

Kauno Technologijos Universitetas

Informatikos Fakultetas

**T125B114 Robotų programavimo technologijos**

Projektinio darbo aprašymas

**Atliko:**

Tadas Laurinaitis

**Priėmė:**

Lekt. Ulinskas Martynas

Lekt. Lukošius Tomas

Kaunas 2019

# Sprendžiamo uždavinio aprašymas

## Sistemos paskirtis

Sistemos paskirtis – patogaus maisto produktų, iš kelių vietų užsakymas viena programa. Antrinė paskirtis – susipažinti su API kūrimo procesu bei išmokti kurti programą tam tikram API.

Projekto tikslas – sukurti maisto prekių užsakymui iš kelių skirtingų tiekėjų, skirtą API bei jį naudojančia aplikaciją, šitaip suteikiant patogią alternatyvą jau esamoms maisto užsakymo programėlėms.

Veikimo principas – kuriamą programinę įrangą sudaro:

1. API (Back-End)
2. API naudojanti programa (Front-End)

Vartotojas, norėdamas naudotis šia maisto užsakymo programėle turės atsidaryti šios programėlės tinklapį bei prisijungti. Prisijungęs vartotojas matys įvairių prekių sąraša, matys atskiras prekių kategorijas, galės kurti, modifikuoti bei turėti keletą krepšelių su skirtingomis prekėmis. Užpildžius krepšelį prekėmis, vartotojas galės jas pirkti bei pasirinkti pristatymą į namus ar kitą vietą.

## Funkciniai reikalavimai

Neregistruotas sistemos vartotojas galės:

* 1. Peržiūrėti pagrindinį puslapį
  2. Užsiregistruoti

Registruotas sistemos vartotojas galės:

1. Prisijungti
2. Atsijungti
3. Peržiūrėti prekių sąrašą
4. Pridėti prekę į krepšelį
5. Peržiūrėti krepšelio prekes
6. Peržiūrėti savo krepšelius

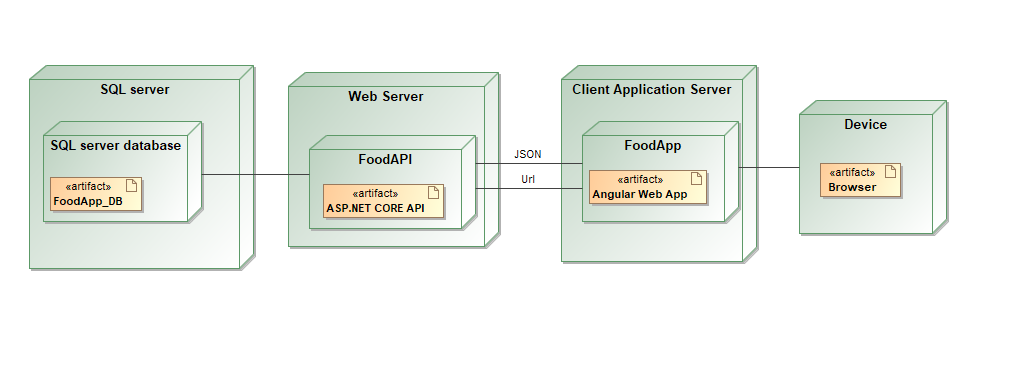
Vartotojas turintis administratoriaus teises galės:

1. Prisijungti
2. Atsijungti
3. Peržiūrėti prekių sąrašą
4. Redaguoti pasirinktą prekę
5. Pridėti naują prekę
6. Ištrinti esamą prekę
7. Peržiūrėti vartotojus
8. Pridėti vartotojus
9. Redaguoti vartotojo duomenis
10. Ištrinti vartotoją
11. Peržiūrėti prekių kategorijas
12. Redaguoti prekių kategorijas
13. Ištrinti prekių kategorijas
14. Pridėti prekių kategorijas

