

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**Deividas Bakanas, Džiugas Molis**

**Ekonomiško šiukšlių išvežimo sistema**

„Sistemų projektavimas Framework aplinkose“ modulio projektas

|  |  |
| --- | --- |
|  | **Vadovas**  Doc. dr. Jevgenijus Toldinas |

**KAUNAS, 2017**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

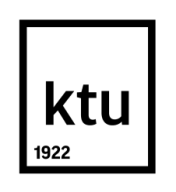
**Ekonomiško šiukšlių išvežimo sistema**

„Sistemų projektavimas Framework aplinkose“ modulio projektas

**Sistemų projektavimas Framework aplinkose (kodas T120B126)**

|  |  |
| --- | --- |
|  | **Vadovas**  (parašas) Doc. dr. Jevgenijus Toldinas  (data) |
| **Recenzentas**  (parašas)  (data) |
| **Projektą atliko**  (parašas) Deividas Bakanas  (parašas) Džiugas Molis  (data) |

**KAUNAS, 2017**



KAUNO TECHNOLOGIJOS UNIVERSITETAS

|  |
| --- |
| Informatikos fakultetas |
| (Fakultetas) |
| Deividas Bakanas |
| (Studento vardas, pavardė) |
|  |
| (Studijų programos pavadinimas, kodas) |

Baigiamojo projekto „Pavadinimas“

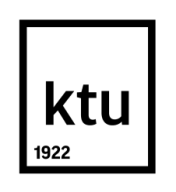
**AKADEMINIO SĄŽININGUMO DEKLARACIJA**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 20 | 17 | m. |  |  |  | d. |
|  |  |  | Kaunas |  |  |  |

Patvirtinu, kad mano, **Deivido Bakano**,baigiamasis projektas tema „..................................................................................................“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| *(vardą ir pavardę įrašyti ranka)* |  | *(parašas)* |



KAUNO TECHNOLOGIJOS UNIVERSITETAS

|  |
| --- |
| Informatikos fakultetas |
| (Fakultetas) |
| Džiugas Molis |
| (Studento vardas, pavardė) |
|  |
| (Studijų programos pavadinimas, kodas) |

Baigiamojo projekto „Pavadinimas“

**AKADEMINIO SĄŽININGUMO DEKLARACIJA**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 20 | 17 | m. |  |  |  | d. |
|  |  |  | Kaunas |  |  |  |

Patvirtinu, kad mano, **Džiugo Molio**,baigiamasis projektas tema „..................................................................................................“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| *(vardą ir pavardę įrašyti ranka)* |  | *(parašas)* |

Bakanas, D. Molis, D. Ekonomiško šiukšlių išvežimo sistema. „Sistemų projektavimas Framework aplinkose“ modulio projektas / vadovas doc. dr. Jevgenijus Toldinas; Kauno technologijos universitetas, Informatikos fakultetas.

Kaunas, 2017. p.

**SANTRAUKA**

*Santrauka bus realizuota 16 savaite pateiktoje ataskaitoje.*

Bakanas, D. Molis D. Economical garbage disposal system. „Systems design in Framework environment“ module project / Supervisor doc. dr. Jevgenijus Toldinas; Kaunas University of Technology, Faculty of Informatics.

Kaunas, 2017. p.

Summary

*Santrauka bus realizuota 16 savaite pateiktoje ataskaitoje.*

Turinys

[Lentelių sąrašas 8](#_Toc501307611)

[Paveikslų sąrašas 9](#_Toc501307612)

[Terminų ir santrumpų žodynas 10](#_Toc501307613)

[Įvadas 11](#_Toc501307614)

[1. Panašių sistemų Analizė 12](#_Toc501307615)

[1.1. Sistema: SmartBin 12](#_Toc501307616)

[1.2. Sistema: Bin-E 12](#_Toc501307617)

[1.3. Sistema: BigBelly: Smart waste system 13](#_Toc501307618)

[2. Projektuojamos sistemos vizija 14](#_Toc501307619)

[2.1. Sistemos vizijos aprašymas 14](#_Toc501307620)

[3. Projektuojamos sitemos panaudos atvėjų diagramos 15](#_Toc501307621)

[4. Reikalavimai sistemai 38](#_Toc501307622)

[4.1.1. Funkciniai reikalavimai 38](#_Toc501307623)

[4.1.2. Nefunkciniai reikalavimai 38](#_Toc501307624)

[5. Informacinės posistemės projektas 39](#_Toc501307625)

[5.1. Informacinės sistemos struktūra 39](#_Toc501307626)

[5.1.1. Duomenų bazė 39](#_Toc501307627)

[5.1.2. WEB klientas 42](#_Toc501307628)

[5.1.3. Serveris 50](#_Toc501307629)

[5.1.4. Serverio klasių diagrama 51](#_Toc501307630)

[5.1.5. Sistemos elgsena 52](#_Toc501307631)

[5.1.6. Serverio API keliai 60](#_Toc501307632)

[5.1.7. Diegimo projektas 64](#_Toc501307633)

[6. Realizacija ir testavimas 65](#_Toc501307634)

[7. Priedai 66](#_Toc501307635)

[7.1. Serverio diegimo vadovas 66](#_Toc501307636)

[7.2. WEB kliento PĮ diegimo vadovas 66](#_Toc501307637)

# Lentelių sąrašas

[Lentelė 1 Sistemų palyginimo lentelė 13](#_Toc501307582)

[Lentelė 2 Prideda EGD Raptor 16](#_Toc501307583)

[Lentelė 3 Mato šiukšlių konteinerių statusus 18](#_Toc501307584)

[Lentelė 4 Konfigūruoja EGD Raptor 19](#_Toc501307585)

[Lentelė 5 Redaguoja konteinerio informaciją 20](#_Toc501307586)

[Lentelė 6 Patvirtina šiukšlių konteinerio statusą 21](#_Toc501307587)

[Lentelė 7 Registruoja šiukšlių vežėjus 22](#_Toc501307588)

[Lentelė 8 Redaguoja šiukšlių vežėjų informaciją 24](#_Toc501307589)

[Lentelė 9 Sudaro vežėjų konteinerių sąrašųs 26](#_Toc501307590)

[Lentelė 10 Prisijungimas 27](#_Toc501307591)

[Lentelė 11 Mato priskirtų konteinerių sąrašą 29](#_Toc501307592)

[Lentelė 12 Sudaro priskirtų konteinerių išvežimo maršrutą 30](#_Toc501307593)

[Lentelė 13 Keičia konteinerio statusą į IŠVEŽTA 31](#_Toc501307594)

[Lentelė 14 Reguliariai tikrina konteinerio būseną 32](#_Toc501307595)

[Lentelė 15 Daro konteinerio nuotrauką 33](#_Toc501307596)

[Lentelė 16 Tikrina padaryta nuotrauką su konteinerio šablonu 34](#_Toc501307597)

[Lentelė 17 Praneša apie konteinerio būseną 35](#_Toc501307598)

[Lentelė 18 Reguliariai tikrina konfigūraciją 36](#_Toc501307599)

[Lentelė 19 Išsaugo konfigūraciją SD kortelėje 37](#_Toc501307600)

[Lentelė 20 CollectorsController metodai bei jų keliai 60](#_Toc501307601)

[Lentelė 21 Collectors\_ContainersController metodai ir jų keliai 61](#_Toc501307602)

[Lentelė 22 Collectors\_TripsController metodai ir jų keliai 61](#_Toc501307603)

[Lentelė 23 ContainersController metodai ir jų keliai 62](#_Toc501307604)

[Lentelė 24 EGDController metodai ir jų keliai 62](#_Toc501307605)

[Lentelė 25 LoginController metodai ir jų keliai 62](#_Toc501307606)

[Lentelė 26 State\_ValuesController metodai ir jų keliai 63](#_Toc501307607)

[Lentelė 27 StatesController metodai ir jų keliai 63](#_Toc501307608)

[Lentelė 28 TripsController metodai ir jų keliai 63](#_Toc501307609)

[Lentelė 29 Trips\_ContainersController metodai ir jų keliai 64](#_Toc501307610)

# Paveikslų sąrašas

[Pav. 1 Metro įmonės užpildymo lygio sensorius 12](#_Toc501307540)

[Pav. 2 Išmani Bin-E šiukšlinė 13](#_Toc501307541)

[Pav. 3 BigBelly miestams skirta šiukšlinė 13](#_Toc501307542)

[Pav. 4 Projektuojamos sistemos vizija 14](#_Toc501307543)

[Pav. 5 Projektuojamos sistemos panaudos atvejų diagrama 15](#_Toc501307544)

[pav. 6 Sistemos struktūra 39](#_Toc501307545)

[pav. 7 Duomenų bazės schema 40](#_Toc501307546)

[Pav. 8 Duomenų judėjimas Flux architektūroje 43](#_Toc501307547)

[Pav. 9 Flux ir MVC architektūrų palyginimas 43](#_Toc501307548)

[Pav. 10 - Vartotojo sąsajos projekto struktūrą 44](#_Toc501307549)

[Pav. 11 Administratoriaus puslapių klasės 44](#_Toc501307550)

[Pav. 12 Administratoriaus pradinio puslapio diagrama 45](#_Toc501307551)

[Pav. 13 Vieno konteinerio puslapio atvaizdavimo klasių diagrama 45](#_Toc501307552)

[Pav. 14 Konteinerių sąrašo puslapio klasių diagrama 46](#_Toc501307553)

[Pav. 15 Vežėjo peržiūros puslapio klasių diagrama 46](#_Toc501307554)

[Pav. 16 Vežėjų sąrašo puslapio klasių diagramos 47](#_Toc501307555)

[Pav. 17 Vežėjo aplinkos puslapių klasių diagrama 47](#_Toc501307556)

[Pav. 18 Vežėjo pradinio puslapio klasių diagrama 48](#_Toc501307557)

[Pav. 19 Vežėjui priskirtų konteinerių atvaizdavimo puslapio klasių diagrama 48](#_Toc501307558)

[Pav. 20 Vežėjo kelionių sąrašo puslapio klasių diagrama 49](#_Toc501307559)

[Pav. 21 Vežėjo kelionės puslapio klasių diagrama 49](#_Toc501307560)

[pav. 22 Dapper ORM veikimo diagrama 50](#_Toc501307561)

[pav. 23 Serverio klasių diagrama 51](#_Toc501307562)

[pav. 24 Prideda EGD sekų diagrama 52](#_Toc501307563)

[pav. 25 Konteinerių statusų peržiūros sekų diagrama 52](#_Toc501307564)

[pav. 26 Konteinerio konfigūracijos sekų diagrama 53](#_Toc501307565)

[pav. 27 EGD konfigūracijos sekų diagrama 53](#_Toc501307566)

[pav. 28 Konteinerio statuso patvirtinimo sekų diagrama 54](#_Toc501307567)

[pav. 29 Šiukšlių vežėjų registracijos sekų diagrama 54](#_Toc501307568)

[pav. 30 Šiukšlių vežėjo redagavimo sekų diagrama 55](#_Toc501307569)

[pav. 31 Vežėjų konteinerių sąrašo sudarymo sekų diagrama 55](#_Toc501307570)

[pav. 32 Vartotojo prisijungimo sekų diagrama 56](#_Toc501307571)

[pav. 33 Priskirtų šiukšlių vežėjų konteinerių sąrašo atvaizdavimo sekų diagrama 56](#_Toc501307572)

[pav. 34 Išvežimo maršruto sudarymo sekų diagrama 56](#_Toc501307573)

[pav. 35 Konteinerio statuso keitimo į išvežta sekų diagrama 57](#_Toc501307574)

[pav. 36 EGD būsenos tikrinimo sekų diagrama 57](#_Toc501307575)

[pav. 37 Konteinerio nuotraukos darymo sekų diagrama 58](#_Toc501307576)

[pav. 38 Nuotraukos lyginimo su šablonu sekų diagrama 58](#_Toc501307577)

[pav. 39 Pranešimo apie konteinerio būseną sekų diagrama 59](#_Toc501307578)

[pav. 40 Konfigūracijos tikrinimo sekų diagrama 59](#_Toc501307579)

[pav. 41 Paleidus serverį su komanda „dotnet run“ 66](#_Toc501307580)

[Pav. 42 CLI aplinkos langas paleidus npm run start 67](#_Toc501307581)

# Terminų ir santrumpų žodynas

**EGD** –Ecomonical garbage disposal – projekto pavadinimo santrumpa anglų kalba.

**Web** – pasaulinis žiniatinklis.

**Single-page application** – internetinės aplikacijos principas, kuris pagal atitinkamą vartotojo veiksmą dinamiškai perrašo esamą puslapį, vietoj to, kad užkrautų vartotojui naują puslapį.

**JavaScript** – objektiškai orientuota, interpretuojama, tvirtų tipų neturinti programavimo kalba, dažnai naudojama interaktyvių internetinių realizacijai, palaikoma modernių naršyklių. Daugiau informacijos: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (nuoroda paskutinį kartą tikrinta 2017-12-16).

**TypeScript** – objektiškai orientuota, tvirtus tipus turinti programavimo kalba, kompiliuojama į JavaScript. Daugiau informacijos: <https://www.typescriptlang.org/> (nuoroda paskutinį kartą tikrinta 2017-12-16).

**React** – JavaScript biblioteka skirta realizuoti vartotojo sąsajoms. Daugiau informacijos: <https://reactjs.org/> (nuoroda paskutinį kartą tikrinta 2017-12-16).

**Flux architektūra** – aplikacijų architektūra, kurios pagrindiniai elementai - Action, Dispatcher, Store, o duomenys šiais elementais keliauja viena kryptimi. Daugiau informacijos: <https://facebook.github.io/flux/> (nuoroda paskutinį kartą tikrinta 2017-12-16).

**MVC architektūra** -aplikacijų architektūra, kurioje pagrindiniai elementai – Controller, Model, View. Pasižymi dvikrypčiu duomenų judėjimu tarp elementų. Daugiau informacijos: <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller> (nuoroda paskutinį kartą tikrinta 2017-12-16).

**PĮ –** programinė įranga.

**CLI** – Command Line Interface – komandinės eilutės vartotojo sąsaja, Windows aplinkoje Command Prompt, Linux aplinkoje Terminal. Daugiau informacijos <https://en.wikipedia.org/wiki/Command-line_interface> (nuoroda paskutinį kartą tikrinta 2017-12-17).

# Įvadas

Nusprendėme kurti išmanios šiukšlių išvežimo sistemos dalį, kuri būtų atsakinga už šiukšlių konteinerio užpildymo lygio nustatymą ir gautų duomenų išsiuntimą sistemos naudotojams.

Idėja kurti šį įrenginį kilo tuomet, kai atkreipėme dėmesį į atliekų tvarkymo įmonių, dirbančiu pagal griežtą grafiką, darbą. Pastebėjome, kad prie gyvenamųjų namų stovintys konteineriai dažnai yra perpildyti, o atokiau stovintys konteineriai yra tušti. Problema ta, kad sudaryti atliekų išvežimo grafikai dažnai neužtikrina sklandaus atliekų išvežimo – atvykstama išgabenti atliekų neatsižvelgus į konteinerio užpildymą, todėl dalis konteinerių persipildo, o kita dalis išvežami praktiškai tušti. Mūsų kuriama sistema padėtų spręsti šio tipo problemas.

Šio projekto tikslas realizuoti sistemą, kuri galėtų patobulinti šiukšlių surinkimo procesą.

Darbo uždaviniai

- Susipažinti su programinės įrangos projektavimo principais;

- Susipažinti su programinės įrangos kūrimu *Framework* aplinkoje;

- Realizuoti informacinę sistemą naudojant kliento – serverio architektūrą;

- Į bendrą sistemą serverio pagalba apjungti skirtingo tipo klientus (Web klientas ir EGD Raptor įrenginys).

Projekto saugyklos URL: <https://github.com/martynasraila/EGD>

# Panašių sistemų Analizė

Pagrindinė šiukšlių išvežimo problema yra informacijos trūkumas apie šiukšlių konteinerių užpildymo būsenas. Dažnai atvykstama ištuštinti tuščių šiukšlių konteinerių arba jie yra perpildyti, o tai sukelia įvairias problemas tiek gyventojams, tiek verslui.

