

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INformatikos fakultetas

**Komanda: ARmATŪRA**

**Projektas:Optimalus Maršruto Radimas**

Moduliai: Semestro projektas, Programų sistemų inžinerija  
Projekto dokumentacija

**Laboratorinių darbų dėstytojas**:  
lekt. A.Ušianov

**Vertintojas**

Doc. Tomas Blažauskas

**Studentai:**

Tadas Meškauskas IFF-6/3

Remigijus Žebrauskas IFF-6/11

Deividas Patalauskas IFF-6/11

Kaunas, 2018

KAUNAS, 2018

Turinys

[Įvadas 3](#_Toc514240797)

[Techninė užduotis 3](#_Toc514240798)

[Sistemos kūrimo technologijos 4](#_Toc514240799)

[Sistemos kūrimo įrankiai 4](#_Toc514240800)

[Įrankių panaudojimas sistemos kūrimo procese 4](#_Toc514240801)

[Kūrimo proceso valdymo įrankis 4](#_Toc514240802)

[Saugykla 7](#_Toc514240803)

[Klaidų, darbų, idėjų sekimo įrankis 9](#_Toc514240804)

[Bendradarbiavimo įrankis 9](#_Toc514240805)

[Testavimo įrankiai/karkasai 10](#_Toc514240806)

[Dokumentacija naudotojui 11](#_Toc514240807)

[Rezultatų apibendrinimas ir išvados 16](#_Toc514240808)

[Literatūra 16](#_Toc514240809)

**Paveikslėlių sąrašas:**

[1 pav. 1 sprinto užduotys bei jų vertinimai 4](#_Toc515359681)

[2 pav. Antrojo sprinto apžvalga 5](#_Toc515359682)

[3 pav. Trečiojo sprinto apžvalga 5](#_Toc515359683)

[4 pav. Ketvirtojo sprinto įvykdytos užduotys 6](#_Toc515359684)

[5 pav. Mūsų projekto statistika(burn-down ir burn-up grafikai) 6](#_Toc515359685)

[6 pav. Master šaka 7](#_Toc515359686)

[7 pav. Tado šaka 7](#_Toc515359687)

[8 pav. Remigijaus šaka 8](#_Toc515359688)

[9 pav. Deivido šaka 8](#_Toc515359689)

[10 pav. Iššukių užklausos. 9](#_Toc515359690)

[11 pav. Suliejimo užklausos. 9](#_Toc515359691)

[12 pav. Mūsų komandos bendradarbiavimo įrankis 10](#_Toc515359692)

[13 pav. Atlikto testavimo rezultatai 10](#_Toc515359693)

[14 pav. pradinis puslapio langas 11](#_Toc515359694)

[15 pav. puslapio pagrindinis langas 12](#_Toc515359695)

[16 pav. žemėlapio puslapis 1 paveikslėlis 12](#_Toc515359696)

[17 pav. žemėlapio puslapis 2 paveikslėlis 13](#_Toc515359697)

[18 pav. optimalus maršruto paieškos pavyzdys, naudojant paiešką su grįžimu atgal. 14](#_Toc515359698)

[19 pav. optimalus maršruto paieškos pavyzdys, naudojant paiešką su važiavimu į paskutinį tašką 14](#_Toc515359699)

[20 pav. projekto komanda 15](#_Toc515359700)

[21 pav. trumpa naudojimo instrukcija 15](#_Toc515359701)

[22 pav. Komponento įrodymas 16](#_Toc515359702)