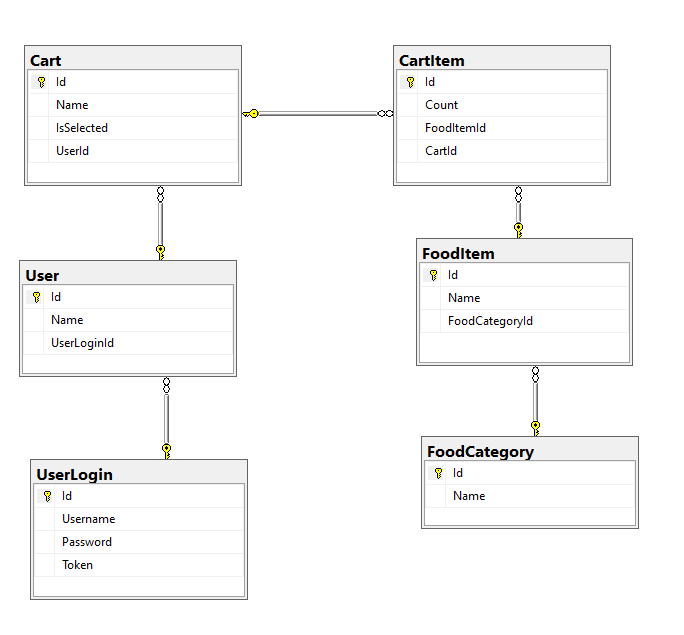
# Sistemos architektūra

Sistemos sudedamosios dalys:

1. Kliento pusė – Front-End – sukurta naudojant NodeJS, TypeScript ir Angular
2. Serverio pusė – Back-End – sukurta naudojant ASP.NET Core v3.1.0, FluentAPI, SQL server 2019 ir Microsoft SQL server management studio 2018