Perpildyti konteineriai yra opi gyventojų problema. Šalia konteinerių paliktos šiukšlės pasklinda aplinkoje dėl vėjo arba gyvūnų veiklos. Šiukšlės skleidžia blogą kvapą, teršia gruntinius vandenis. Tokių šiukšlių surinkimą tenka vykdyti gyvenvietės administracijai arba šiukšlių išvežimo įmonėms.

Kadangi šiukšlių išvežimo įmonės neturi informacijos apie konteinerių užpildymo šiukšlėmis lygi, papildomi įmonės kaštai sunaudojami atvykti iki tuščio arba pusiau pilno šiukšlių konteinerio. Tuo pat metu šių įmonių transportas neefektyviai išnaudodamas kuro sąnaudas teršia aplinką bei švaisto darbuotojų darbo laiką.

## Sistema: SmartBin[[1]](#footnote-1)

Tai pramoninė sistema, kuri ultragarso sensoriaus (pav 1.) pagalba stebi šiukšlių konteinerio užpildymo lygi. Kartu su įmontuotu GPS ir temperatūros siųstuvu ši sistema perduoda informaciją apie šiukšių konteinerį jį aptarnaujančiai įmonei. Sistema veikia bevieliu ryšiu - komunikuoja mobiliojo interneto pagalba. Sistema neturi savo informacinės sistemos – ją kiekvienas vežėjas pagal savo poreikius gali susprojektuoti pats.



Pav. 1 Metro įmonės užpildymo lygio sensorius

## Sistema: Bin-E[[2]](#footnote-2)

Tai sistema skirta išmaniems namams, įmonėms bei įvairioms viešosioms erdvėms. Sistemos pagrindas yra išmani šiukšlinė (pav 2.), kuri pati rūšiuoja atliekas - stiklą, plastiką, popierių, mišrias atliekas, esant galimybei sumažina jų tūrį (suspaudžia plastikinius butelius) ir nukreipia į joms skirtas talpyklas. Kiekvieno išrušiuotų atliekų skyriaus užpildymo lygis siunčiamas į informacinę sistemą. Šiukšlių išvežimo įmonė gali stebėti šiukšlines ir nusprendžia kada atvažiuoti jų ištuštinti.



Pav. 2 Išmani Bin-E šiukšlinė

## Sistema: BigBelly: Smart waste system[[3]](#footnote-3)

Ši sistema yra pritaikyta miestui. Ji keičia įprastas šiukšlines išmaniomis (pav 3.), kurios realiu laiku siunčia informaciją apie šiukšlinės užpildymą. Šiukšlinės pilnumas yra matuojamas sensorių ir kiekvieną kartą pasipildžius šiukslinei jos turinys yra supresuojamas, taip iki 8 kartų padidinama talpa. Naudojant šią sistemą šiukšlės nepasiklis, bei neskleis kvapo. Kūrėjai teigia, jog naudojant tokias išmanias šiukšlines iki 80 % yra sumažinamas išvežimų kiekis bei iki 75% sumažinami išvežimo kaštai. Tai yra pasiekiama sutaupant didelius kiekius kuro.

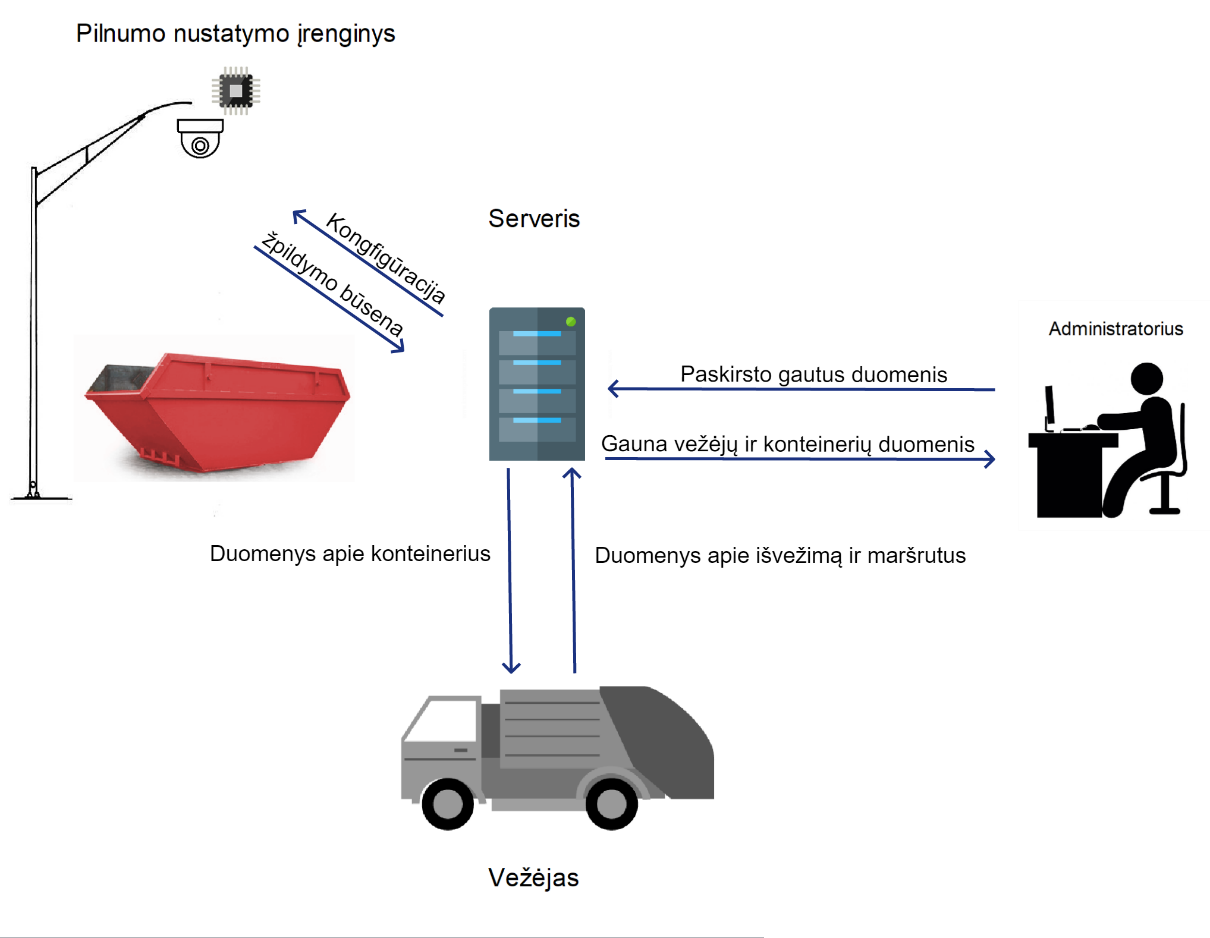


Pav. 3 BigBelly miestams skirta šiukšlinė

Lentelė 1 Sistemų palyginimo lentelė

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Lyginimo kriterijai** | **SmartBin** | **Bin-E** | **BigBelly** | **Mūsų sistema** |
| Užpildymo tikrinimo būdas | Ultragarso sensorius | Sensoriai | Ultragarso sensorius | Vaizdo kamera |
| Informacinė sistema | Realizuoja vartotojas | WEB klientas ir programėlė išmaniajame telefone | Realizuota | WEB klientas |
| Įrenginio apsauga nuo išorės poveikių | Neapsaugota | Apsaugota | Apsaugota | Neapsaugota |
| Reikalingas specialus konteineris / šiukšlinė [[4]](#footnote-4) | Nereikalingas | Reikalingas | Reikalingas | Nereikalingas |
| Rūšiuoja šiukšles | Ne | Taip | Ne | Ne |
| Turinio presavimas[[5]](#footnote-5) | Ne | Dalinis[[6]](#footnote-6) | Taip | Ne |

# Projektuojamos sistemos vizija



Pav. 4 Projektuojamos sistemos vizija

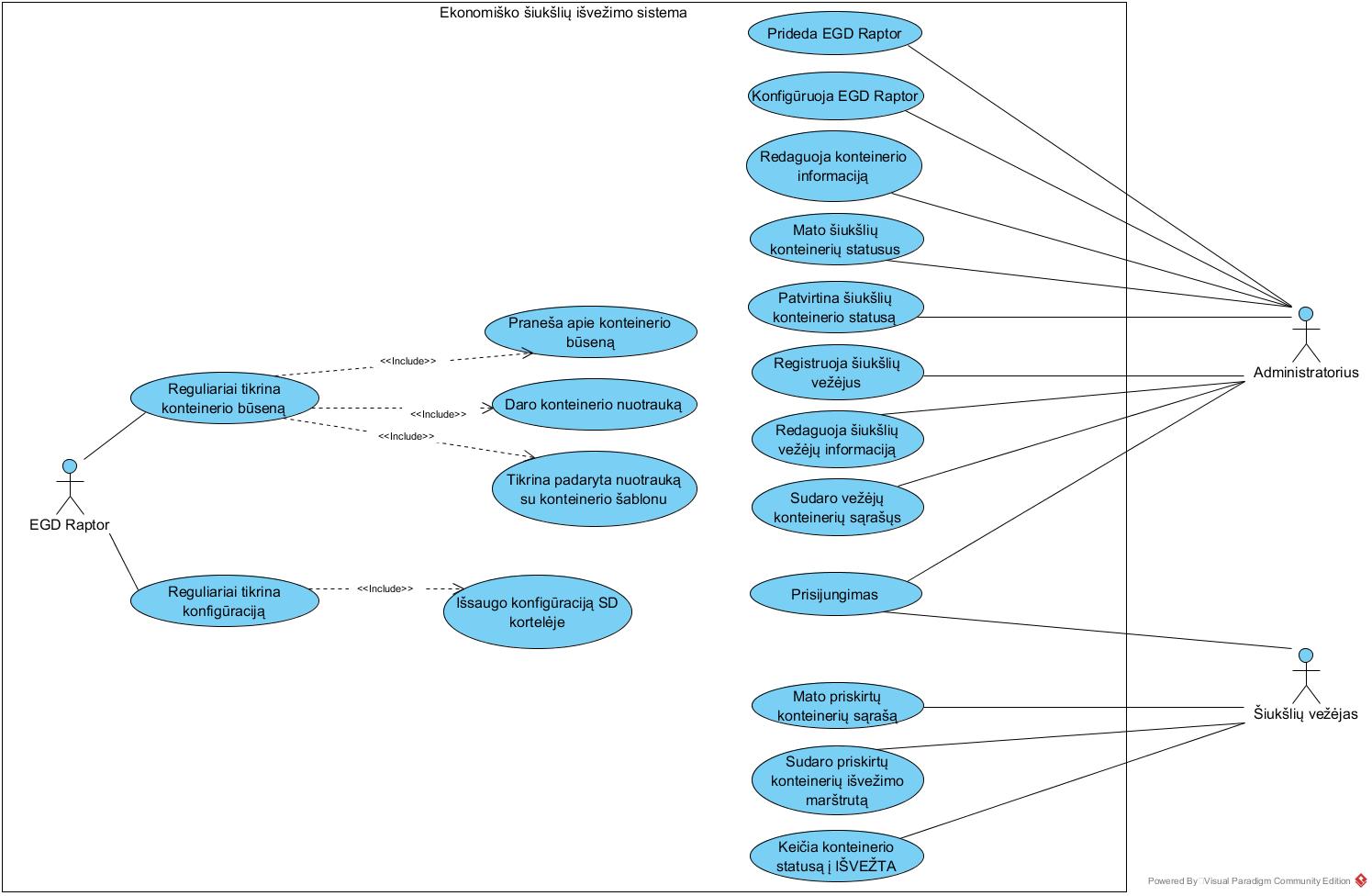
## Sistemos vizijos aprašymas

Projektuojamos sistemos vizijoje pavaizduoti pagrindiniai sistemos dalyvių komunikaciniai ryšiai. EGD Raptor įrenginys stebi šiukšlių konteinerio užpildymą ir reguliariai siunčia duomenis į duomenų bazę.

Iš duomenų bazės informacija yra pasiekiama sistemos naudotojams:

* administratoriui, kuris redaguoja šiukšlių konteinerių informaciją bei prideda šiukšlių išvežimo sunkvežimius
* sunkvežimio ekipažas, kuris gali matyti jiems priskirtų konteinerių būsenas bei lokaciją, susikurti šiukšlių išvežimo maršrutą bei sistemos pagalba pranešti, kad konteineris buvo ištuštintas.

# Projektuojamos sitemos panaudos atvėjų diagramos



Pav. 5 Projektuojamos sistemos panaudos atvejų diagrama

* *EGD Raptor* – sistemos dalis savarankiškai stebinti konteinerio užpildymo būseną.
* *Administratorius* – sistemos naudotojas, kuris koordinuoja *EGD Raptor* informaciją bei *Šiukšlių vežėjus*.
* *Šiukšlių vežėjas* – sistemos naudotojas, gaunantis informaciją apie jam priskirtus konteinerius.

Lentelė Prideda EGD Raptor

|  |  |
| --- | --- |
| **ID** | UC01 |
| **Pavadinimas** | Prideda EGD Raptor |
| **Aprašymas** | Administratorius prie sistemos prideda EGD Raptor įrenginį. |
| **Aktoriai** | Administratorius |
| **Pradinės sąlygos** | Administratorius privalo būti prisijungęs prie sistemos. |
| **Pagrindiniai žingsniai** | Administratorius įveda informaciją apie įrenginį ir paspaudžia mygtuką „Pridėti.” |
| **Alternatyvūs žingsniai** | Nėra |
| **Išskirtinės situacijos** | Jeigu vartotojas įveda neteisingus duomenis arba laukeliai yra tušti, klaidingi laukeliai turi būti pažymėti raudonai. |
| **Galutinės sąlygos** | Jeigu duomenys buvo įvesti teisingai, EDG Raptor pridedamas. |
| **Veiklos diagrama** | C:\Users\Deividas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\22752075_1852594061437001_895177276_n.jpg |
| **Grafinės sąsajos eskizas** | C:\Users\Dziugas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\administratorconteinerviewadd_device.png |

Lentelė 3 Mato šiukšlių konteinerių statusus

|  |  |
| --- | --- |
| **ID** | UC02 |
| **Pavadinimas** | Mato šiukšlių konteinerių statusus |
| **Aprašymas** | Administratorius mato EGD Raptor įrenginių |
| **Aktoriai** | Administratorius |
| **Pradinės sąlygos** | Administratorius privalo būti prisijungęs prie sistemos. |
| **Pagrindiniai žingsniai** | Administratorius pasirinkęs meniu punktą konteineriai mato visą EGD Raptor įrenginių stebimų konteinerių sąrašą su jų užpildymo būsenomis ir pagrindine informacija. |
| **Alternatyvūs žingsniai** | Nėra |
| **Išskirtinės situacijos** | Nėra |
| **Galutinės sąlygos** | Nėra |
| **Veiklos diagrama** | C:\Users\Deividas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\22709988_1852477318115342_120417110_n.jpg |
| **Grafinės sąsajos eskizas** | C:\Users\Dziugas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\administratorcontainers.png |

Lentelė 4 Konfigūruoja EGD Raptor

|  |  |
| --- | --- |
| **ID** | UC04 |
| **Pavadinimas** | Konfigūruoja EGD Raptor |
| **Aprašymas** | Administratorius užpildo visus su konfigūracija susijus laukus. |
| **Aktoriai** | Administratorius |
| **Pradinės sąlygos** | Administratorius privalo būti prisijungęs prie sistemos.  EGD Raptor turi būti pridėtas prie sistemos.  Turi būti atidarytas konteinerių sąrašas. |
| **Pagrindiniai žingsniai** | Administratorius konteinerių sąraše išsirenka norimą konteinerį ir spaudžia mygtuką *Konfigūruoti*. Užpildo konfigūracijos formos laukus ir spaudžia *Konfigūruoti.* |
| **Alternatyvūs žingsniai** | Nėra |
| **Išskirtinės situacijos** | Jeigu vartotojas įveda neteisingus duomenis arba laukeliai yra tušti, klaidingi laukeliai turi būti pažymėti raudonai. |
| **Galutinės sąlygos** | Jeigu duomenys buvo įvesti teisingai, konfigūracijos duomenys išsaugomi. |
| **Veiklos diagrama** |  |
| **Grafinės sąsajos eskizas** | C:\Users\Dziugas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\administratorconteinerview.png |

