "asdasdas",          "dateTime": "2022-12-16T00:00:00"      } |
| Atsakymas | 200(kai egzistuoja prekė ar pardavėjas), 404(kai nerandama pardavėjo ar prekės) |
| Galimos klaidos | 400(kai API kelias yra blogas), 401(kai vartotojas nėra prisijungęs) |

Lentelė 11

|  |  |
| --- | --- |
| API metodas | Get many |
| Paskirtis | Gauti visas saskaitas |
| Kelias iki metodo | api/sellers/{sellerId}/items{itemId}/bills |
| Užklausos struktūra | - |
| Header dalis | Authorization: Bearer(token) |
| Atsakymo struktūra | {          "id": 82,          "buyerName": "dasdasd",          "buyerSecondName": "asdasd",          "city": "asdasd",          "address": "asdasdas",          "dateTime": "2022-12-16T00:00:00"      } |
| Atsakymas | 200(kai egzistuoja prekė ar pardavėjas), 404(kai nerandama pardavėjo ar prekės) |
| Galimos klaidos | 400(kai API kelias yra blogas), 401(kai vartotojas nėra prisijungęs) |

Lentelė 12

|  |  |
| --- | --- |
| API metodas | Create |
| Paskirtis | Sukurti naują sąskaitą |
| Kelias iki metodo | api/sellers/{sellerId}/items{itemId}/bills |
| Užklausos struktūra | {      "city": "miestas",      "address": "adresas",      "BuyerName": "vardas",      "BuyerSecondName" : "pavarde"  } |
| Header dalis | Authorization: Bearer(token) |
| Atsakymo struktūra | {      "id": 83,      "buyerName": "vardas",      "buyerSecondName": "pavarde",      "city": "miestas",      "address": "adresas",      "dateTime": "2022-12-22T23:52:09.2010633+02:00"  } |
| Atsakymas | 201(kai sukuriama) |
| Galimos klaidos | 401(kai vartotojas nėra prisijungęs), 403(kai vartotojas nėra pardavėjas, arba tos parduotuvės pardavėjas) |

Lentelė 13

|  |  |
| --- | --- |
| API metodas | Update |
| Paskirtis | Atnaujinti sąskaitą |
| Kelias iki metodo | api/sellers/{sellerId}/items{itemId}/bills/{billsId} |
| Užklausos struktūra | {      "city": "miestas",      "address": "adresas",      "BuyerName": "vardas",      "BuyerSecondName" : "pavarde"  } |
| Header dalis | Authorization: Bearer(token) |
| Atsakymo struktūra | {      "id": 83,      "buyerName": "vardas",      "buyerSecondName": "pavarde",      "city": "miestas",      "address": "adresas",      "dateTime": "2022-12-22T23:52:09.2010633+02:00"  } |
| Atsakymas | 200(kai atnaujinama) |
| Galimos klaidos | 401(kai vartotojas nėra prisijungęs), 403(kai vartotojas nėra pardavėjas) |

Lentelė 14

|  |  |
| --- | --- |
| API metodas | Delete |
| Paskirtis | Ištrinti sąskaitą |
| Kelias iki metodo | api/sellers/{sellerId}/items{itemId}/bills/{billsId} |
| Užklausos struktūra | - |
| Header dalis | Authorization: Bearer(token) |
| Atsakymo struktūra | - |
| Atsakymas | 204(kai sėkmingai ištrinimą) |
| Galimos klaidos | 401(kai vartotojas nėra prisijungęs), 403(kai vartotojas nėra pardavėjas ir tai nėra jo prekė) |

Lentelė 15

## Vartotojo sąsaja

Vartotojo sąsaja yra sukurta tam, kad klientai galėtų lengviau susirasti prekes ir lengviau suprasti kaip veikia ši Sistema.