Pav. #1 Sistemos architektūros deployment diagrama



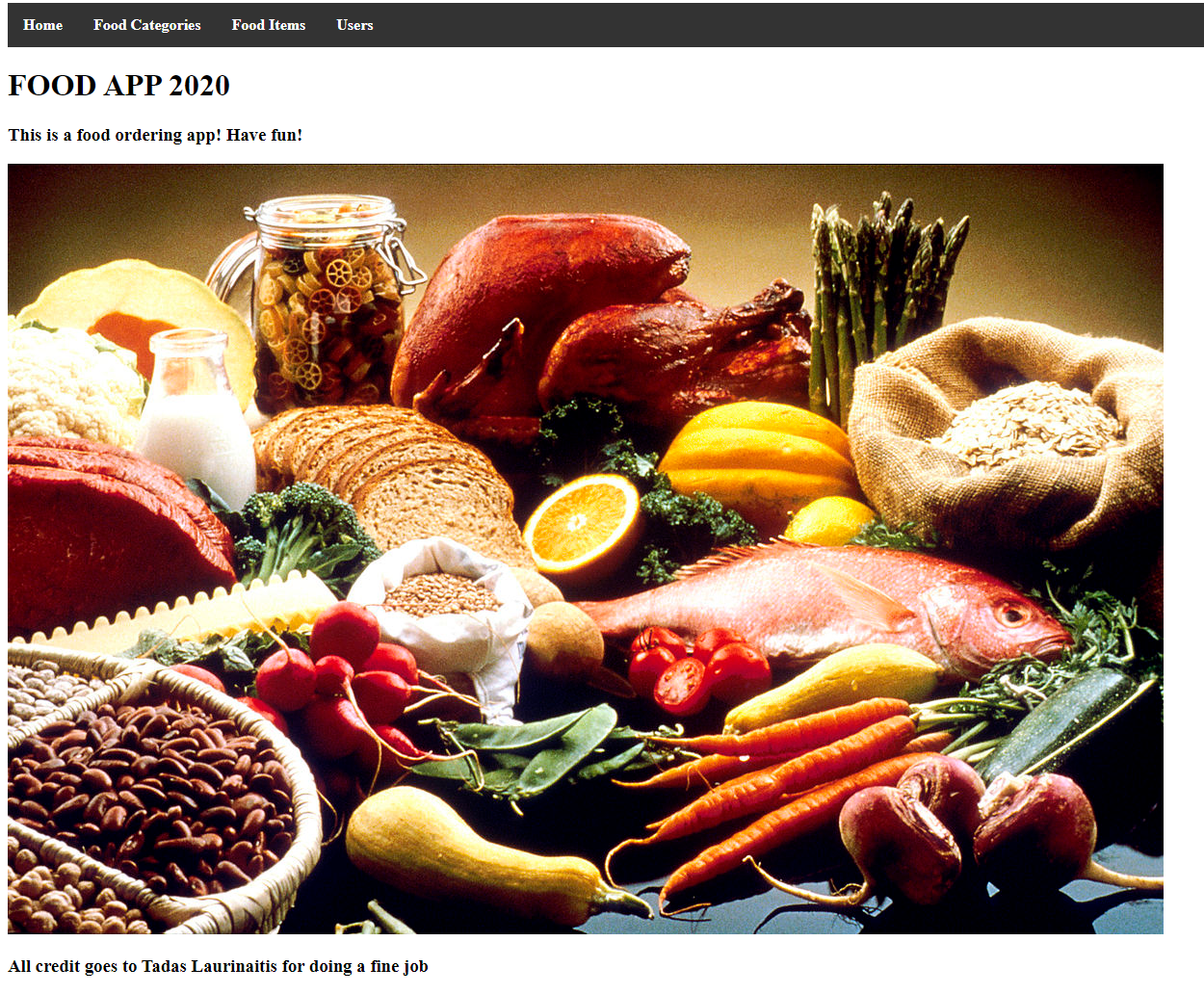
Pav. #2 Sql server duomenų bazės schema

Iš diagramos matosi, kad sistemos failų talpinimui yra naudojamas Microsoft SQL server ir jame sukurta duomenų bazė. Kiekviena sistemos dalis yra diegiama atskirame serveryje. Internetinė aplikacija yra pasiekiama per HTTP protokolą. Internetinė aplikacija per URL bando iš API gauti/duoti reikiamus duomenis bei atlikti reikiamus veiksmus. Vartotojas naudodamas savo įrenginio naršyklę jungiasi prie internetinės aplikacijos.

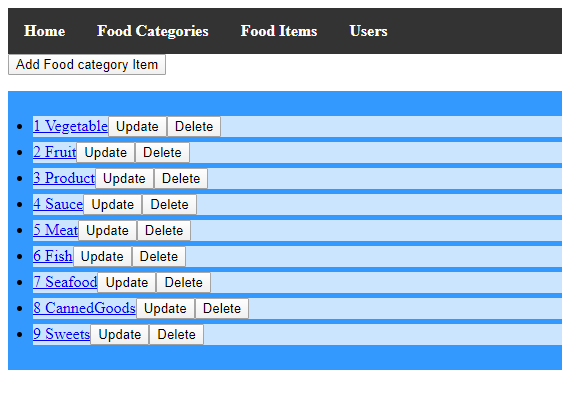
Projekto API ir front-end kodą galite rasti mano github repositorijoje: <https://github.com/sandybridge9/FoodApi>

# Vartotojo sąsajos projektas

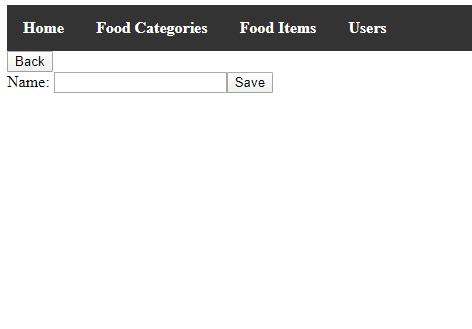
Vartotojo sąsajos projektas buvo daryta naudojantis WebStorm IDE bei NodeJS , Typescript ir Angular. Vartotojo sąsajos projekto pilnai realizuoti nepavyko, kadangi iki galo neveikė autentifikacija, todėl buvo padaryta admin vartotojo sąsaja. Taip pat visas vartotojo sąsajos stilius yra gana netvarkingas, kadangi nelabai žinojau kaip viskas veikia, o išsiaiškinti pritrūko laiko.