Lentelė 5 Redaguoja konteinerio informaciją

|  |  |
| --- | --- |
| **ID** | UC06 |
| **Pavadinimas** | Redaguoja konteinerio informaciją |
| **Aprašymas** | Administratorius užpildo visus su konteinerio informacija susijus laukus. |
| **Aktoriai** | Administratorius |
| **Pradinės sąlygos** | Administratorius privalo būti prisijungęs prie sistemos.  EGD Raptor turi būti pridėtas prie sistemos.  Turi būti atidarytas konteinerių sąrašas. |
| **Pagrindiniai žingsniai** | Administratorius konteinerių sąraše išsirenka norimą konteinerį ir spaudžia mygtuką *Redaguoti*. Užpildo konfiguracijos formos laukus ir spaudžia *Redaguoti.* |
| **Alternatyvūs žingsniai** | Nėra |
| **Išskirtinės situacijos** | Jeigu vartotojas įveda neteisingus duomenis arba laukeliai yra tušti, klaidingi laukeliai turi būti pažymėti raudonai. |
| **Galutinės sąlygos** | Jeigu duomenys buvo įvesti teisingai, konteinerio duomenys išsaugomi. |
| **Veiklos diagrama** |  |
| **Grafinės sąsajos eskizas** | C:\Users\Dziugas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\administratorconteinerview_no_device.png |

Lentelė 6 Patvirtina šiukšlių konteinerio statusą

|  |  |
| --- | --- |
| **ID** | UC07 |
| **Pavadinimas** | Patvirtina šiukšlių konteinerio statusą |
| **Aprašymas** | Administratorius patvirtina, kad konkretų konteinerį galima perduoti vežėjui. |
| **Aktoriai** | Administratorius |
| **Pradinės sąlygos** | Administratorius privalo būti prisijungęs prie sistemos.  EGD Raptor turi būti pridėtas prie sistemos.  Turi būti atidarytas konteinerių sąrašas.  Pasirinktas konteineris turi būti pilnas. |
| **Pagrindiniai žingsniai** | Administratorius konteinerių sąraše išsirenka norimą konteinerį ir spaudžia mygtuką *Patvirtinti*. |
| **Alternatyvūs žingsniai** | Nėra |
| **Išskirtinės situacijos** | Jeigu vartotojas bando patvirtinti tuščią konteinerį į vartotojo sąsaja išvedamas pranešimas apie netinkamą veiksmą. |
| **Galutinės sąlygos** | Nėra |
| **Veiklos diagrama** |  |
| **Grafinės sąsajos eskizas** | C:\Users\Dziugas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\administratorconteinerview.png |

Lentelė 7 Registruoja šiukšlių vežėjus

|  |  |
| --- | --- |
| **ID** | UC03 |
| **Pavadinimas** | Registruoja šiukšlių vežėjus |
| **Aprašymas** | Administratorius užpildo visus vežėjo registracijai reikalingus laukus. |
| **Aktoriai** | Administratorius |
| **Pradinės sąlygos** | Administratorius privalo būti prisijungęs prie sistemos.  Turi būti atidarytas vežėjų sąrašas. |
| **Pagrindiniai žingsniai** | Administratorius vežėjų sąraše spaudžia mygtuką *Pridėti vežėją,* užpildo vežėjo registracijos formos laukus ir spaudžia mygtuką *Pridėti*. |
| **Alternatyvūs žingsniai** | Nėra |
| **Išskirtinės situacijos** | Jeigu vartotojas įveda neteisingus duomenis arba laukeliai yra tušti, klaidingi laukeliai turi būti pažymėti raudonai. |
| **Galutinės sąlygos** | Jeigu duomenys buvo įvesti teisingai, vežėjas priregistruojamas. |
| **Veiklos diagrama** | C:\Users\Deividas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\22751934_1852592054770535_698332467_n.jpg |
| **Grafinės sąsajos eskizas** | C:\Users\Dziugas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\administratorcollectorscreate.png |

Lentelė 8 Redaguoja šiukšlių vežėjų informaciją

|  |  |
| --- | --- |
| **ID** | UC05 |
| **Pavadinimas** | Redaguoja šiukšlių vežėjų informaciją |
| **Aprašymas** | Administratorius užpildo vežėjo redagavimo formos laukus. |
| **Aktoriai** | Administratorius |
| **Pradinės sąlygos** | Administratorius privalo būti prisijungęs prie sistemos.  Turi būti atidarytas vežėjų sąrašas.  Redaguojamas vežėjas turi būti registruotas sistemoje. |
| **Pagrindiniai žingsniai** | Administratorius vežėjų sąraše spaudžia mygtuką *Redaguoti,* užpildo vežėjo redagavimo formos laukus ir spaudžia mygtuką *Redaguoti*. |
| **Alternatyvūs žingsniai** | Nėra |
| **Išskirtinės situacijos** | Jeigu vartotojas įveda neteisingus duomenis arba laukeliai yra tušti, klaidingi laukeliai turi būti pažymėti raudonai. |
| **Galutinės sąlygos** | Jeigu duomenys buvo įvesti teisingai, vežėjo duomenys išsaugomi. |
| **Veiklos diagrama** |  |
| **Grafinės sąsajos eskizas** | C:\Users\Dziugas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\administratorcollectorsview.png |

Lentelė 9 Sudaro vežėjų konteinerių sąrašųs

|  |  |
| --- | --- |
| **ID** | UC10 |
| **Pavadinimas** | Sudaro vežėjų konteinerių sąrašųs |
| **Aprašymas** | Administratorius prideda konteinerius pasirinktam vežėjui. |
| **Aktoriai** | Administratorius |
| **Pradinės sąlygos** | Administratorius privalo būti prisijungęs prie sistemos.  Turi būti atidarytas vežėjų sąrašas.  Vežėjas turi būti registruotas sistemoje. |
| **Pagrindiniai žingsniai** | Administratorius vežėjų sąraše spaudžia mygtuką *Pridėti konteinerius* ir pasirenka konteinerius. |
| **Alternatyvūs žingsniai** | Nėra |
| **Išskirtinės situacijos** | Nėra |
| **Galutinės sąlygos** | Nėra |
| **Veiklos diagrama** |  |
| **Grafinės sąsajos eskizas** | C:\Users\Dziugas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\administratorcollectorsviewadd_container.png |

Lentelė 10 Prisijungimas

|  |  |
| --- | --- |
| **ID** | UC18 |
| **Pavadinimas** | Prisijungimas |
| **Aprašymas** | Administratorius arba vežėjas prisijungia prie sistemos suvesdamas savo prisijungimo duomenis. |
| **Aktoriai** | Administratorius, Vežėjas |
| **Pradinės sąlygos** | Nėra |
| **Pagrindiniai žingsniai** | Administratorius arba vežėjas prisijungia įveda savo prisijungimo duomenis. Paspaudžia mygtuką prisijungti. |
| **Alternatyvūs žingsniai** | Nėra |
| **Išskirtinės situacijos** | Jeigu įvesti prisijungimo duomenys neteisingi, į vartotojo sąsają išvedama žinutė. |
| **Galutinės sąlygos** | Jeigu įvesti prisijungimo duomenys teisingi, suteikiama prieiga prie sistemos. |
| **Veiklos diagrama** | C:\Users\Deividas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\22711463_1852589308104143_1543812564_n.jpg |
| **Grafinės sąsajos eskizas** | C:\Users\Dziugas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\log_in.png |

Lentelė 11 Mato priskirtų konteinerių sąrašą

|  |  |
| --- | --- |
| **ID** | UC18 |
| **Pavadinimas** | Mato priskirtų konteinerių sąrašą |
| **Aprašymas** | Vežėjas mato jam Administroriaus priskirtus konteinerius. |
| **Aktoriai** | Vežėjas |
| **Pradinės sąlygos** | Vežėjas turi būti registruotas sistemoje.  Vežėjas turi būti prisijungęs prie sistemos. |
| **Pagrindiniai žingsniai** | Vežėjas pasirenka meniu punktą *Konteineriai.* Atidaromas šiam vežėjui priskirtų konteinerių sąrašas. |
| **Alternatyvūs žingsniai** | Nėra |
| **Išskirtinės situacijos** | Jeigu įvesti prisijungimo duomenys neteisingi, į vartotojo sąsają išvedama žinutė. |
| **Galutinės sąlygos** | Jeigu įvesti prisijungimo duomenys teisingi, suteikiama prieiga prie sistemos. |
| **Veiklos diagrama** | C:\Users\Deividas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\22782274_1852582098104864_1421990448_n.jpg |
| **Grafinės sąsajos eskizas** | https://scontent.fkun1-1.fna.fbcdn.net/v/t35.0-12/23335624_1791171170895720_246995918_o.png?oh=ac66a96f44b3d3eb6c81c96b95e93b18&oe=5A01F2AD |

Lentelė Sudaro priskirtų konteinerių išvežimo maršrutą

|  |  |
| --- | --- |
| **ID** | UC11 |
| **Pavadinimas** | Sudaro priskirtų konteinerių išvežimo maršrutą |
| **Aprašymas** | Vežėjas pasirenka konteinerių apvažiavimo eiliškumą |
| **Aktoriai** | Vežėjas |
| **Pradinės sąlygos** | Vežėjas turi būti registruotas sistemoje.  Vežėjas turi būti prisijungęs prie sistemos. |
| **Pagrindiniai žingsniai** | Vežėjas pasirenka meniu punktą *Maršrutai.* Atidaromas šiam vežėjui priskirtų pilnų konteinerių sąrašas. Vežėjas renkasi konteinerių apvažiavimo eiliškumą. Sąrašas patvirtinamas. |
| **Alternatyvūs žingsniai** | Nėra |
| **Išskirtinės situacijos** | Nėra |
| **Galutinės sąlygos** | Nėra |
| **Veiklos diagrama** |  |
| **Grafinės sąsajos eskizas** | https://scontent.fkun1-1.fna.fbcdn.net/v/t35.0-12/23314463_1791190184227152_647079458_o.png?oh=5b3afa021464d5fccee137f7d346121c&oe=5A01CB47 |

Lentelė 13 Keičia konteinerio statusą į IŠVEŽTA

|  |  |
| --- | --- |
| **ID** | UC12 |
| **Pavadinimas** | Keičia konteinerio statusą į IŠVEŽTA |
| **Aprašymas** | Ištuštinęs konteinerį vežėjas praneša apie įvykdytą veiksmą. |
| **Aktoriai** | Vežėjas |
| **Pradinės sąlygos** | Vežėjas turi būti registruotas sistemoje.  Vežėjas turi būti prisijungęs prie sistemos. |
| **Pagrindiniai žingsniai** | Vežėjas pasirenka meniu punktą *Konteineriai.* Atidaromas šiam vežėjui priskirtų konteinerių sąrašas. Spaudžiamas mygtukas IŠVEŽTA. |
| **Alternatyvūs žingsniai** | Nėra |
| **Išskirtinės situacijos** | Nėra |
| **Galutinės sąlygos** | Nėra |
| **Veiklos diagrama** |  |
| **Grafinės sąsajos eskizas** | https://scontent.fkun1-1.fna.fbcdn.net/v/t35.0-12/23314463_1791190184227152_647079458_o.png?oh=5b3afa021464d5fccee137f7d346121c&oe=5A01CB47 |

Lentelė 14 Reguliariai tikrina konteinerio būseną

|  |  |
| --- | --- |
| **ID** | UC08 |
| **Pavadinimas** | Reguliariai tikrina konteinerio būseną |
| **Aprašymas** | Patikrina konteinerio būseną ir informuoja apie ją. |
| **Aktoriai** | EGD Raptor |
| **Pradinės sąlygos** | EGD Raptor turi būti priregistruotas sistemoje. |
| **Pagrindiniai žingsniai** | EGD Raptor atlieka konteinerio nuotrauką, ją lygina su SD kortelėje saugomu šablonu. |
| **Alternatyvūs žingsniai** | Nėra |
| **Išskirtinės situacijos** | Jeigu konteineris tuščias arba pusiau pilnas pranešimas nesiunčiamas. |
| **Galutinės sąlygos** | Jeigu konteineris pilnas arba perpildytas informuojama apie šią būseną. |
| **Veiklos diagrama** | C:\Users\Deividas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\22711181_1852643151432092_172041630_n.jpg |

Lentelė 15 Daro konteinerio nuotrauką

|  |  |
| --- | --- |
| **ID** | UC14 |
| **Pavadinimas** | Daro konteinerio nuotrauką |
| **Aprašymas** | Atlieka konteinerio nuotrauką. |
| **Aktoriai** | EGD Raptor |
| **Pradinės sąlygos** | EGD Raptor turi būti priregistruotas sistemoje. |
| **Pagrindiniai žingsniai** | EGD Raptor atlieka konteinerio nuotrauką, sugeneruoja BITMAP. |
| **Alternatyvūs žingsniai** | Nėra |
| **Išskirtinės situacijos** | Nėra |
| **Galutinės sąlygos** | Nėra |
| **Veiklos diagrama** | C:\Users\Deividas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\22752195_1852649664764774_592113671_n.jpg |

Lentelė 16 Tikrina padaryta nuotrauką su konteinerio šablonu

|  |  |
| --- | --- |
| **ID** | UC17 |
| **Pavadinimas** | Tikrina padaryta nuotrauką su konteinerio šablonu |
| **Aprašymas** | Sugeneruoja konteinerio šablono BITMAP ir jį lygina su dabar padarytu BITMAP. |
| **Aktoriai** | EGD Raptor |
| **Pradinės sąlygos** | EGD Raptor turi būti priregistruotas sistemoje. |
| **Pagrindiniai žingsniai** | EGD Raptor sugeneruoja šablono BITMAP, jį lygina su dabar padarytos nuotraukos BITMAP. |
| **Alternatyvūs žingsniai** | Nėra |
| **Išskirtinės situacijos** | Nėra |
| **Galutinės sąlygos** | Nėra |
| **Veiklos diagrama** | C:\Users\Deividas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\22711615_1852652714764469_1379139451_n.jpg |

Lentelė 17 Praneša apie konteinerio būseną

|  |  |
| --- | --- |
| **ID** | UC15 |
| **Pavadinimas** | Praneša apie konteinerio būseną |
| **Aprašymas** | Atsižvelgiant į konteinerio būsena siunčiamas pranešama į duomenų bazę. |
| **Aktoriai** | EGD Raptor |
| **Pradinės sąlygos** | EGD Raptor turi būti priregistruotas sistemoje. |
| **Pagrindiniai žingsniai** | Jeigu nustatyta būsena yra PILNAS arba PERPILDYTAS – pranešimas siunčiamas, kitu atveju laukiama kito veiklos ciklo. |
| **Alternatyvūs žingsniai** | Nėra |
| **Išskirtinės situacijos** | Nėra |
| **Galutinės sąlygos** | Nėra |
| **Veiklos diagrama** | C:\Users\Deividas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\22752180_1852672181429189_1382915776_n (1).jpg |

Lentelė 18 Reguliariai tikrina konfigūraciją

|  |  |
| --- | --- |
| **ID** | UC13 |
| **Pavadinimas** | Reguliariai tikrina konfigūraciją |
| **Aprašymas** | Tikrinama ar EGD Raptor turi naujausią konfiguraciją. |
| **Aktoriai** | EGD Raptor |
| **Pradinės sąlygos** | EGD Raptor turi būti priregistruotas sistemoje. |
| **Pagrindiniai žingsniai** | Atliekama konfigūracijos užklausa į serverį. |
| **Alternatyvūs žingsniai** | Nėra |
| **Išskirtinės situacijos** | Nėra |
| **Galutinės sąlygos** | Jei konfigūracija atsinaujino, ji išsaugoma SD kortelėje. |
| **Veiklos diagrama** | C:\Users\Deividas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\22773572_1852676051428802_893548102_n.jpg |
| **Grafinės sąsajos eskizas** |  |

Lentelė 19 Išsaugo konfigūraciją SD kortelėje

|  |  |
| --- | --- |
| **ID** | UC16 |
| **Pavadinimas** | Išsaugo konfigūraciją SD kortelėje |
| **Aprašymas** | Gauta nauja konfigūracija išsaugoma SD kortelėje. |
| **Aktoriai** | EGD Raptor |
| **Pradinės sąlygos** | EGD Raptor turi būti priregistruotas sistemoje. |
| **Pagrindiniai žingsniai** | Gaunama nauja konfigūracija. Saugojimas SD kortelėje. |
| **Alternatyvūs žingsniai** | Nėra |
| **Išskirtinės situacijos** | Nėra |
| **Galutinės sąlygos** | Nėra |
| **Veiklos diagrama** | C:\Users\Deividas\AppData\Local\Microsoft\Windows\INetCache\Content.Word\22773607_1852667711429636_923830561_n.jpg |
| **Grafinės sąsajos eskizas** |  |

# Reikalavimai sistemai

### Funkciniai reikalavimai

* Turi būti sukurtos Vežėjo ir Administratoriaus vartotojų aplinkos.
* Prie vartotojo aplinkos turi būti prisijungiama vartotojo vardu bei slaptažodžiu.
* Vartotojas (tiek Administratorius, tiek Vežėjas) turi matyti bendrą sistemos statistiką.
* Administratorius turi matyti prie sistemos priregistruotus vežėjus.
* Administratorius turi galimybę pridėti vežėjus.
* Administratorius turi galimybę pridėti vežėjams konteinerius.
* Administratorius turi matyti prie sistemos pridėtus konteinerius.
* Administratorius turi galimybę prie sistemos pridėti konteinerius.
* Administratorius turi galimybę prie sistemoje esančio konteinerio pridėti EGD įrenginius.
* Vežėjas turi matyti jam priskirtus konteinerius.
* Vežėjas turi galimybę sukurti keliones.
* Vežėjas turi galimybę prie kelionės pridėti konteinerius ir keisti jų aplankymo tvarką.