#### Pradinis puslapis





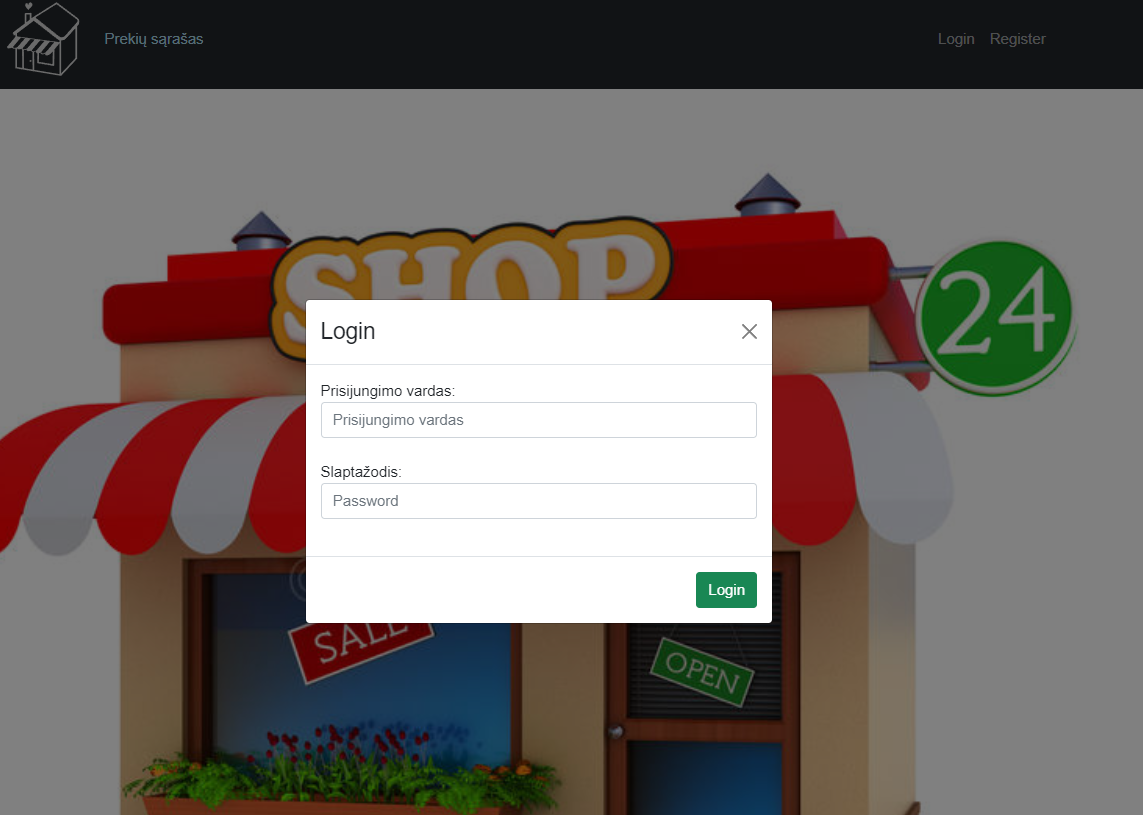
#### Prisijungimas ir registracija

Registracijos puslapį reikia ivęsti vartotojo vardą, elektroninį paštą ir slaptažodį.

Paveikslėlis, kuriame yra žinutė, siena, vidinis, ekrano nuotrauka

Automatiškai sugeneruotas aprašymas

Prisijungimo puslapį reikia įvesti savo prisijungimo vardą ir slaptažodį.