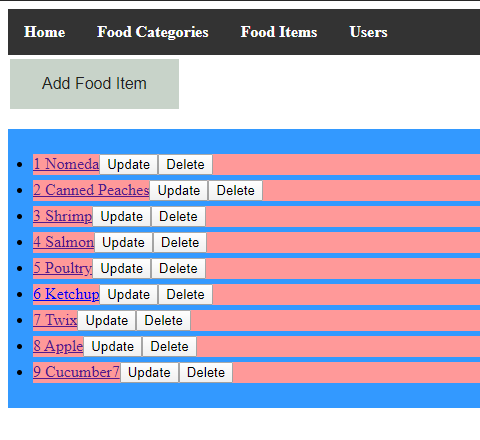
Pav. #2 Pagrindinis namų puslapis.



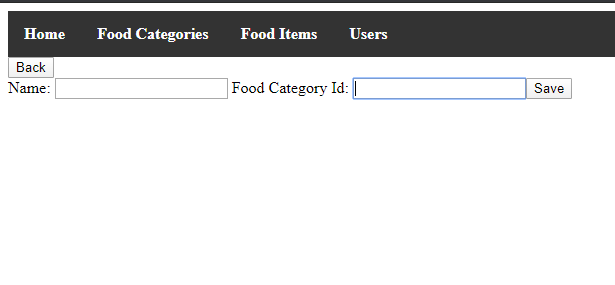
Pav. #3 Maisto kategorijų puslapis

.

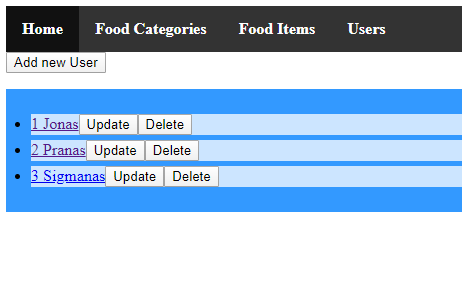
Pav. #4 Maisto kategorijų pridėjimo forma



Pav. #4 Maisto prekių sąrašas su jį valdančiais mygtukais



Pav. #4 Maisto prekės pridėjimo forma



Pav. #4 Vartotojų sąrašas su visais valdančiais mygtukais

# API specifikacija

Tam, kad būtų išlaikomas nuoseklumas, sistemos API metodai bus aprašomi žemiau pateiktos lentelės formatu. Lentelėse bus pateikiama svarbiausia bei būtina informacija sėkmingam užklausos vykdymui. Taip pat pateikiama informacija apie galimas klaidas, kurios gali iškilti apdorojant tam tikrą API užklausą.

|  |  |
| --- | --- |
| API metodas |  |
| Paskirtis |  |
| Kelias iki metodo (route) |  |
| Užklausos struktūra |  |
| Atsakymo struktūra |  |
| Atsakymo kodas |  |
| Galimi klaidų kodai |  |