### Nefunkciniai reikalavimai

Reikalavimai vartotojo sąsajai:

* Vartotojo sąsajos valdymo elementai turi būti standartinei ir palaikomi modernių naršyklių[[7]](#footnote-7).
* Vartotojo sąsaja turi veikti moderniose naršyklėse.
* Vartotojo sąsaja turi veikti be papildomų puslapio perkrovimų.

Reikalavimai programavimo kalboms

* Projekto baziniai elementai turi būti parašyti tvirtus tipus turinčia programavimo kalba.
* Projekto baziniai elementai turi būti parašyti objektiškai orientuota programavimo kalba.

Reikalavimai saugai

* Duomenų bazėje turi būti saugojamas vartotojo slaptažodžio maišos funkcijos rezultatas, ne pats slaptažodis.

# Informacinės posistemės projektas

## Informacinės sistemos struktūra

Informacinė sistema yra realizuota kliento – serverio architektūra. Kaip matyti šeštame paveikslėlyje sistemą sudaro du klientai:

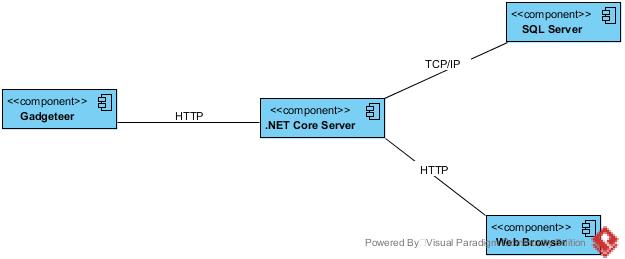
**WEB klientas**

Apima vartotojo sąsaja, vartotojo funkcionalumą su sistemoje esančiais duomenimis.

**EGD įrenginys – klientas**

Atlieka konteinerio užpildymo matavimus, siunčia duomenis į serverį.

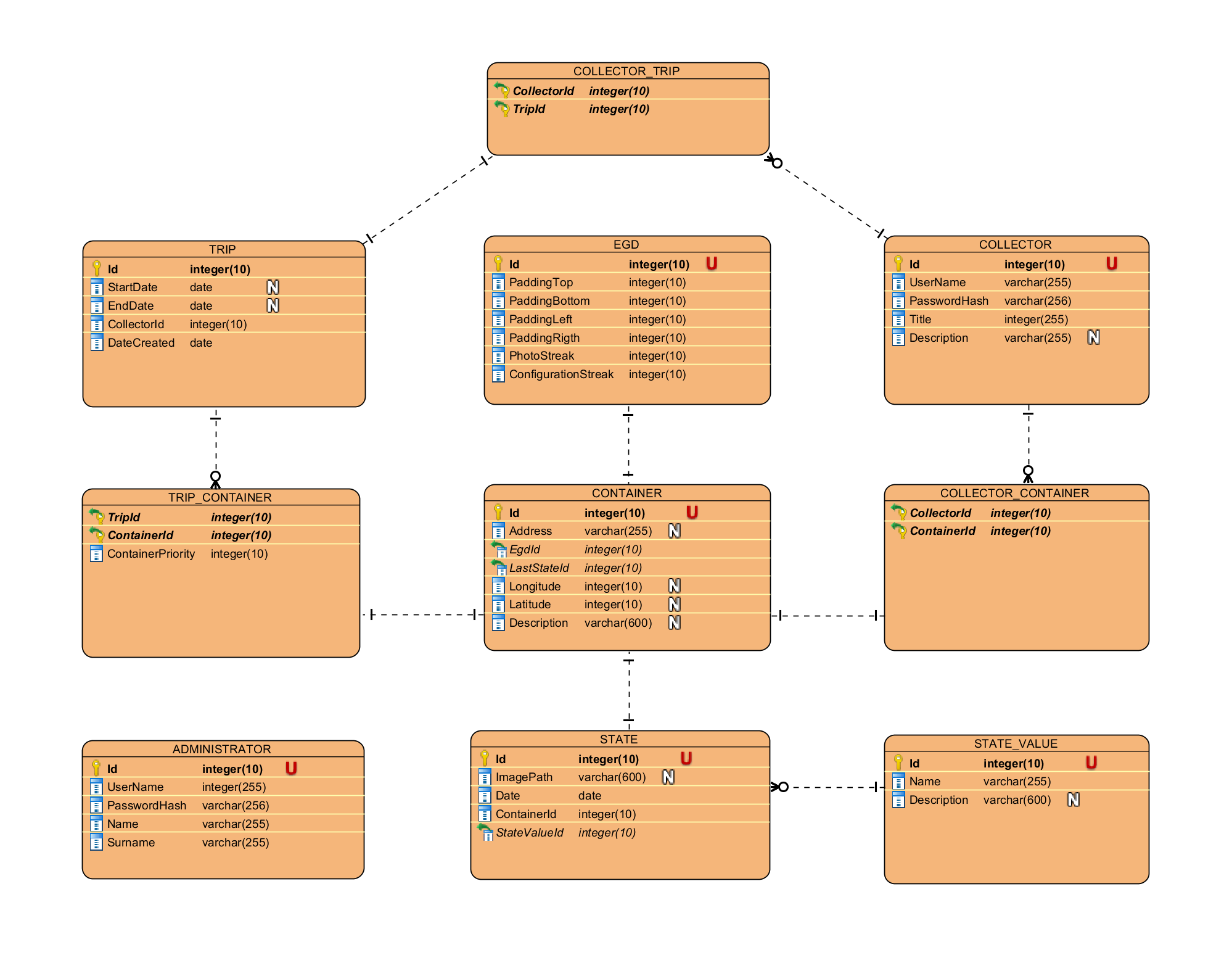
Abu šiuos klientus apjungia serveris, per kurį vyksta komunikacija su duomenų baze. Klientai neturi tarpusavio ryšio, jie atskirai komunikuoja su serveriu, kuris duomenis saugo duomenų bazėje, tačiau abu klientai per serverį gali pasiekti duomenis duomenų bazėje, taip dalintis informacija tarpusavyje.



pav. 6 Sistemos struktūra

### Duomenų bazė

Duomenis saugoti pasirinkta reliacinė duomenų bazę, kurios schema pateikta 7 paveikslėlyje.



pav. 7 Duomenų bazės schema

**ADMINISTRATOR** – lentelė, sauganti administratoriaus duomenis:

* Id – identifikacijos numeris,
* UserName – vartotojo vardas,
* PasswordHash – slaptažodžio maišos funkcijos rezultatas
* Name – vardas,
* Surname – pavardė.

**COLLECTOR** – lentelė, sauganti vežėjo duomenis:

* Id – identifikacijos numeris,
* UserName – vartotojo vardas,
* PasswordHash – slaptažodžio maišos funkcijos rezultatas,
* Title – vežėjo pavadinimas,
* Description - vežėjo aprašymas

**EGD** – lentelė, sauganti sistemoje priregistruotų EGD įrenginių konfigūraciją:

* Id – identifikacijos numeris,
* PaddingTop – EGD įrenginio daromų nuotraukų viršutinė korekcijos riba[[8]](#footnote-8),
* PaddingBottom - EGD įrenginio daromų nuotraukų apatinė korekcijos riba,
* PaddingLeft - EGD įrenginio daromų nuotraukų kairė korekcijos riba,
* PaddingRight - EGD įrenginio daromų nuotraukų dešinė korekcijos riba,
* PhotoStreak – EGD įrenginio nuotraukų darymo intervalo laikas,
* ConfigurationStreak – EGD įrenginio konfigūracijos atnaujinimo intervalo laikas.

**STATE** – lentelė, sauganti konteinerių būsenas:

* Id – būsenos identifikacijos numeris,
* Date – būsenos fiksavimo data,
* ContainerId – konteinerio identifikacijos numeris CONTAINER lentelėje,
* StateValueId – konteinerio būsenos identifikacijos numeris STATE\_VALUE lentelėje.

**STATE\_VALUE** – lentelė, sauganti galimas konteinerių būsenų reikšmes:

* Id – būsenos reikšmės identifikavimo numeris,
* Name – būsenos reikšmės pavadinimas,
* Description - būsenos reikšmės aprašymas

**CONTAINER** – lentelė, sauganti konteinerio duomenis:

* Id – identifikacijos numeris,
* Address – konteinerio adresas,
* EgdId – EGD įrenginio identifikacijos numeris EGD lentelėje,
* LastStateId – paskutinės užfiksuotos būsenos identifikacijos numeris STATE lentelėje,
* Longitude – konteinerio ilgumos koordinatės reikšmė,
* Latitude – konteinerio platumos koordinatės reikšmė,
* Description – konteinerio aprašymas.

**COLLECTOR\_CONTAINER** – lentelė, kuri sieja vežėją ir jam priklausančius konteinerius:

* CollectorId – vežėjo identifikacijos numeris COLLECTOR lentelėje,
* ContainerId – konteinerio identifikacijos numeris CONTAINER lentelėje.

**TRIP** – lentelė, sauganti kelionių informaciją:

* Id – kelionės identifikacijos numeris,
* StartDate – kelionės pradžios data,
* EndDate – kelionės pabaigos data,
* DateCreated – kelionės sukūrimo data.

**COLLECTOR\_TRIP** – lentelė, kuri sieja vežėja ir jam priklausančias keliones:

* CollectorId – vežėjo identifikacijos numeris COLLECTOR lentelėje.
* TripId – kelionės identifikacijos numeris TRIP lentelėje.

**TRIP\_CONTAINER** – lentelė, kuri sieja kelionę ir jai priskirtus konteinerius:

* ContainerId – konteinerio identifikacijos numeris CONTAINER lentelėje,
* TripId – kelionės identifikacijos numeris TRIP lentelėje,
* ContainerPriority – konteinerio išvežimo prioritetas šios kelionės metu.

### WEB klientas

Šioje informacinėje sistemoje WEB klientas realizuotas kaip Single-page application (žiūrėti terminų žodyne).

**Programavimo kalba:**

WEB kliento bazinis kodas parašytas **TypeScript** programavimo kalba.

Ši kalba pasirinkta dėl to, kad turi griežtus tipus, yra objektiškai orientuota bei kompiliuojama į **JavaScript**, kuris leidžia kurti interaktyvius interneto puslapius ir yra palaikomas modernių interneto naršyklių.

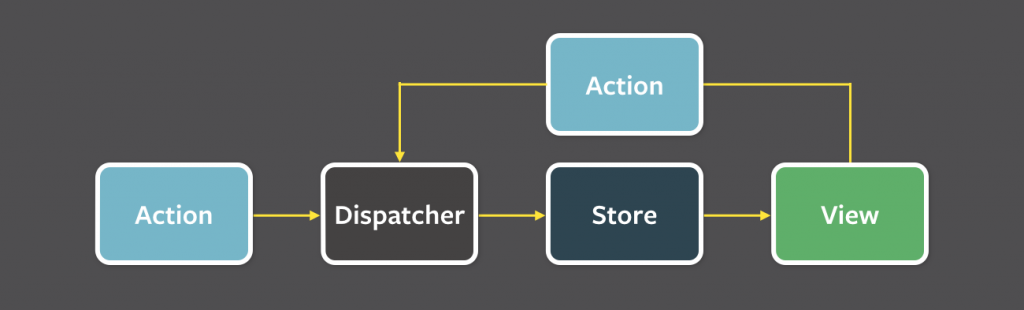
**Vaizdavimas**:

Vartotojo sąsajai realizuoti buvo pasitelkta JavaScript biblioteka **React**, kuri pasirinkta dėl galimybės, kurti vartotojo sąsaja komponentų principu. React komponentai yra dinamiški ir gali būti per panaudoti visoje sistemoje.

**Duomenų valdymas**:

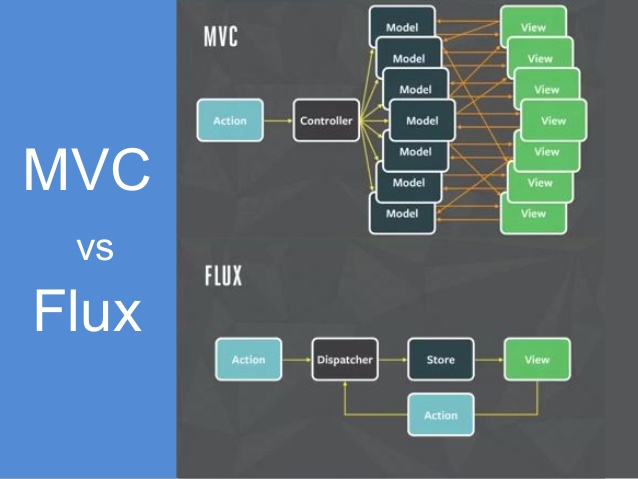
Duomenims vartotojo sąsajoje valdyti buvo panaudota **Flux architektūra**, kurios pagrindiniai elementai:

* Action – asinchroninis vartotojo atliktas veiksmas,
* Dispatcher – dispečeris, informuojantis duomenų logikos elementą (Store) apie įvykdytą vartotojo veiksmą,
* Store – duomenų logikos bei kaupimo elementas. Reaguoja į dispečerio (Dispatcher) perduotus vartotojo veiksmus (Actions) ir realizuoja gautą veiksmą atitinkantį duomenų apdorojimą. Informuoja vaizdavimo elementus (View) elementus apie duomenų pakitimą.
* View – vaizdavimo elementas, kuris yra informuojamas apie duomenų logikos elemente pakitusius duomenis. Jame yra iššaukiami veiksmai (Actions).



Pav. 8 Duomenų judėjimas Flux architektūroje

Flux architektūra pasirinkta dėl to, kad joje duomenys visuomet juda viena kryptimi, o tai palengvina duomenų apdorojimą ir vaizdavimą. Alternatyvios architektūros **MVC** atveju, vaizdavimo ir logikos elementus gali sieti dvikrypčiai ryšiai, kurių valdymas tampa sudėtingas, kai vaizdavimo elementai priklauso nuo kelių modelių (9 paveikslėlis).