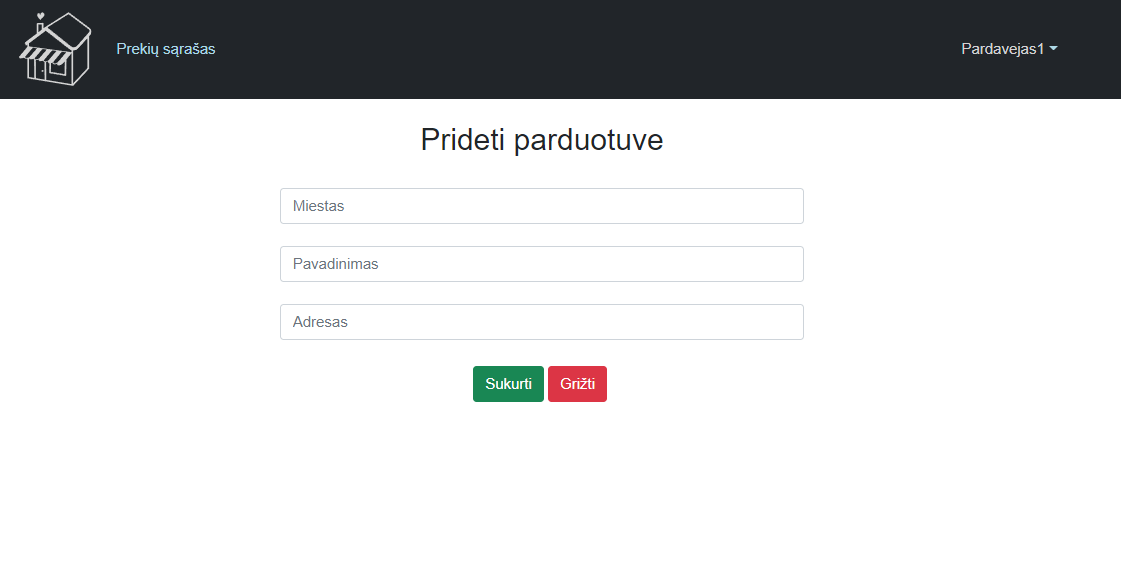
#### Parduotuvės

Paspaudę prekių sąrašas jus būsite nukreipti į pardavėjų sąraša pasirinkę vieną iš pardavėjų būsitę nukreipti į šio pardavėjų prekių sąrašą.

Paveikslėlis, kuriame yra žinutė, ekrano nuotrauka, monitorius

Automatiškai sugeneruotas aprašymas

Pardavėjai gali sukurti naują parduotuvę. Norint pridėti naują parduotuvę reikia užpildyti šiuos laukus miestas, pavadinimas, adresas.



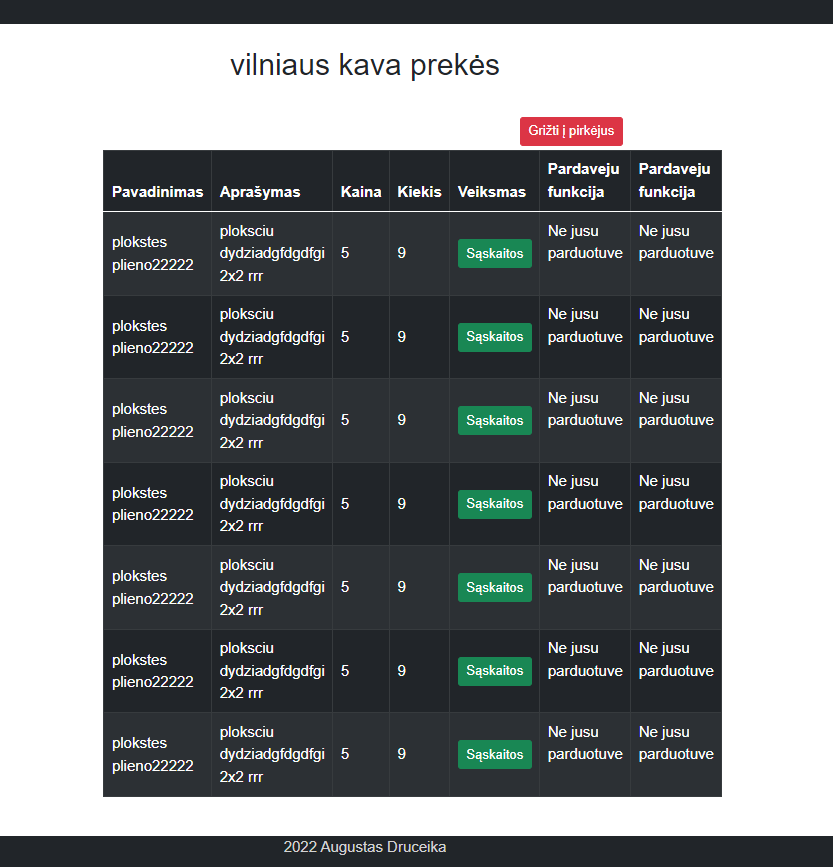
Norint atnaujinti parduotuvės informaciją reikia įrašyti miesta, pavadinimą ir adresa.