## Maisto kategorijų klasės API metodai

|  |  |
| --- | --- |
| API metodas | (GET) GetFoodCategories |
| Paskirtis | Gražinti visas maisto kategorijas |
| Kelias iki metodo (route) | /api/FoodCategory |
| Užklausos struktūra | - |
| Atsakymo struktūra | [  {  „Id“: 1,  „Name“: „Vegetable“  },  ...  ] |
| Atsakymo kodas | 200 (OK) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas. |

|  |  |
| --- | --- |
| API metodas | (GET) GetFoodCategory |
| Paskirtis | Gražinti vieną pasirinktą maisto kategoriją |
| Kelias iki metodo (route) | /api/FoodCategory/{id} |
| Užklausos struktūra | - |
| Atsakymo struktūra | {  „Id“: 1,  „Name“: „Vegetable“  } |
| Atsakymo kodas | 200 (OK) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  404 – jeigu neegzistuoja tokia kategorija |

|  |  |
| --- | --- |
| API metodas | (POST) PostFoodCategory |
| Paskirtis | Pridėti naują maisto kategoriją |
| Kelias iki metodo (route) | /api/FoodCategory |
| Užklausos struktūra | {  „Name“: „Vegetable“  } |
| Atsakymo struktūra | - |
| Atsakymo kodas | 204 (No content) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  500 – jei iškyla problema/klaida saugant duomenis duomenų bazėje. |

|  |  |
| --- | --- |
| API metodas | (PUT) PutFoodCategory |
| Paskirtis | Pakeisti esamą maisto kategoriją |
| Kelias iki metodo (route) | /api/FoodCategory/id |
| Užklausos struktūra | {  „Id“: 1,  „Name“: „Vegetable“  } |
| Atsakymo struktūra | - |
| Atsakymo kodas | 204 (No content) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  500 – jei iškyla problema/klaida saugant duomenis duomenų bazėje.  404 – jeigu neegzistuoja tokia kategorija |

|  |  |
| --- | --- |
| API metodas | (DELETE) DeleteFoodCategory |
| Paskirtis | Ištrinti esamą maisto kategoriją |
| Kelias iki metodo (route) | /api/FoodCategory/id |
| Užklausos struktūra | - |
| Atsakymo struktūra | - |
| Atsakymo kodas | 204 (No content) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  500 – jei iškyla problema/klaida saugant duomenis duomenų bazėje.  404 – jeigu neegzistuoja tokia kategorija |

## Maisto prekių klasės API metodai

|  |  |
| --- | --- |
| API metodas | (GET) GetFoodItems |
| Paskirtis | Gražinti visas maisto prekes |
| Kelias iki metodo (route) | /api/FoodItem |
| Užklausos struktūra | - |
| Atsakymo struktūra | [  {  „Id“: 1,  „Name“: „Apple“  „FoodCategoryId“ : 1,  „FoodCategory“:  {  „Id“: 1,  „Name“: „Fruit“  }  },  ...  ] |
| Atsakymo kodas | 200 (OK) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas. |

|  |  |
| --- | --- |
| API metodas | (GET) GetFoodItem |
| Paskirtis | Gražinti vieną pasirinktą maisto prekę |
| Kelias iki metodo (route) | /api/FoodItem/{id} |
| Užklausos struktūra | - |
| Atsakymo struktūra | {  „Id“: 1,  „Name“: „Apple“  „FoodCategoryId“ : 1,  „FoodCategory“:  {  „Id“: 1,  „Name“: „Fruit“  }  } |
| Atsakymo kodas | 200 (OK) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  404 – jeigu neegzistuoja tokia kategorija |

|  |  |
| --- | --- |
| API metodas | (POST) PostFoodItem |
| Paskirtis | Pridėti naują maisto prekę |
| Kelias iki metodo (route) | /api/FoodItem |
| Užklausos struktūra | [  {  „Name“: „Vegetable“  }  ] |
| Atsakymo struktūra | - |
| Atsakymo kodas | 204 (No content) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  500 – jei iškyla problema/klaida saugant duomenis duomenų bazėje. |

|  |  |
| --- | --- |
| API metodas | (PUT) PutFoodItem |
| Paskirtis | Pakeisti esamą maisto prekę |
| Kelias iki metodo (route) | /api/FoodItem/id |
| Užklausos struktūra | [  {  „Id“: 1,  „Name“: „Vegetable“  }  ] |
| Atsakymo struktūra | - |
| Atsakymo kodas | 204 (No content) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  500 – jei iškyla problema/klaida saugant duomenis duomenų bazėje.  404 – jeigu neegzistuoja tokia kategorija |