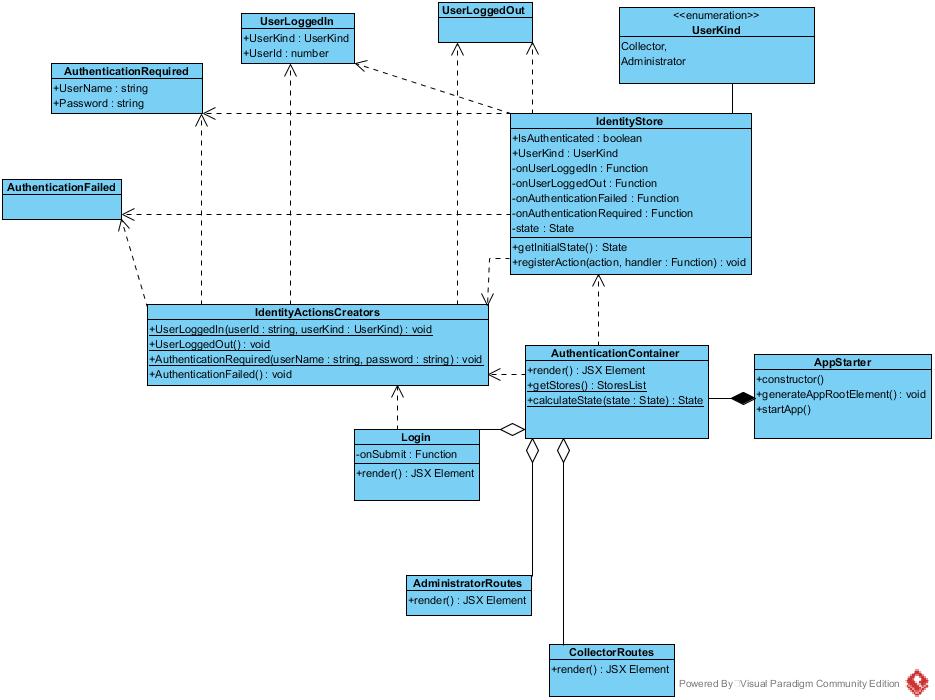


Pav. 9 Flux ir MVC architektūrų palyginimas

**Vartotojo sąsajos projekto struktūra:**

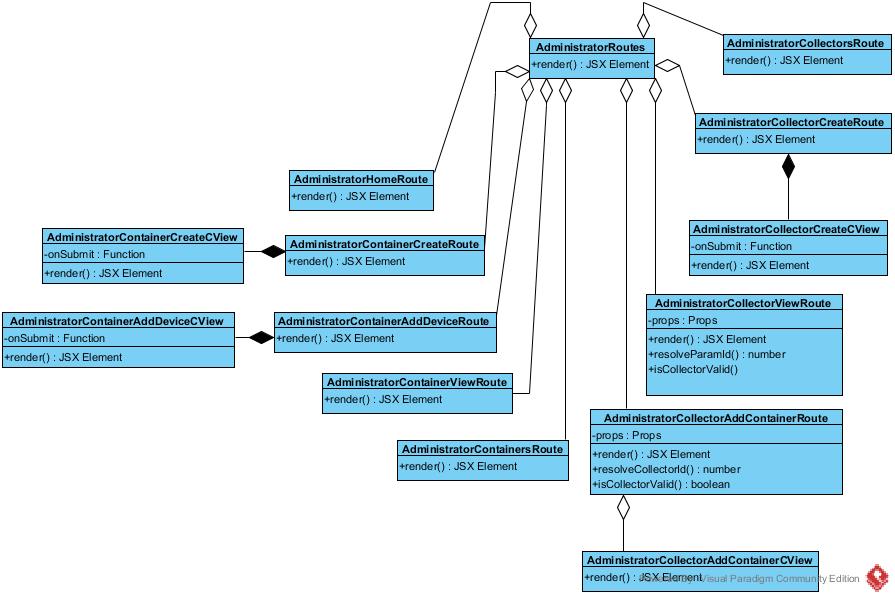
Projekto pradinė klasė – *AppStarter*, kuri paleidžia projektą. Į ją įeina klasė *AuthenticationContainer*, atsakinga už autentikacijos duomenų gavimo iš *IdentityStore* klasės. *IdentityStore* klasė užkrauna duomenis iš serverio arba lokalios naršyklės saugyklos ir saugo juos visu aplikacijos veikimo metu. *AuthenticationContainer* pagal esamą autentikacijos lygį parenką vieną iš vartotojų aplinkų:

*Login*, jeigu vartotojas neprisijungęs, *AdministratorRoutes -* administratoriaus aplinkos puslapius turinti klasė, jeigu vartotojas prisijungęs ir jo rolė Administratorius, *CollectorRoutes –* vežėjo puslapius turinti klasė, jeigu vartotojas prisijungęs ir jo role Vežėjas.



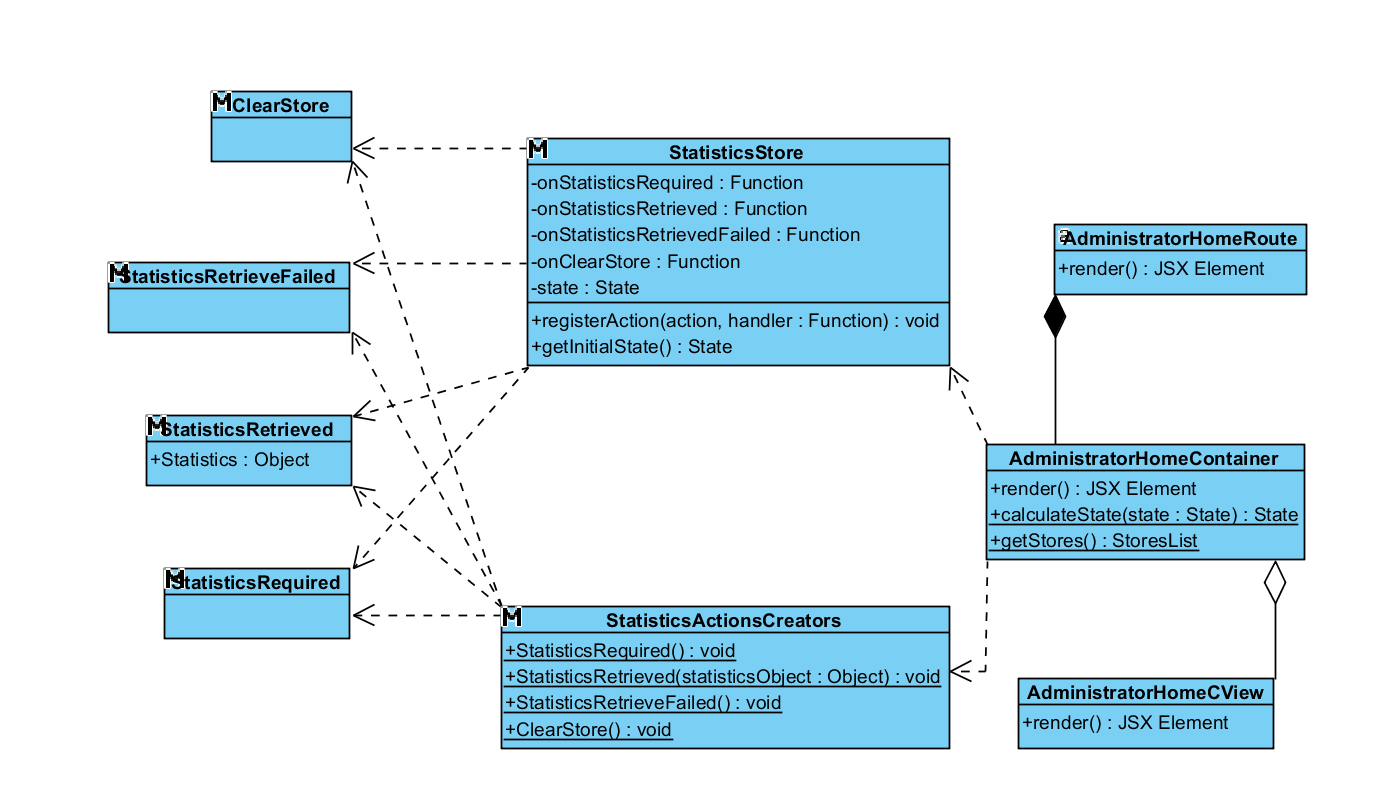
Pav. - Vartotojo sąsajos projekto struktūrą

Pagal tai, ar vartotojas autentikuotas, kokia jo rolė ir koks puslapis atidarytas, parenkama klasė, atvaizduojanti puslapį:



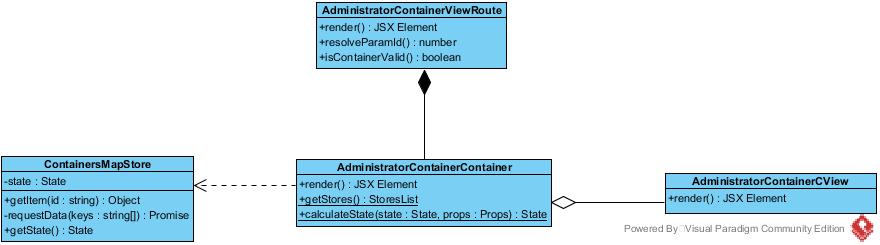
Pav. Administratoriaus puslapių klasės

Duomenų gavimo iš serverio nereikalaujantys puslapiai, gali tiesiogiai atvaizduoti vartotojo sąsajos klases (CView – ControllerView). Sudėtingesnės struktūros klasės buvo išskirtos į atskiras diagramas.



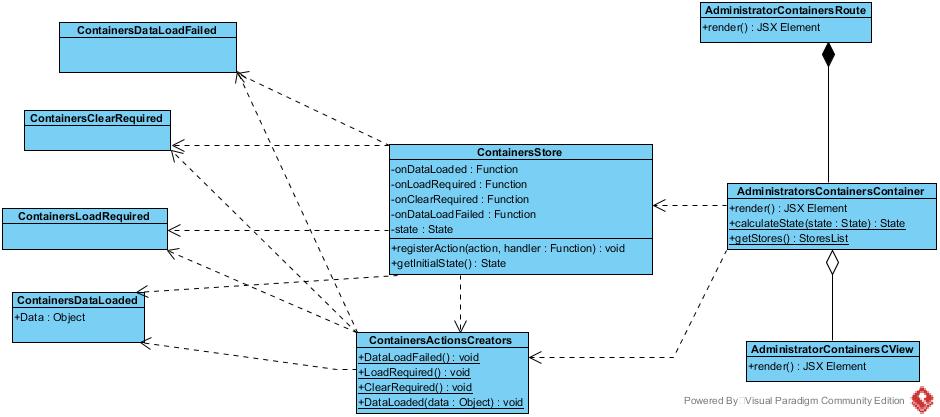
Pav. Administratoriaus pradinio puslapio diagrama

*AdministratorHomeContainer* užtikrina duomenų gavimą iš *StatisticsStore* klasės, kuri gauna sistemos statistinius duomenis iš serverio. Jeigu duomenys gauti sėkmingai, *AdministratorHomeCView* atvaizduoja sistemos statistinius duomenis.



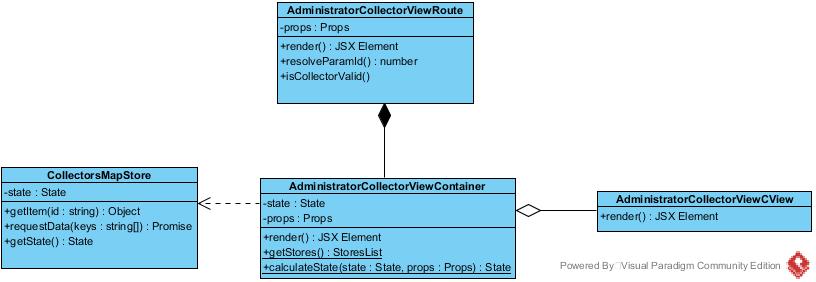
Pav. Vieno konteinerio puslapio atvaizdavimo klasių diagrama

*AdministratorContainerContainer ­*klasė užtikrina vieno konteinerio duomenų gavimą iš *ContainerMapStore* klasės, kuri užkrauna konteinerių duomenis iš serverio pagal pateiktą *id* ir savo *state* kintamajame kaupia visus aplikacijos veikimo metu užkrautų konteinerių duomenis.



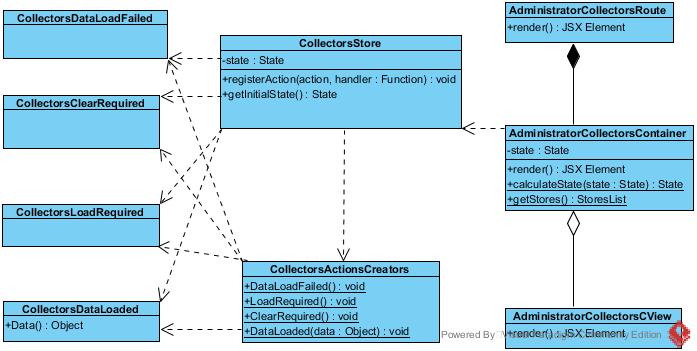
Pav. Konteinerių sąrašo puslapio klasių diagrama

*AdministratorContainersContainer* klasė atsakinga už duomenų gavimą iš *ContainersStore* klasės, kuri užkrauna konteinerių sąrašo duomenis iš serverio. Jeigu duomenys gauti sėkmingai klasė *AdministratorContainersCView* atvaizduoja konteinerių sąrašą.



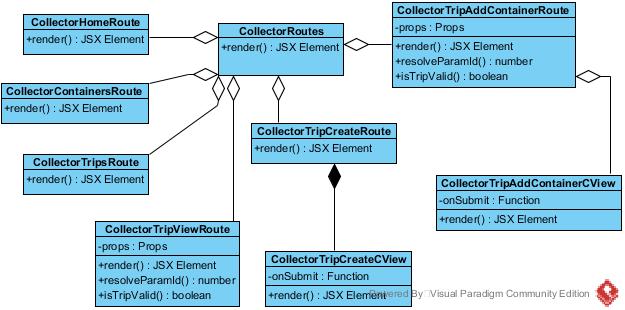
Pav. Vežėjo peržiūros puslapio klasių diagrama

*AdministratorCollectorViewContainer* užtikrina duomenų gavimą iš *CollectorsMapStore* klasės, kuri užkrauna vežėjo duomenis iš serverio pagal jo *id*. Jeigu duomenys gauti sėkmingai *AdministratorCollectorViewCView* atvaizduoja vežėjo duomenis.



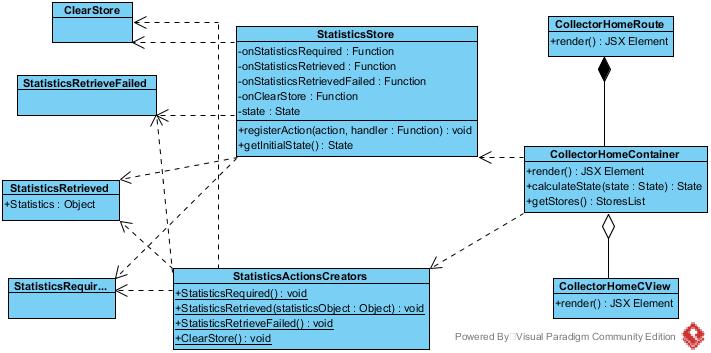
Pav. Vežėjų sąrašo puslapio klasių diagramos

*AdministratorCollectorsContainer* klasė užtikrina duomenų gavimą iš *CollectorsStore* klasės, kuri užkrauna konteinerių sąrašo duomenis iš serverio. Jeigu duomenys užkrauti sėkmingai, klasė *AdministratorCollectorsCView* atvaizduoja konteinerių sąrašą.



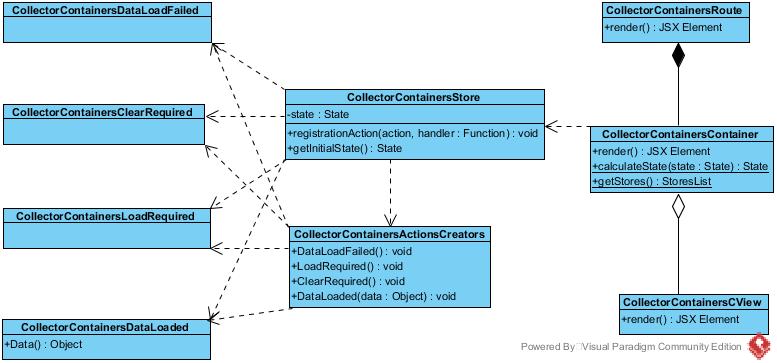
Pav. Vežėjo aplinkos puslapių klasių diagrama

Viena iš aplikacijoje realizuotų aplinkų – Vežėjo rolę turinčio vartotojo aplinka. Ją atvaizduoja *CollectorRoutes* klasė pagal puslapį, kurį šiuo metu atidaręs vartotojas.