Paveikslėlis, kuriame yra žinutė

Automatiškai sugeneruotas aprašymas

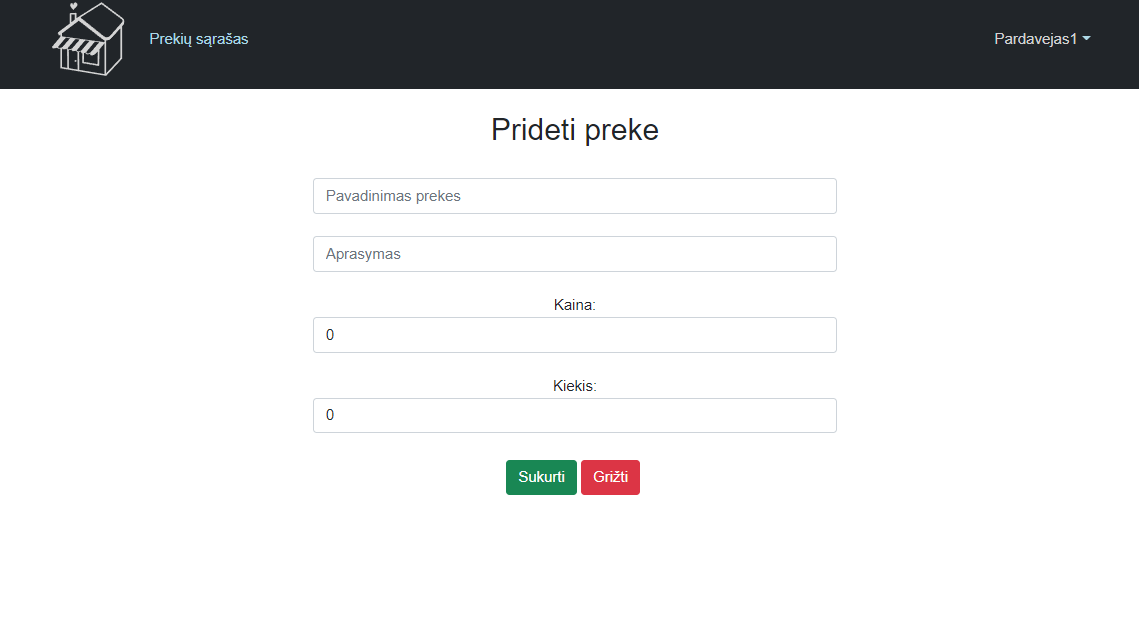
#### Prekės

Pasirinkus parduotuvę busite nukreipti į prekių sąrašą kuriame galėsite pasirinkti prekę.

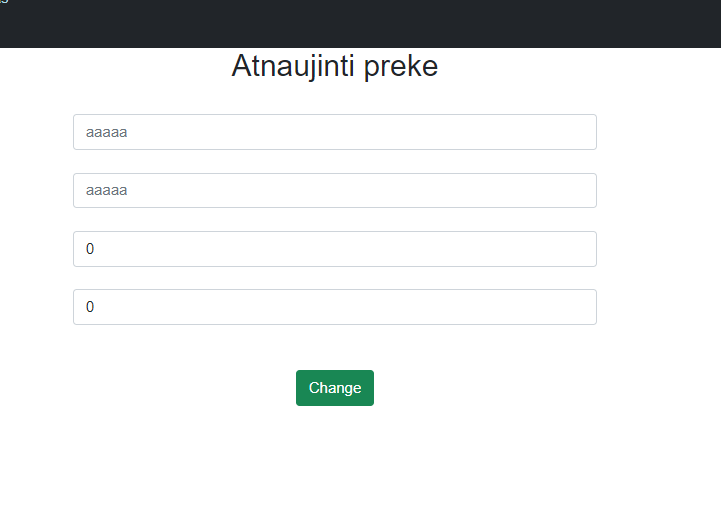




Pardavėjai gali pridėti naujas prekęs. Norint pridėti naujas prekes reikia užpildyti šiuos laukus prekės pavadinimas, aprasymas, kaina ir kieki.



Norint atnaujinti prekės informacija reikia užpildyti šiuos laukus. Pavadinimo, aprašymo, kainos ir kiekio.