# Įvadas

Viskas prasidėjo nuo to, jog mūsų komandos nariai: Tadas, Remigijus ir Deividas vieną gūdų ir šaltą vasario mėnesio vakarą susėdome prie bendro stalo bendrabučio kambaryje ir bandėme nutarti, ką mes galėtumėm padaryti semestro projektui. Minčių buvo įvarių nuo paprasto google chrome addons(boto), kuris sektų prekių kainas internete bei jas perduotų vartotojui iki paprasto flash žaidimo. Tačiau mūsų gerbiamam Remigijui kilo idėja padaryti optimalus maršruto paieškos sistemą, nes pasidomėję Lietuvos rinka, tokio dalyko(analogo) mes neradome. Todėl ir nusprendėme tai daryti, nes norėjome sukurti kažką unikalaus, to ko nebūtų kiti studentai padarę. Lankydami programų sistemų inžinerijos teorines paskaitas, sužinojome daug būdų, kaip galime realizuoti savo projektą. Tada bendrai pasitarę, nusprendėme viską atlikti pagal Agile metodologiją, darydami dviejų savačių trukmės sprintus bei naudodami kliento-serverio architechtūrą. Agile metodologiją mes pasirinkome todėl, nes mūsų nuomone, tai yra geriausias būdas užsakovui parodyti projekto progresą. Tarkime darome dviejų savaičių vadinamąjį sprintą, po jo parodome tarpinius rezultatus užsakovui, išklausome jo pastabų, naujų reikalavimų ir tada imame atlikti antrą sprintą, po jo vėl seka susitikimas su užsakovu ir per sprintą pasiektų rezultatų aptarimas. Ir tas apytakos ratas vyksta tol, kol yra pabaigiamas projektas arba užsakovas nusprendžia nutraukti finansavimą projektui ir taip jį padaryti neatliktu iki galo. Kalbant apie architechtūrinį stilių, kurį mes pasirinkome, jis yra naudingas tuo, jog mūsų atveju vartotojo kompiuteris neatlieka esminių skaičiavimo operacijų, t.y. jos yra siunčiamos į puslapio sąvininko(hostingo) serverį ir ten atliekamos ir atlikus jas, grąžinamas rezultatas, kuris pasirodo vartotojo kompiuterio ekrane. Tokiu būdu yra sutaupomi vartotojo resursai ir operacijos atliekamos grečiau. Galime užtikrinti, jog puslapis, tiek naujesnio, tiek senesnio modelio vartotojo kompiuteryje veiks vienodu pajėgumu.

# Techninė užduotis

Mums užsakovas davė užduotį sukurti programinę įrangą, kuri rastų optimalų maršrutą įvedus arba pažymėjus žemėlapyje n taškų(n nedaugiau kaip 10).

Funkciniai reikalavimai:

1. Programa stengtųsi rasti tinkamą, palankiausią, geriausią maršrutą.
2. Programos grafikos sąsaja su vartotaju turi būti aiški ir paprasta.
3. Grafikos sąsajoje bus keliata mygtukų, kur bus galima atsiversti pradinį, žemėlapį ir apie mus puslapius.
4. Žemėlapyje, grafikos sąsoje, bus galima įvest adresus arba pasirinkti žemėlapyje.
5. Bus galima nustatyti mašruto galutinį tašką.

Nefunkciniai reikalavimai:

1. Nedideliem ir dideliem tikslų kiekiams naudosime skirtingus algoritmus.
2. Bus atskiri laukai įvesti adresus, nustatyti maršruto pradžią bei pabaigą.
3. Žemėlapiui sukurti bei kelių informacijai gauti naudosime Google API.
4. Front-end bus naudojama JavaSript.

# Sistemos kūrimo technologijos

Savo projektui, tiek frontend‘ui, tiek backend‘ui pasirinkome JavaScript, todėl nes mums lengviau šia kalba buvo realizuoti savo paieškos algoritmus. Kita priežastis būtų: geras suderinamumas su html kalba, pvz su html kalba sukurtą lentelę yra patogu užpildyti su JavaScript programavimo kalba. Be to, JavaScript savo sintakse yra artimesnė programavimo kalboms, su kuriomis mes jau dirbome anksteniais pusmečiais(Java,C#).

# Sistemos kūrimo įrankiai

Mes pasirinkome easybacklog sistemą. Pasirinkome, nes ji mums paliko įspūdį savo paprastu naudojimu, nėra perkrauta įvairiais nereikalingais dalykais ir pan.