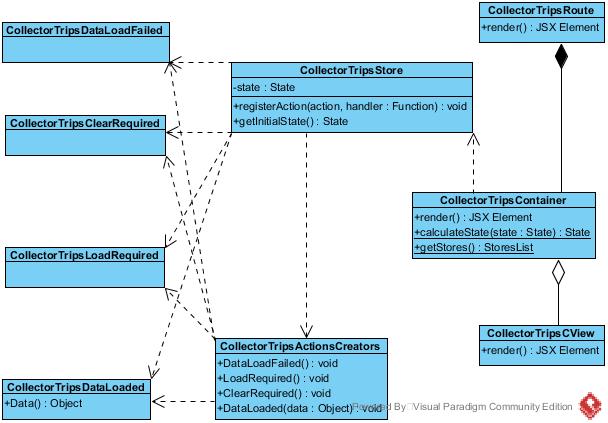
Pav. Vežėjo pradinio puslapio klasių diagrama

*CollectorHomeContainer* užtikrina duomenų gavimą iš *StatisticsStore* klasės, kuri užkrauna sistemos statistinius duomenis iš serverio. Jeigu duomenys gauti sėkmingai *CollectorHomeCView* atvaizduoja sistemos statistinius duomenis.



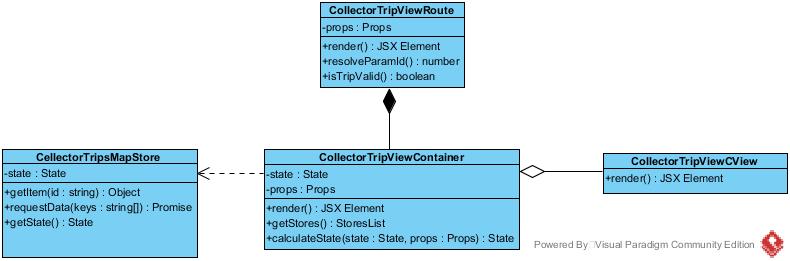
Pav. Vežėjui priskirtų konteinerių atvaizdavimo puslapio klasių diagrama

*CollectorContainersContainer* užtikrina duomenų gavimą iš *CollectorContainersStore* klasės, kuri užkrauna vežėjų sąrašo duomenis iš serverio. Jeigu duomenys gauti sėkmingai *CollectorContainersCView* atvaizduoja vežėjų sąrašą.



Pav. Vežėjo kelionių sąrašo puslapio klasių diagrama

*CollectorTripsContainer* užtikrina duomenų gavimą iš *CollectorTripsStore* klasės, kuri užkrauna vežėjo kelionių sąrašo duomenis iš serverio. Jeigu duomenys gauti sėkmingai *CollectorTripsCView* atvaizduoja vežėjo kelionių sąrašą.



Pav. Vežėjo kelionės puslapio klasių diagrama

*CollectorTripViewContainer* užtikrina duomenų gavimą iš *CollectorTripsMapStore* klasės, kuri užkrauna vežėjo kelionės duomenis iš serverio pagal jos *id*. Jeigu duomenys gauti sėkmingai *CollectorTripViewCView* atvaizduoja vežėjo kelionės duomenis.

### Serveris

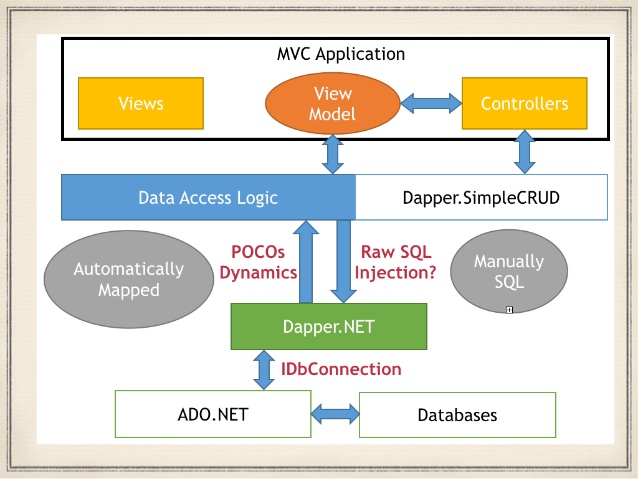
Šioje informacinėje sistemoje buvo naudojama .Net Core plėtinio aplinka.

**Programavimo kalba:**

Serverio kodas parašytas C# programavimo kalba. Ji buvo pasirinkta, nes ji yra objektiškai orientuota, turinti griežtus tipus programavimo kalba. Taip pat ji grąžina duomenis iš duomenų bazės JSON formatu.

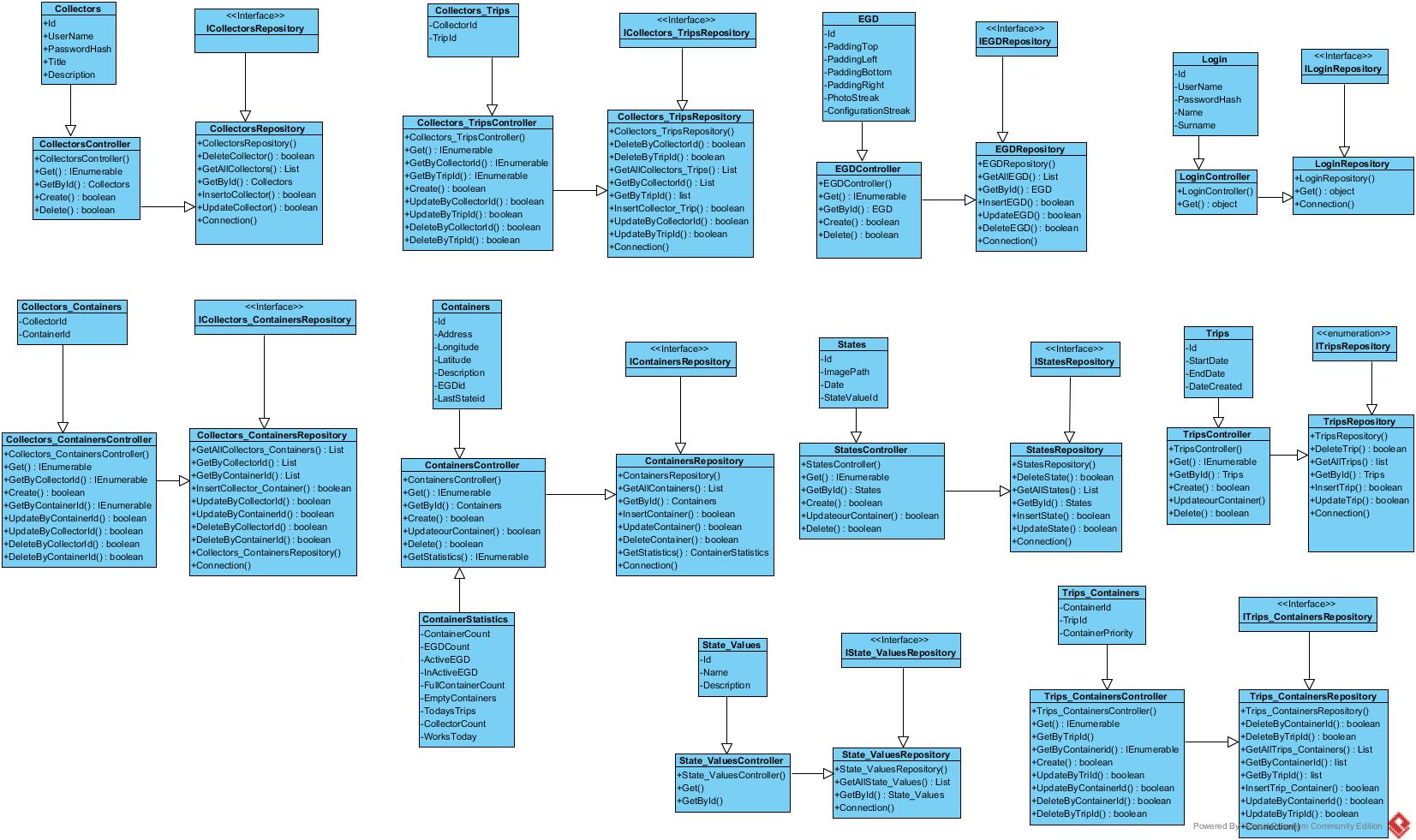
**Duomenų valdymas:**

Duomenys buvo valdomi Dapper ORM. Tai objektų vaidmenų simuliavimo būdas C# programavimo kalbai. Tai yra vienas iš greičiausių būdų valdyti duomenis tarp duomenų bazės ir programavimo kalbos. Jis veikia gana paprastai. Reikia inicializuoti prisijungimą prie duomenų bazės, įvykdyti norimą užklausą, perduoti norimą užklausą inicializavimui.



pav. Dapper ORM veikimo diagrama

### Serverio klasių diagrama

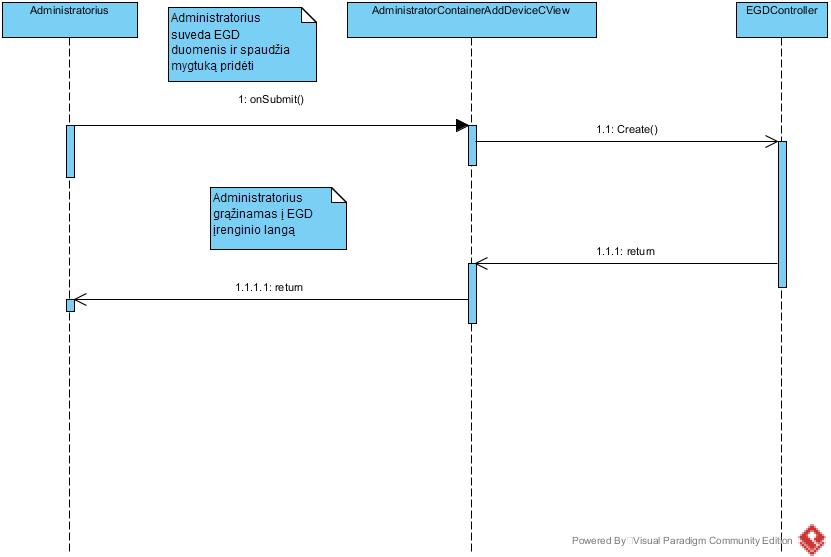


pav. Serverio klasių diagrama

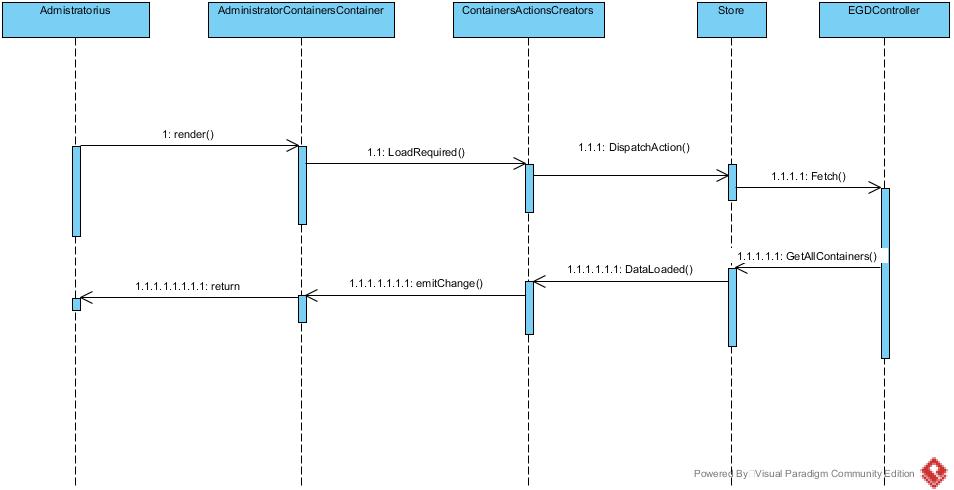
Kiekvienai duomenų bazės lentelei buvo kuriama klasė jos atributams aprašyti. Kiekviena klasė turi valdiklį(Controller), kuris yra skirtas valdyti keliams(routes), šitame valdiklyje aprašomi visi metodai naudojami tai klasei. Kiekvienas valdiklis kreipiasi į repozitoriją, kuri bendrauja su duomenų baze. Repozitorije aprašomos SQL užklausos , gaunami, įrašomi, keičiami duomenys. Kiekviena repozitorija turi pagrindinius metodus įrašo kūrimas, redagavimas, visų įrašų gavimas,pasirinkto įrašo gavimas, įrašo ištrynimas.

### Sistemos elgsena

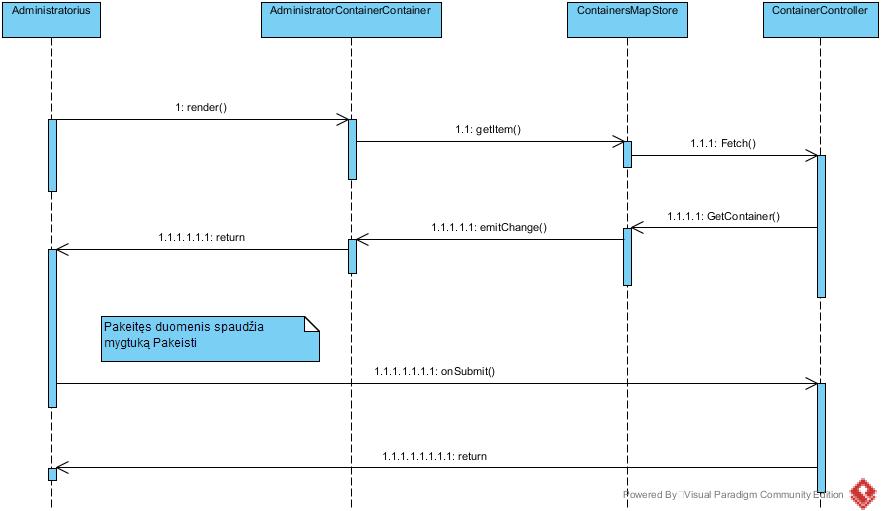
Sistemos elgseną pavaizdavome sekų diagramomis, kurios atitinka panaudos atvejus.