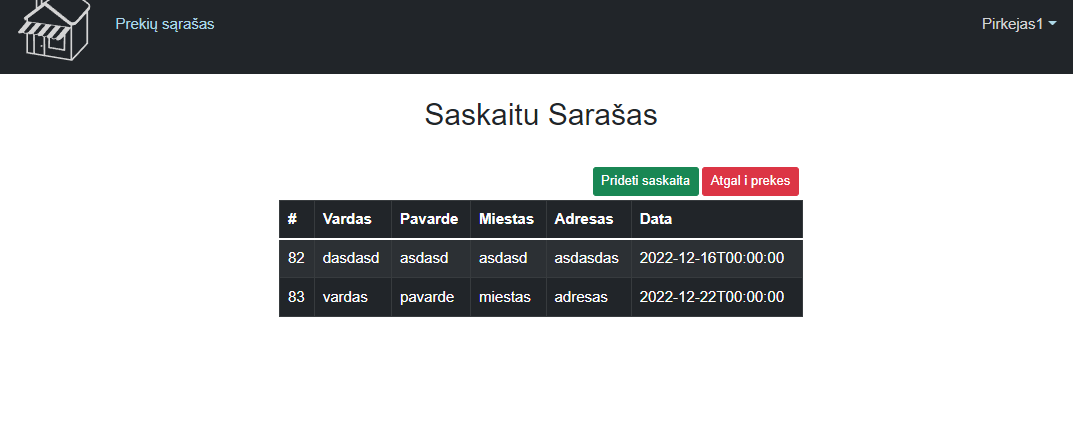
#### Sąskaitos

Pardavėjai negali pridėti naujų sąskaitų.