|  |  |
| --- | --- |
| API metodas | (DELETE) DeleteFoodItem |
| Paskirtis | Ištrinti esamą maisto prekę |
| Kelias iki metodo (route) | /api/FoodItem/id |
| Užklausos struktūra | - |
| Atsakymo struktūra | - |
| Atsakymo kodas | 204 (No content) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  500 – jei iškyla problema/klaida saugant duomenis duomenų bazėje.  404 – jeigu neegzistuoja tokia kategorija |

## Krepšelio prekių klasės API metodai

|  |  |
| --- | --- |
| API metodas | (GET) GetCartItems |
| Paskirtis | Gražinti visas krepšelių prekes |
| Kelias iki metodo (route) | /api/CartItem |
| Užklausos struktūra | - |
| Atsakymo struktūra | [  {  „Id“: 1,  „FoodItemId“: 1  „FoodItem“ : {},  „CartId“: 1,  },  ...  ] |
| Atsakymo kodas | 200 (OK) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas. |

|  |  |
| --- | --- |
| API metodas | (GET) GetCartItem |
| Paskirtis | Gražinti vieną pasirinktą krepšelio prekę |
| Kelias iki metodo (route) | /api/CartItem/{id} |
| Užklausos struktūra | - |
| Atsakymo struktūra | {  „Id“: 1,  „FoodItemId“: 1  „FoodItem“ : {},  „CartId“: 1,  } |
| Atsakymo kodas | 200 (OK) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  404 – jeigu neegzistuoja tokia kategorija |

|  |  |
| --- | --- |
| API metodas | (POST) PostCartItem |
| Paskirtis | Pridėti naują krepšelio prekę |
| Kelias iki metodo (route) | /api/CartItem |
| Užklausos struktūra | [  {  „Name“: „Vegetable“  }  ] |
| Atsakymo struktūra | - |
| Atsakymo kodas | 204 (No content) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  500 – jei iškyla problema/klaida saugant duomenis duomenų bazėje. |

|  |  |
| --- | --- |
| API metodas | (PUT) PutCartItem |
| Paskirtis | Pakeisti esamą krepšelio prekę |
| Kelias iki metodo (route) | /api/CartItem/id |
| Užklausos struktūra | [  {  „Id“: 1,  „Name“: „Vegetable“  }  ] |
| Atsakymo struktūra | - |
| Atsakymo kodas | 204 (No content) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  500 – jei iškyla problema/klaida saugant duomenis duomenų bazėje.  404 – jeigu neegzistuoja tokia kategorija |

|  |  |
| --- | --- |
| API metodas | (DELETE) DeleteCartItem |
| Paskirtis | Ištrinti esamą krepšelio prekę |
| Kelias iki metodo (route) | /api/CartItem/id |
| Užklausos struktūra | - |
| Atsakymo struktūra | - |
| Atsakymo kodas | 204 (No content) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  500 – jei iškyla problema/klaida saugant duomenis duomenų bazėje.  404 – jeigu neegzistuoja tokia kategorija |

## Krepšelių klasės API metodai

|  |  |
| --- | --- |
| API metodas | (GET) GetCarts |
| Paskirtis | Gražinti visus krepšelius |
| Kelias iki metodo (route) | /api/Cart |
| Užklausos struktūra | - |
| Atsakymo struktūra | [  {  „Id“: 1,  „UserId“: 1  „User“ : {},  „CartItems“: [{...}, {...}...],  },  ...  ] |
| Atsakymo kodas | 200 (OK) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas. |