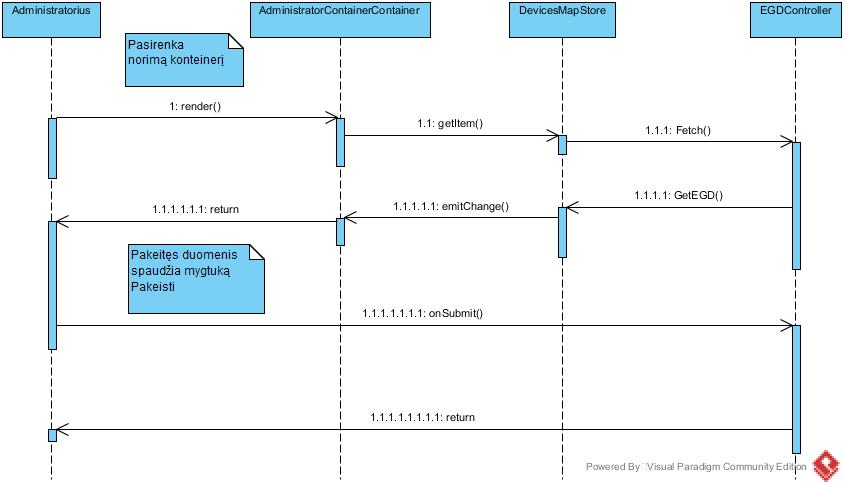
pav. Prideda EGD sekų diagrama



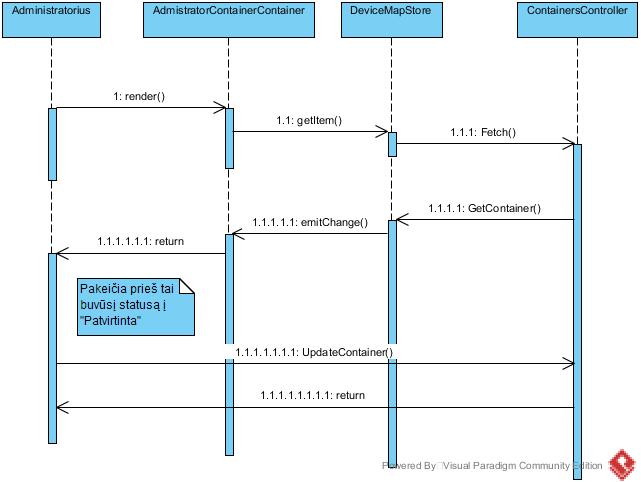
pav. Konteinerių statusų peržiūros sekų diagrama



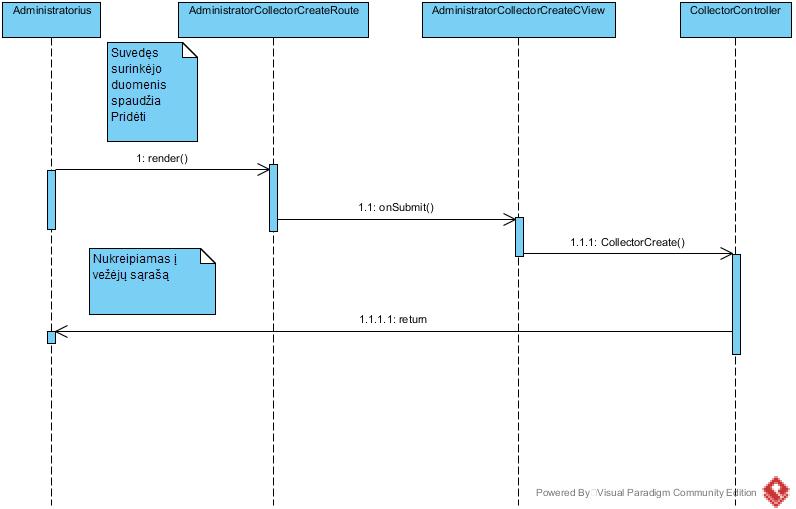
pav. Konteinerio konfigūracijos sekų diagrama



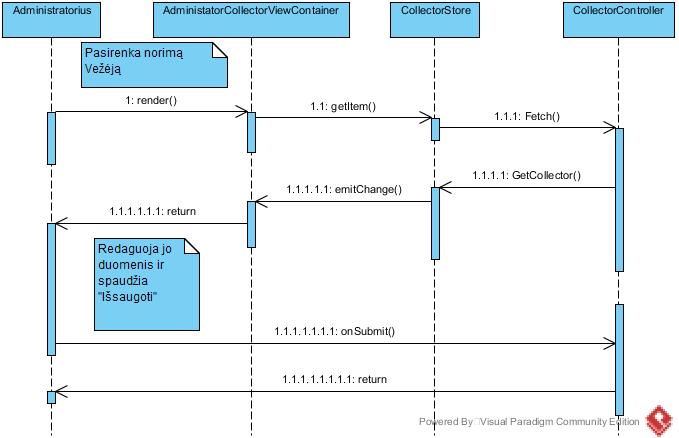
pav. EGD konfigūracijos sekų diagrama



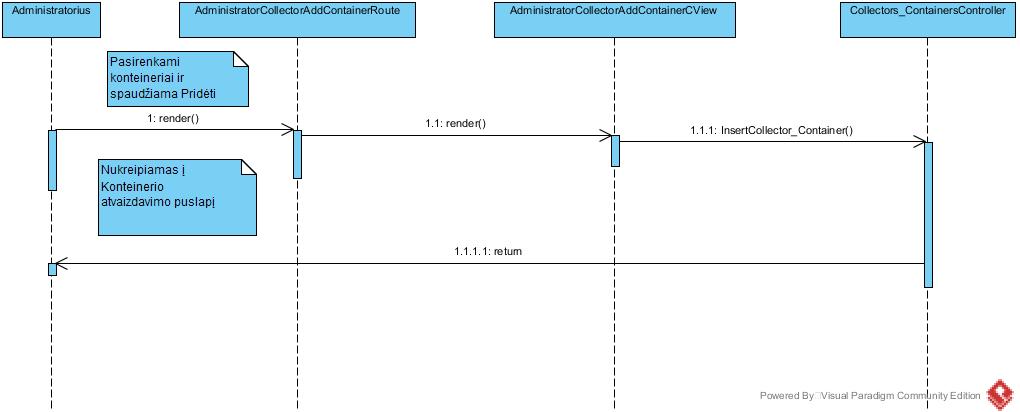
pav. Konteinerio statuso patvirtinimo sekų diagrama



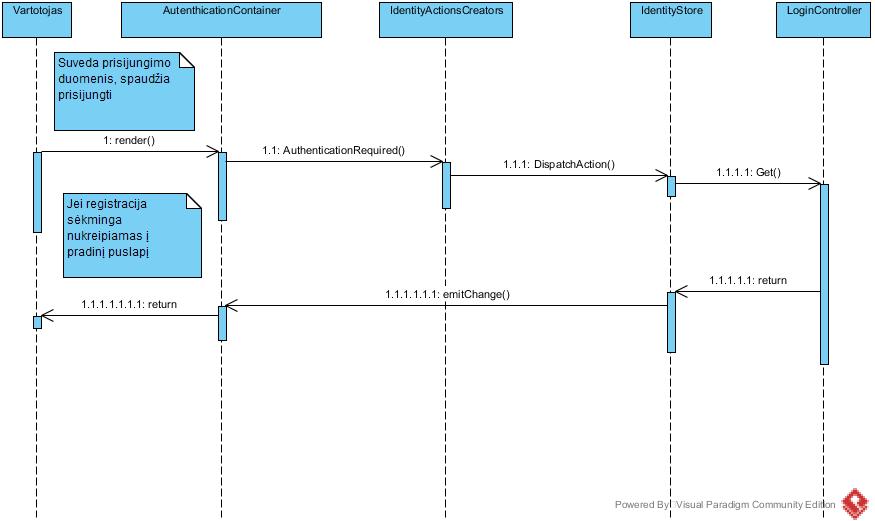
pav. Šiukšlių vežėjų registracijos sekų diagrama



pav. Šiukšlių vežėjo redagavimo sekų diagrama



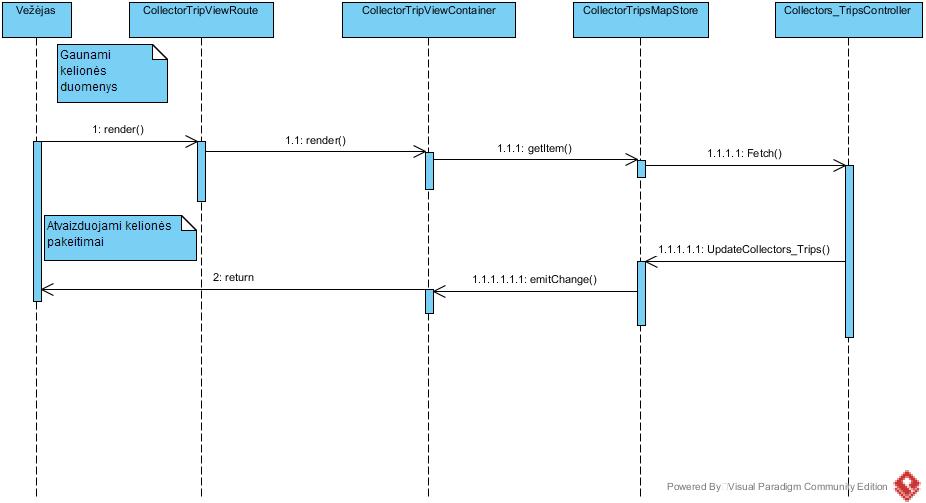
pav. Vežėjų konteinerių sąrašo sudarymo sekų diagrama



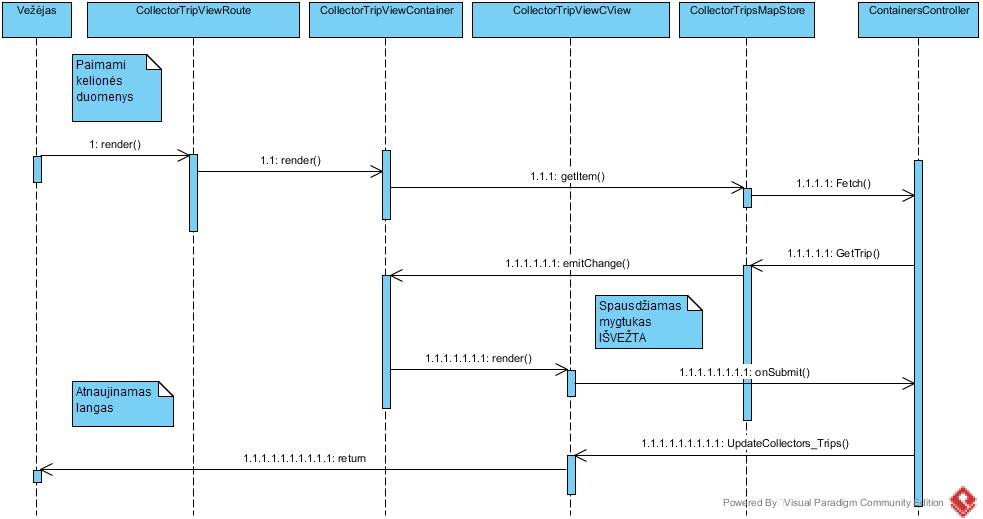
pav. Vartotojo prisijungimo sekų diagrama



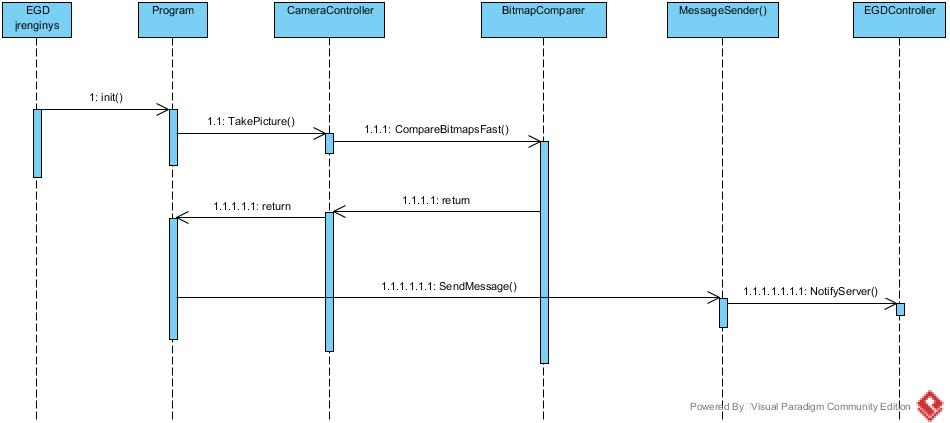
pav. Priskirtų šiukšlių vežėjų konteinerių sąrašo atvaizdavimo sekų diagrama



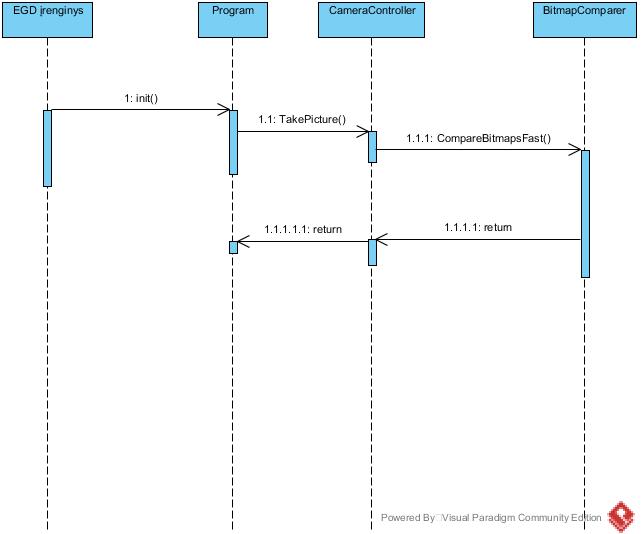
pav. Išvežimo maršruto sudarymo sekų diagrama



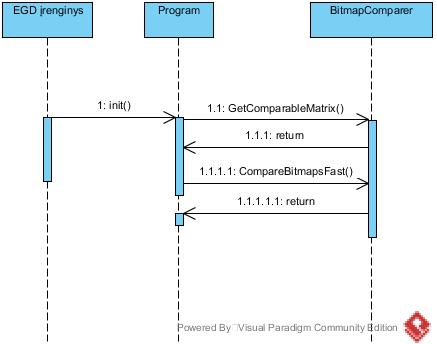
pav. Konteinerio statuso keitimo į išvežta sekų diagrama



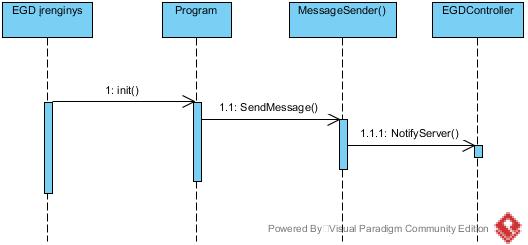
pav. EGD būsenos tikrinimo sekų diagrama



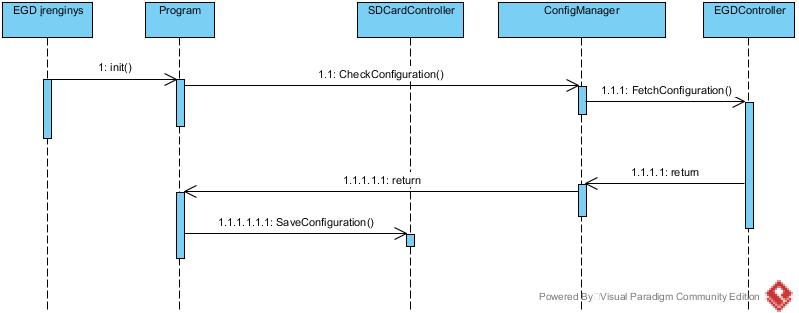
pav. Konteinerio nuotraukos darymo sekų diagrama



pav. Nuotraukos lyginimo su šablonu sekų diagrama



pav. Pranešimo apie konteinerio būseną sekų diagrama



pav. Konfigūracijos tikrinimo sekų diagrama

### Serverio API keliai

#### CollectorsController

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kelias** | **HTTP metodas** | **Metodo pavadinimas** | **Grąžinamas tipas** | **Atliekama funkcija** |
| Api/Collectors | HttpGet | Get() | IEnumerable<Collectors> | Grąžina visų surinkėjų duomenis |
| Api/Collectors/{id} | HttpGet | GetById() | Collectors | Grąžina pasirinkta surinkėją |
| Api/Collectors | HttpPost | Create() | Boolean | Sukuria surinkėją |
| Api/Collectors | HttpPut | UpdateourContainer() | Boolean | Pakeičia esamus duomenis apie surinkėją |
| Api/Collectors/{id} | HttpDelete | Delete() | Boolean | Ištrina pasirinktą surinkėją |

#### Lentelė CollectorsController metodai bei jų keliai

#### Collectors\_ContainersController

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kelias** | **HTTP metodas** | **Metodo pavadinimas** | **Grąžinamas tipas** | **Atliekama funkcija** |
| Api/Collectors\_Containers | HttpGet | Get() | IEnumerable<Collectors\_Containers> | Grąžina visus duomenis apie vežėjų priskirtus konteinerius |
| Api/ Collectors\_Containers/cc/collector /{id} | HttpGet | GetByCollectorId() | IEnumerable<Collectors\_Containers> | Grąžina pasirinkto surinkėjo konteinerius |
| Api/ Collectors\_Containers/cc/container /{id} | HttpGet | GetByContainerId() | IEnumerable<Collectors\_Containers> | Grąžina pasirinkto konteinerio surinkėjus |
| Api/Collectors\_Containers | HttpPost | Create() | Boolean | Sukuria surinkėjo ir konteinerio įrašą |
| Api/Collectors\_Containers/cc/container | HttpPut | UpdateByCollectorId() | Boolean | Pakeičia esamus duomenis palei surinkėjo ID |
| Api/Collectors\_Containers/cc/collector | HttpPut | UpdateByContainerId() | Boolean | Pakeičia esamus duomenis palei konteinerio ID |
| Api/Collectors\_Containers/cc/collector /{id} | HttpDelete | DeleteByCollectorId() | Boolean | Ištrina pasirinktą įrašą pagal surinkėjo ID |
| Api/Collectors\_Containers/cc/container /{id} | HttpDelete | DeleteByContainerId() | Boolean | Ištrina pasirinktą įrašą pagal konteinerio ID |