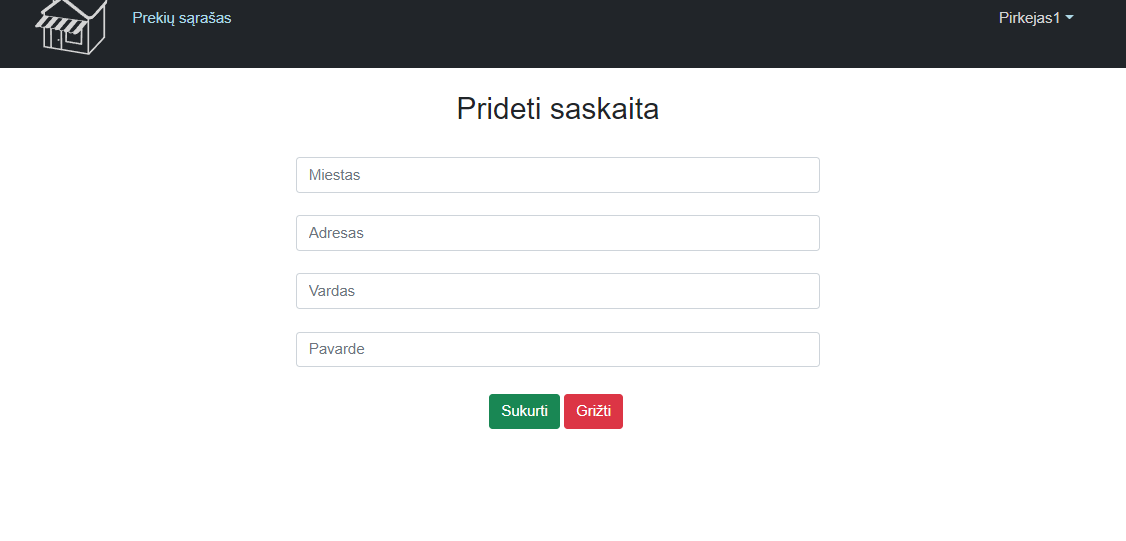
Paveikslėlis, kuriame yra žinutė

Automatiškai sugeneruotas aprašymas

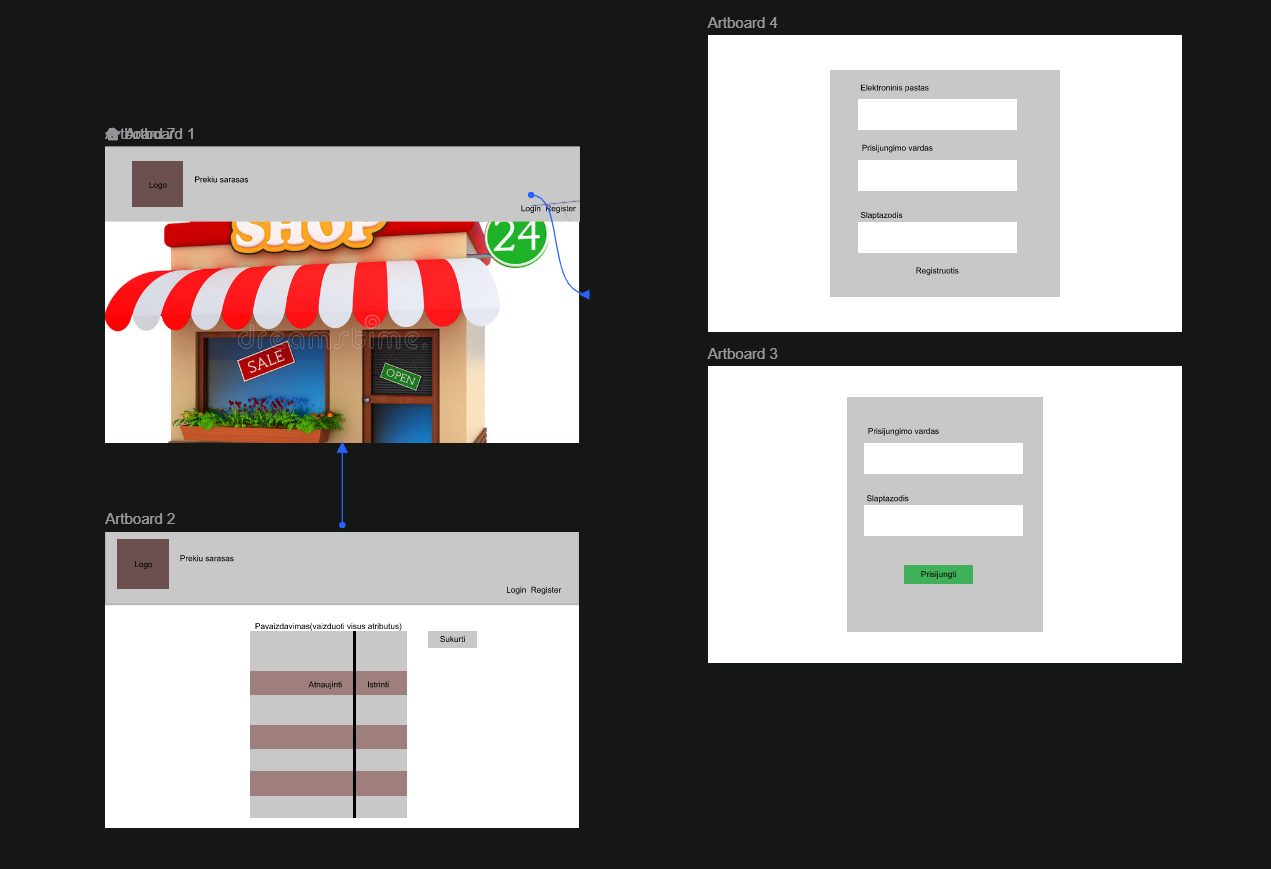
Pirkėjai gali pridėti naujas sąskaitas.



Pridedant naują sąskaita pirkėjas turi užpildyti šiuos laukus: miestas, adresas, vardas, pavardė.



## WireFrame



## Išvados

Reikėtų ši projektą tobulinti, kad jis būtų tobulas, bet šis projektas gavosi kaip prototipas prie kurio padirbėjus būtų galima sukurti tikrai veikianti ir puikų mažmeninės prekybos internetinę parduotuvę. Išmokau kaip dirbti su .net back-end ir taip pat front-end naudojant react.js javascript.

## Projekto repositories

<https://github.com/augis1123/FrontEnd>

https://github.com/augis1123/SaitynoGerasis