|  |  |
| --- | --- |
| API metodas | (GET) GetCart |
| Paskirtis | Gražinti vieną pasirinktą krepšelį |
| Kelias iki metodo (route) | /api/Cart/{id} |
| Užklausos struktūra | - |
| Atsakymo struktūra | {  „Id“: 1,  „UserId“: 1  „User“ : {},  „CartItems“: [{...}, {...}...],  } |
| Atsakymo kodas | 200 (OK) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  404 – jeigu neegzistuoja tokia kategorija |

|  |  |
| --- | --- |
| API metodas | (POST) PostCart |
| Paskirtis | Pridėti naują krepšelį |
| Kelias iki metodo (route) | /api/Cart |
| Užklausos struktūra | {  „UserId“: 1  „User“ : {},  „CartItems“: [{...}, {...}...],  } |
| Atsakymo struktūra | - |
| Atsakymo kodas | 204 (No content) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  500 – jei iškyla problema/klaida saugant duomenis duomenų bazėje. |

|  |  |
| --- | --- |
| API metodas | (PUT) PutCart |
| Paskirtis | Pakeisti esamą krepšelį |
| Kelias iki metodo (route) | /api/Cart/id |
| Užklausos struktūra | {  „Id“: 1,  „UserId“: 1  „User“ : {},  „CartItems“: [{...}, {...}...],  } |
| Atsakymo struktūra | - |
| Atsakymo kodas | 204 (No content) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  500 – jei iškyla problema/klaida saugant duomenis duomenų bazėje.  404 – jeigu neegzistuoja tokia kategorija |

|  |  |
| --- | --- |
| API metodas | (DELETE) DeleteCart |
| Paskirtis | Ištrinti esamą krepšelį |
| Kelias iki metodo (route) | /api/Cart/id |
| Užklausos struktūra | - |
| Atsakymo struktūra | - |
| Atsakymo kodas | 204 (No content) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  500 – jei iškyla problema/klaida saugant duomenis duomenų bazėje.  404 – jeigu neegzistuoja tokia kategorija |

## Vartotojo klasės API metodai

|  |  |
| --- | --- |
| API metodas | (GET) GetUsers |
| Paskirtis | Gražinti visus vartotojus |
| Kelias iki metodo (route) | /api/User |
| Užklausos struktūra | - |
| Atsakymo struktūra | [  {  „Id“: 1,  „Name“: „abusda“  „Password“ : „sadasdasd“,  „Username“: „asdfasd“,  „Carts“: [{...}, {...}...]  },  ...  ] |
| Atsakymo kodas | 200 (OK) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas. |

|  |  |
| --- | --- |
| API metodas | (GET) GetUser |
| Paskirtis | Gražinti vieną pasirinktą vartotoją |
| Kelias iki metodo (route) | /api/User/{id} |
| Užklausos struktūra | - |
| Atsakymo struktūra | {  „Id“: 1,  „Name“: „abusda“  „Password“ : „sadasdasd“,  „Username“: „asdfasd“,  „Carts“: [{...}, {...}...]  } |
| Atsakymo kodas | 200 (OK) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  404 – jeigu neegzistuoja tokia kategorija |

|  |  |
| --- | --- |
| API metodas | (POST) PostUser |
| Paskirtis | Pridėti naują vartotoją |
| Kelias iki metodo (route) | /api/User |
| Užklausos struktūra | {  „Name“: „abusda“  „Password“ : „sadasdasd“,  „Username“: „asdfasd“,  „Carts“: [{...}, {...}...],  } |
| Atsakymo struktūra | - |
| Atsakymo kodas | 204 (No content) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  500 – jei iškyla problema/klaida saugant duomenis duomenų bazėje. |

|  |  |
| --- | --- |
| API metodas | (PUT) PutUser |
| Paskirtis | Pakeisti esamą vartotoją |
| Kelias iki metodo (route) | /api/User/id |
| Užklausos struktūra | {  „Id“: 1,  „Name“: „abusda“  „Password“ : „sadasdasd“,  „Username“: „asdfasd“,  „Carts“: [{...}, {...}...],  } |
| Atsakymo struktūra | - |
| Atsakymo kodas | 204 (No content) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  500 – jei iškyla problema/klaida saugant duomenis duomenų bazėje.  404 – jeigu neegzistuoja tokia kategorija |