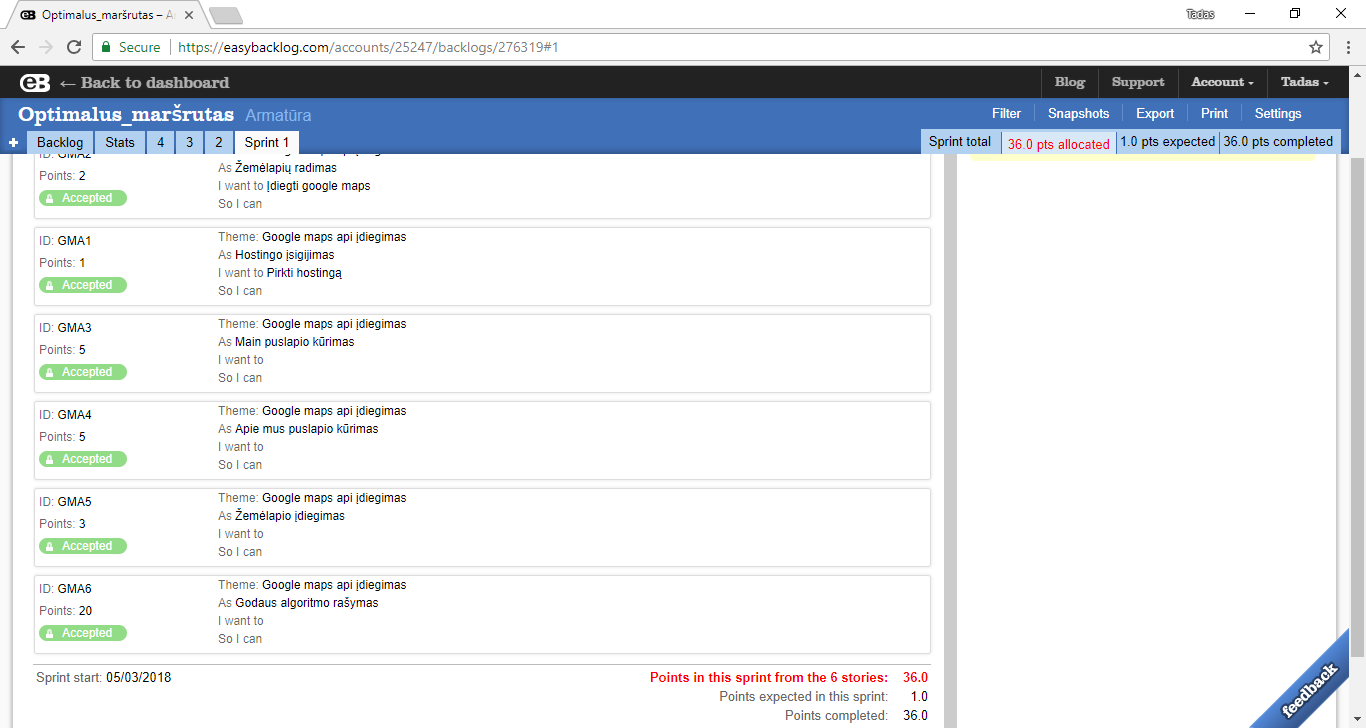
# Įrankių panaudojimas sistemos kūrimo procese

## Kūrimo proceso valdymo įrankis

Procesų valdymui mes pasirinkome easybacklog sistemą. Pasirinkome būtent ją, nes ją yra gan paprasta naudoti, nėra įvarių papildomų bereikalingų priedų kaip to do listai ir pan.

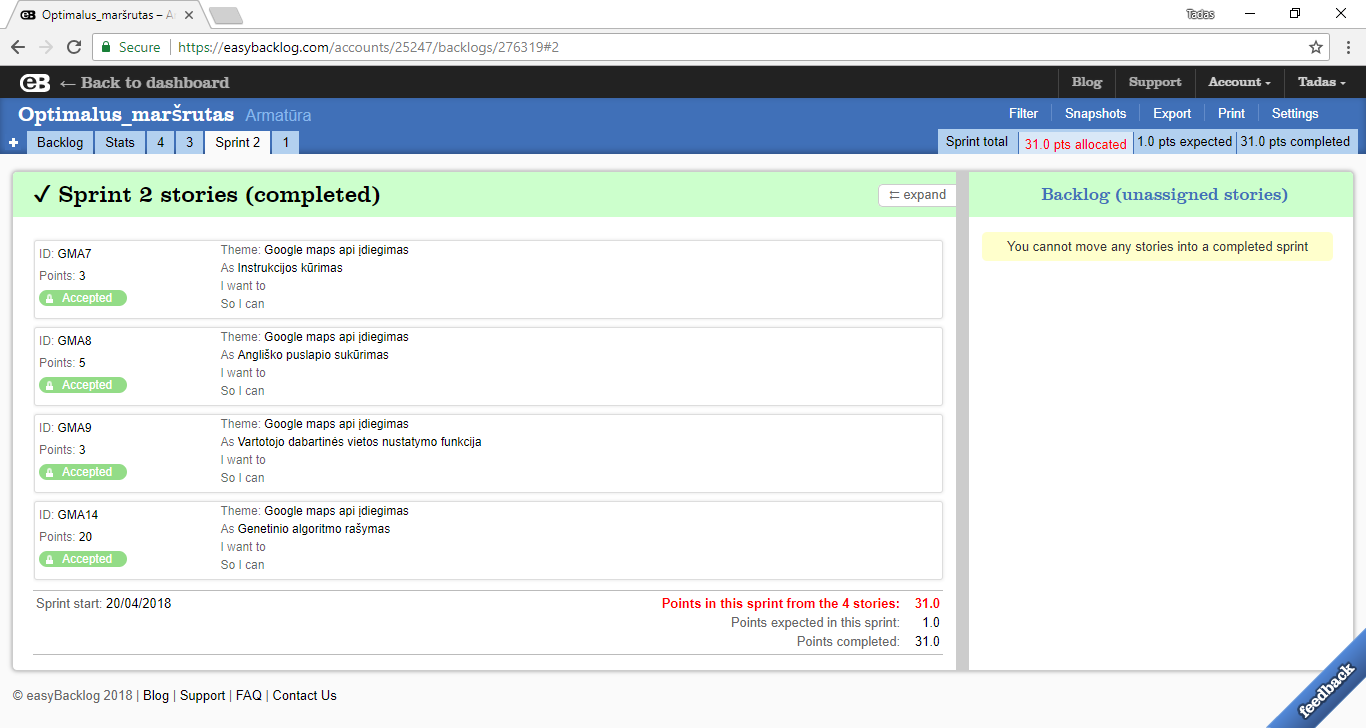
Šiose sekančiose nuotraukose galite pamatyti mūsų įvykdytus sprintus.