#### Lentelė Collectors\_ContainersController metodai ir jų keliai

#### Collectors\_TripsController

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kelias** | **HTTP metodas** | **Metodo pavadinimas** | **Grąžinamas tipas** | **Atliekama funkcija** |
| Api/Collectors\_Trips | HttpGet | Get() | IEnumerable<Collectors\_Trips> | Grąžina visus duomenis apie vežėjų priskirtas keliones |
| Api/ Collectors\_Trips /ct/collector/{id} | HttpGet | GetByCollectorId() | IEnumerable<Collectors\_Trips> | Grąžina pasirinkto surinkėjo keliones |
| Api/ Collectors\_Trips /ct/trip/{id} | HttpGet | GetByTripId() | IEnumerable<Collectors\_Trips> | Grąžina pasirinktos  kelionės surinkėjus |
| Api/Collectors\_Trips | HttpPost | Create() | Boolean | Sukuria surinkėjo ir kelionės įrašą |
| Api/Collectors\_Containers/ct/collector | HttpPut | UpdateByCollectorId() | Boolean | Pakeičia esamus duomenis palei surinkėjo ID |
| Api/Collectors\_Containers/ct/trip | HttpPut | UpdateByTripId() | Boolean | Pakeičia esamus duomenis palei kelionės ID |
| Api/Collectors\_Containers/ct/collector /{id} | HttpDelete | DeleteByCollectorId() | Boolean | Ištrina pasirinktą įrašą pagal surinkėjo ID |
| Api/Collectors\_Containers/ct/trip/{id} | HttpDelete | DeleteByContainerId() | Boolean | Ištrina pasirinktą įrašą pagal kelionės ID |

#### Lentelė Collectors\_TripsController metodai ir jų keliai

#### ContainersController

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kelias** | **HTTP metodas** | **Metodo pavadinimas** | **Grąžinamas tipas** | **Atliekama funkcija** |
| Api/Containers | HttpGet | Get() | IEnumerable<Containers> | Grąžina visų konteinerių duomenis |
| Api/Containers/{id} | HttpGet | GetById() | Containers | Grąžina pasirinkta konteinerį |
| Api/Containers | HttpPost | Create() | Boolean | Sukuria konteinerį |
| Api/Containers | HttpPut | UpdateourContainer() | Boolean | Pakeičia esamus duomenis apie konteinerį |
| Api/Containers/{id} | HttpDelete | Delete() | Boolean | Ištrina pasirinktą konteinerį |
| Api/Containers/stats | HttpPost | GetStatistics() | IEnumerable<Containers> | Gauna statistika |

Lentelė ContainersController metodai ir jų keliai

#### EGDController

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kelias** | **HTTP metodas** | **Metodo pavadinimas** | **Grąžinamas tipas** | **Atliekama funkcija** |
| Api/EGD | HttpGet | Get() | IEnumerable<EGD> | Grąžina visų EGD duomenis |
| Api/EGD/{id} | HttpGet | GetById() | EGD | Grąžina pasirinkta EGD |
| Api/EGD | HttpPost | Create() | Boolean | Sukuria EGD |
| Api/EGD | HttpPut | UpdateEGD() | Boolean | Pakeičia esamus duomenis apie EGD |
| Api/EGD/{id} | HttpDelete | Delete() | Boolean | Ištrina pasirinktą EGD |

Lentelė EGDController metodai ir jų keliai

#### LoginController

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kelias** | **HTTP metodas** | **Metodo pavadinimas** | **Grąžinamas tipas** | **Atliekama funkcija** |
| Api/Login/{UserName} | HttpPost | Get() | Object | Grąžina true jei toks naudotojas yra, false jei nėra |

Lentelė LoginController metodai ir jų keliai

#### State\_ValuesController

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kelias** | **HTTP metodas** | **Metodo pavadinimas** | **Grąžinamas tipas** | **Atliekama funkcija** |
| Api/State\_Values | HttpGet | Get() | IEnumerable<State\_Values> | Grąžina visų būsenų duomenis |
| Api/State\_Values/{id} | HttpGet | GetById() | State\_Values | Grąžina pasirinktą būseną |

Lentelė 26 State\_ValuesController metodai ir jų keliai

#### StatesController

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kelias** | **HTTP metodas** | **Metodo pavadinimas** | **Grąžinamas tipas** | **Atliekama funkcija** |
| Api/States | HttpGet | Get() | IEnumerable<States> | Grąžina visas būsenas |
| Api/States/{id} | HttpGet | GetById() | States | Grąžina pasirinktą būseną |
| Api/States | HttpPost | Create() | Boolean | Sukuria būseną |
| Api/States | HttpPut | UpdateStates() | Boolean | Pakeičia esamus duomenis apie būseną |
| Api/States/{id} | HttpDelete | Delete() | Boolean | Ištrina pasirinktą būseną |

Lentelė StatesController metodai ir jų keliai

#### TripsController

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kelias** | **HTTP metodas** | **Metodo pavadinimas** | **Grąžinamas tipas** | **Atliekama funkcija** |
| Api/Trips | HttpGet | Get() | IEnumerable<Trips> | Grąžina visas keliones |
| Api/Trips/{id} | HttpGet | GetById() | Trips | Grąžina pasirinktą kelionę |
| Api/Trips | HttpPost | Create() | Boolean | Sukuria kelionę |
| Api/Trips | HttpPut | UpdateTrip() | Boolean | Pakeičia esamus duomenis apie kelionę |
| Api/Trips/{id} | HttpDelete | Delete() | Boolean | Ištrina pasirinktą kelionę |

Lentelė TripsController metodai ir jų keliai

#### Trips\_ContainersController

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kelias** | **HTTP metodas** | **Metodo pavadinimas** | **Grąžinamas tipas** | **Atliekama funkcija** |
| Api/Trips\_Container | HttpGet | Get() | IEnumerable<Trips\_Container> | Grąžina visus duomenis apie kelionių priskirtus konteinerius |
| Api/ Trips\_Container/tc/trip/{id} | HttpGet | GetByTripId() | IEnumerable<Trips\_Container> | Grąžina pasirinktos kelionės konteinerius |
| Api/ Trips\_Container/tc/container/{id} | HttpGet | GetByContainerId() | IEnumerable<Trips\_Container> | Grąžina pasirinkto  Konteinerio keliones |
| Api/Trips\_Container | HttpPost | Create() | Boolean | Sukuria konteinerio ir kelionės įrašą |
| Api/Trips\_Container/tc/container | HttpPut | UpdateByContainerId() | Boolean | Pakeičia esamus duomenis palei konteinerio ID |
| Api/Trips\_Container/tc/trip | HttpPut | UpdateByTripId() | Boolean | Pakeičia esamus duomenis palei kelionės ID |
| Api/Trips\_Containers/tc/container /{id} | HttpDelete | DeleteByContainerId() | Boolean | Ištrina pasirinktą įrašą pagal konteinerio ID |
| Api/Trips\_Containers/tc/trip/{id} | HttpDelete | DeleteByTripId() | Boolean | Ištrina pasirinktą įrašą pagal kelionės ID |

Lentelė 29 Trips\_ContainersController metodai ir jų keliai

### Diegimo projektas

Sistema susideda iš 6 pagrindinių komponentų:

* Serverio,
* Duomenų bazės serverio,
* Įrenginio nustatančio konteinerio pilnumo lygį,
* Vartotojo su įdiegta interneto naršykle

Serveris su įrenginiu nustatančiu konteinerio pilnumo lygį bendrauja naudodamas HTTP protokolą. Su duomenų baze komunikuoja naudojant TCP/IP protokolą.

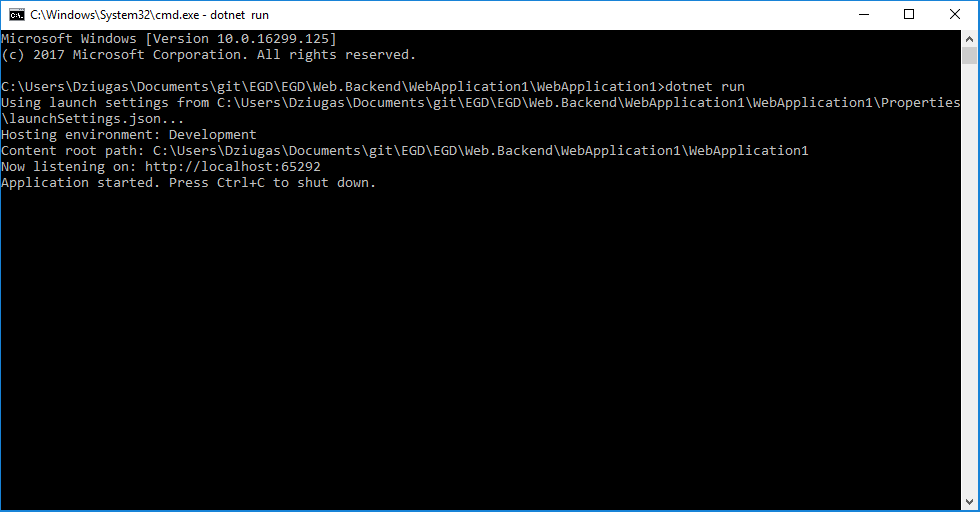
# Realizacija ir testavimas

# Priedai

## Serverio diegimo vadovas

Norint įdiegti serverio PĮ reikia atlikti tam tikrus žingsnius:

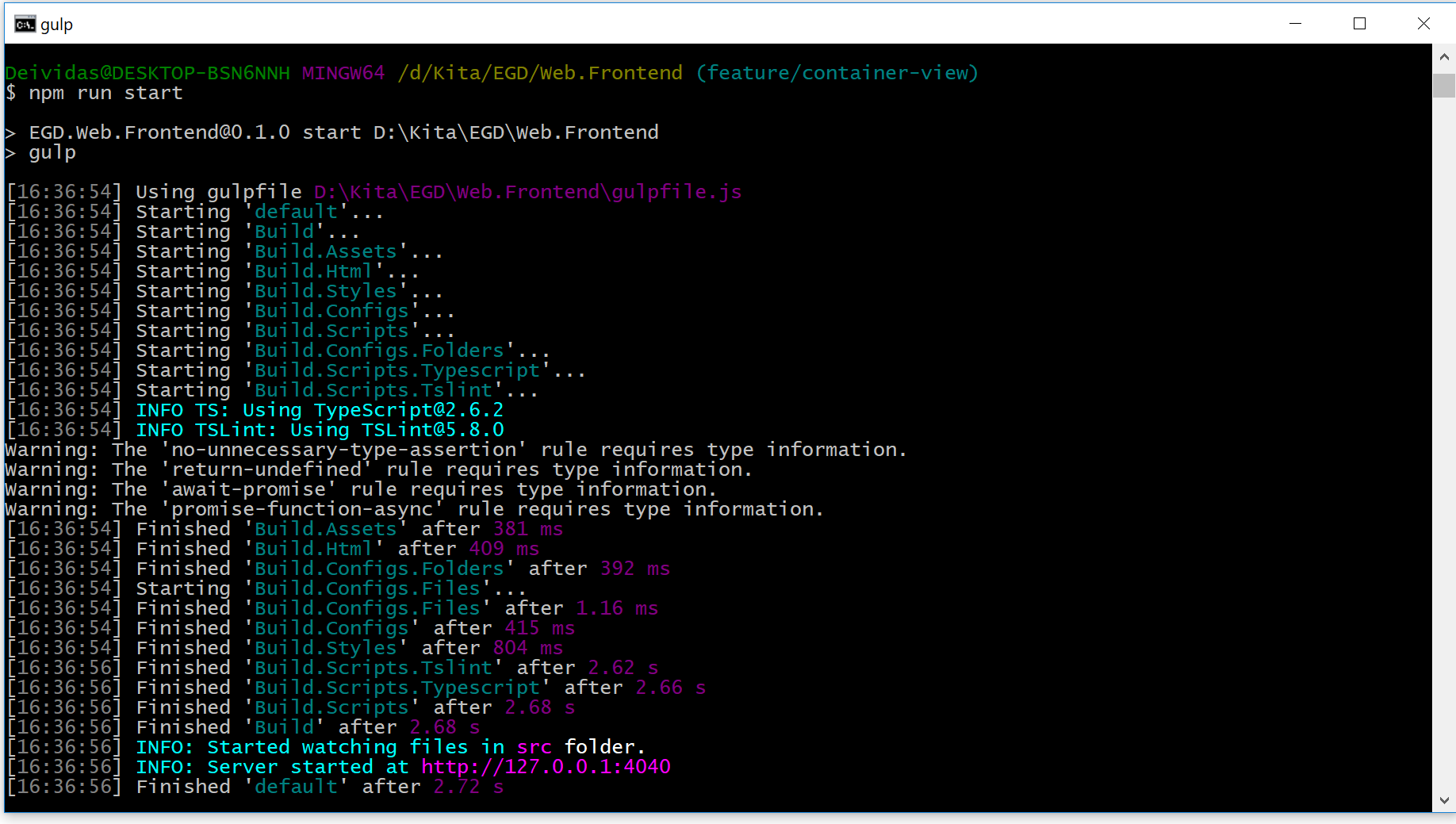
1. Parsisiųsti Github repozitoriją. (<https://github.com/martynasraila/EGD>)
2. Įsidiegti į kompiuterį SQL Server Management Studio 2017.
3. Pasijungti SQL Server Management Studio, susikurti duombazę ir nurodyti parsisiųstoje direktorijoje folderį Web.BackEnd/EGDDB, tada bus įdiegta projekto duombazė jūsų serveryje.
4. Įsidiegti naujausią .NET Core versiją.
5. Pasijungti Windows aplinkoje/Linux aplinkoje Command Line langą ir su cd pagalba nueiti į EGD/Web.Backend/WebApplication1.
6. Nuėjus į šią direktoriją naudoti komandą „dotnet run“, tada bus paleidžiamas serverio programinis kodas, bus atvaizduojama kokiu adresu serveris yra pasiekiamas.
7. Norint patikrinti serverio veikimą reikia atsidaryti naršyklę ir suvesti serverio nuorodą į naršyklės puslapio lauką. „https:/serverioadresas/api/trips“ ir jei viskas sudiegta teisingai bus atvaizduojami kelionių duomenys iš duomenų bazės.



pav. 41 Paleidus serverį su komanda „dotnet run“

## WEB kliento PĮ diegimo vadovas

1. Norint paleisti WEB kliento programinę įrangą, kompiuteryje turi būti įdiegta Node.js v8.6.0 arba naujesnės versijos programinė įranga, ją galima rasti <https://nodejs.org/en/> (nuoroda paskutinį kartą tikrinta 2017-12-17).
2. Parsisiųskite EGD projekto repozitoriją. Ją galima rasti <https://github.com/martynasraila/EGD> (nuoroda paskutinį kartą tikrinta 2017-12-17).
3. CLI aplinkoje atidarykite repozitorijos aplanką *Web.Frontend,* jame įveskite komandas:
   * *npm run setup*
   * *npm run build*
   * *npm run start*
4. CLI lange pamatysite lokalią nuorodą į WEB kliento puslapį:



Pav. 42 CLI aplinkos langas paleidus *npm run start*

1. Detalesnis sprendimo aprašymas: https://www.smartbin.com/solutions/iot-level-sensors/ [↑](#footnote-ref-1)
2. Detalesnis sprendimo aprašymas: http://www.bine.world/ [↑](#footnote-ref-2)
3. Detalesnis sprendimo aprašymas: http://bigbelly.com/platform/#smart\_waste\_system [↑](#footnote-ref-3)
4. Sistemą naudojantys klientai privalo naudoti specialus konkrečios įmonės tiekiamus konteinerius / šiukšlines. [↑](#footnote-ref-4)
5. Sistemoje naudojamos konteineriai / šiukšlinės automatiškai suspaudžai (supresuoja) turinį. [↑](#footnote-ref-5)
6. [↑](#footnote-ref-6)
7. Moderni naršyklė – naujausias žiniatinklio technologijas ir standartus palaikanti naršyklė. [↑](#footnote-ref-7)
8. EGD įrenginio korekcijos riba – įrenginio daromos nuotraukos apkirpimo tikslinei analizei pikselių kiekis. [↑](#footnote-ref-8)