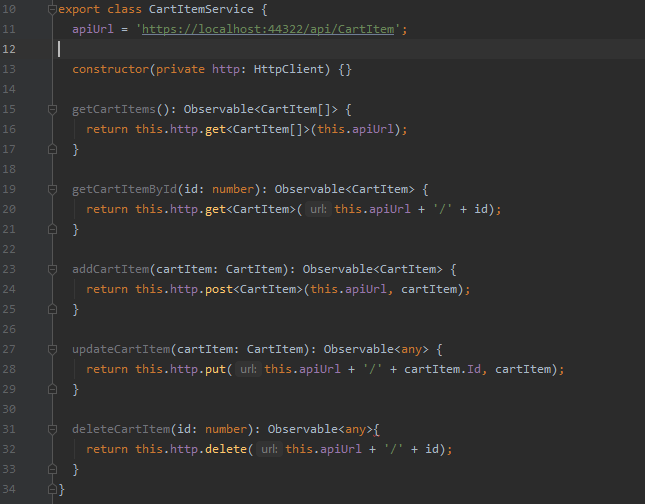
|  |  |
| --- | --- |
| API metodas | (DELETE) DeleteUser |
| Paskirtis | Ištrinti esamą vartotoją |
| Kelias iki metodo (route) | /api/User/id |
| Užklausos struktūra | - |
| Atsakymo struktūra | - |
| Atsakymo kodas | 204 (No content) |
| Galimi klaidų kodai | 401 – jei siunčiamas autorizacijos JWT žetonas yra neteisingas.  500 – jei iškyla problema/klaida saugant duomenis duomenų bazėje.  404 – jeigu neegzistuoja tokia kategorija |

# API panaudojimo pavyzdžiai

Visų klasių API metodai buvo naudojami front-end dalyje, kur visi norimi veiksmai buvo atliekami kreipiantis į naujai susikurtas service klases, kurios yra atsakingos už bendravimą su sukurtu .NET CORE API.



Pav. #2 Krepšelio API panaudojimo pavyzdys front-end serviso pusėje.



Pav. #3 Krepšelio prekių API panaudojimo pavyzdys front-end serviso pusėje.



Pav. #4 Maisto kategorijų API panaudojimo pavyzdys front-end serviso pusėje.



Pav. #5 Maisto prekių API panaudojimo pavyzdys front-end serviso pusėje.



Pav. #6 Vartotojų API panaudojimo pavyzdys front-end serviso pusėje.

# Išvados

* 1. API/Back-End dalį realizuoti gavosi labai gerai. Naudojantis Microsoft pateikta dokumentacija, buvo greitai randami sprendimai iškilusioms problemoms spręsti. Rezultatas gautas geras – veikė visos API dalys ir klasės, išskyrus autentifikaciją (prisijungti eidavo, tačiau JWT negerai išsaugodavo token’us, neradau dėl ko)
  2. Front-end vartotojo sąsają realizuoti sekėsi sunkiai, pastoviai iškildavo visokių neaiškumų. Taip yra dėl to, kad su front-end dalimi esu labai mažai susidūręs, neturiu patirties. Problemų sprendimų ieškojimas trukdavo labai ilgai. Galiausiai buvo rastas geras tutorial’as youtube ir pagal jį viskas padaryta.

# Šaltiniai

<https://github.com/sandybridge9/FoodApi>

<https://www.youtube.com/playlist?list=PL8jcXf-CLpxpatkLTnZZ5dNc6Dvm75JTk>

<https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-2.2&tabs=visual-studio>

<https://docs.microsoft.com/en-us/aspnet/core/data/ef-mvc/intro?view=aspnetcore-2.0>