Pirmo sprinto metu, mes pasidarėme vaidinamuosius dizaino ,,griaučius“ bei įdiegėme brute-force paieškos algoritmą.



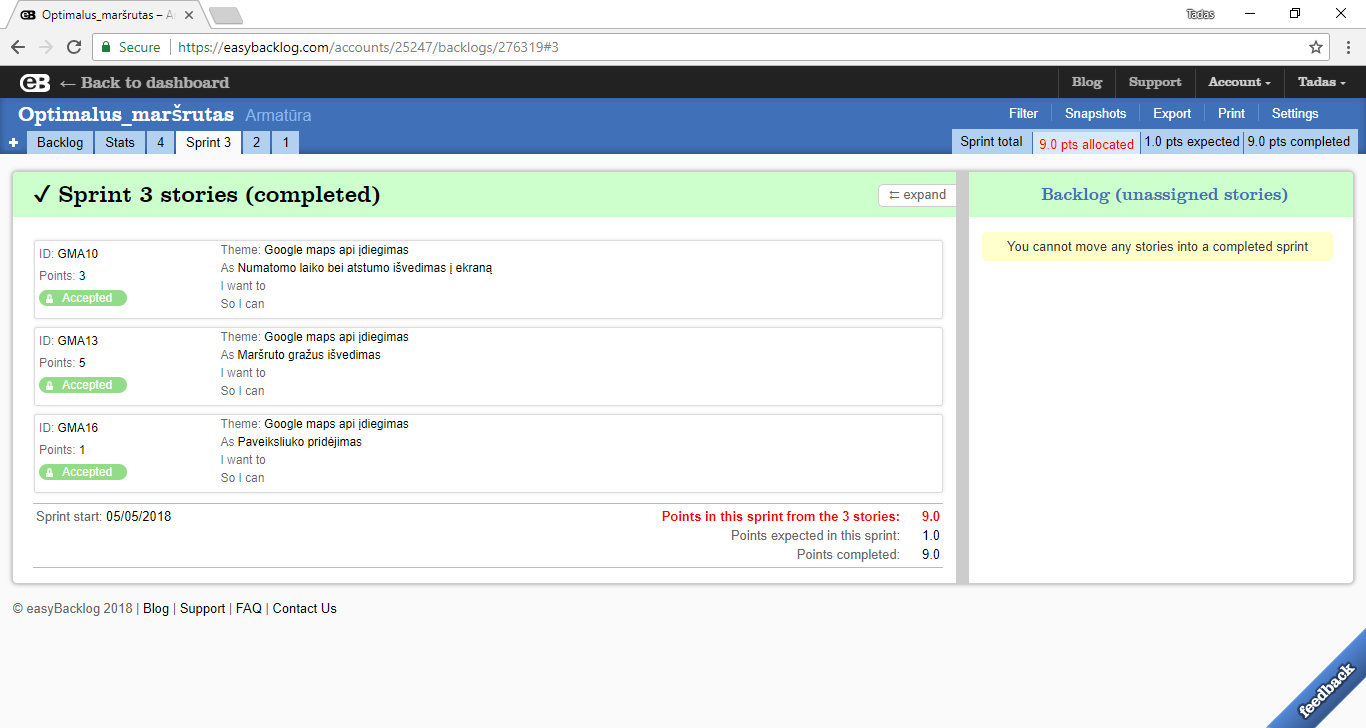
1 pav. 1 sprinto užduotys bei jų vertinimai

Antrojo sprinto metu buvo sukurtas antrasis algoritmas(genetinis) bei įdiegta vietos nustatytmo funkcija. Taip pat buvo sukurta angliška puslapio versija bei instrukcija, kaip naudotis viskuo.



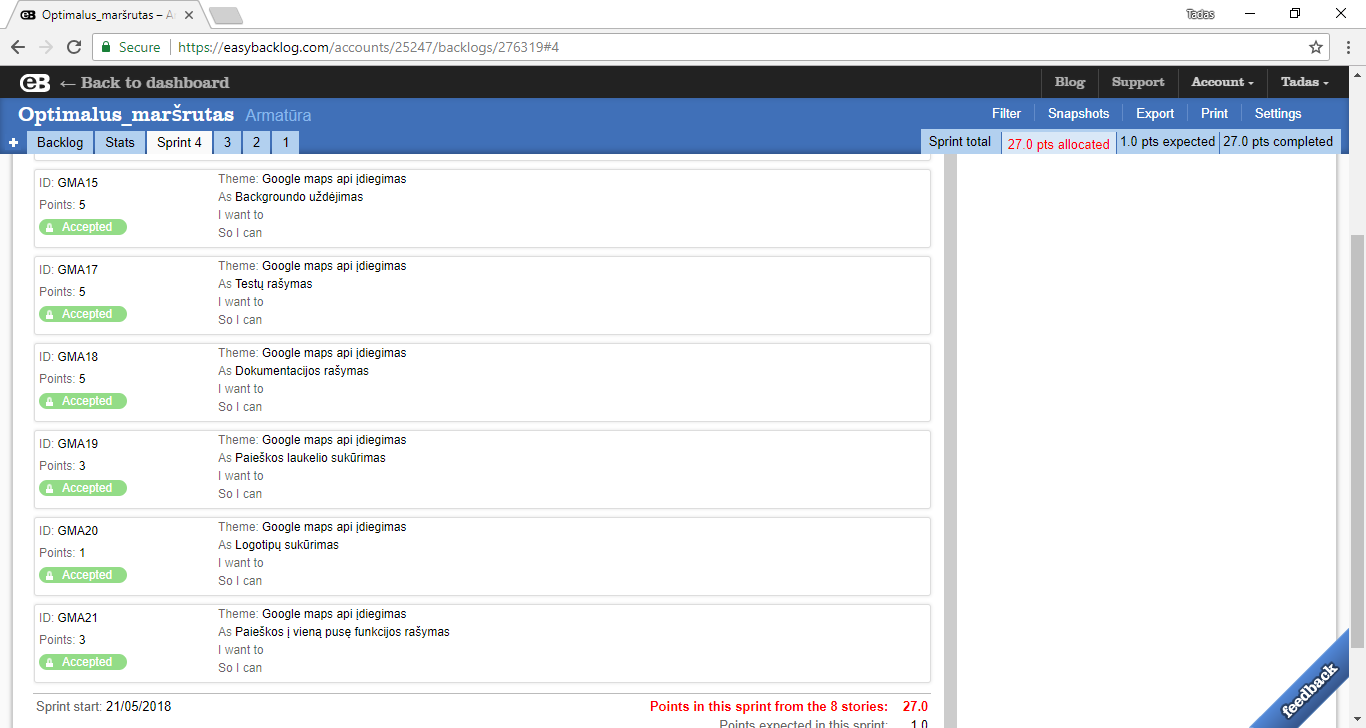
2 pav. Antrojo sprinto apžvalga

Trečiojo sprinto metu buvo sutvarkytas maršruto išvedimas vartotojui(lentelėmis) bei buvo pridėtas dalykas, kurio pagalba yra galima matyti apytikslį kelionės atstumą bei laiką.



3 pav. Trečiojo sprinto apžvalga

Paskutinio sprinto metu buvo pridėta funkcijų kaip maršruto radimas į vieną pusę, paieškos laukelio sukūrimas, backgroundo tvarkymas, kodo optimizacija bei testų rašymas.



4 pav. Ketvirtojo sprinto įvykdytos užduotys

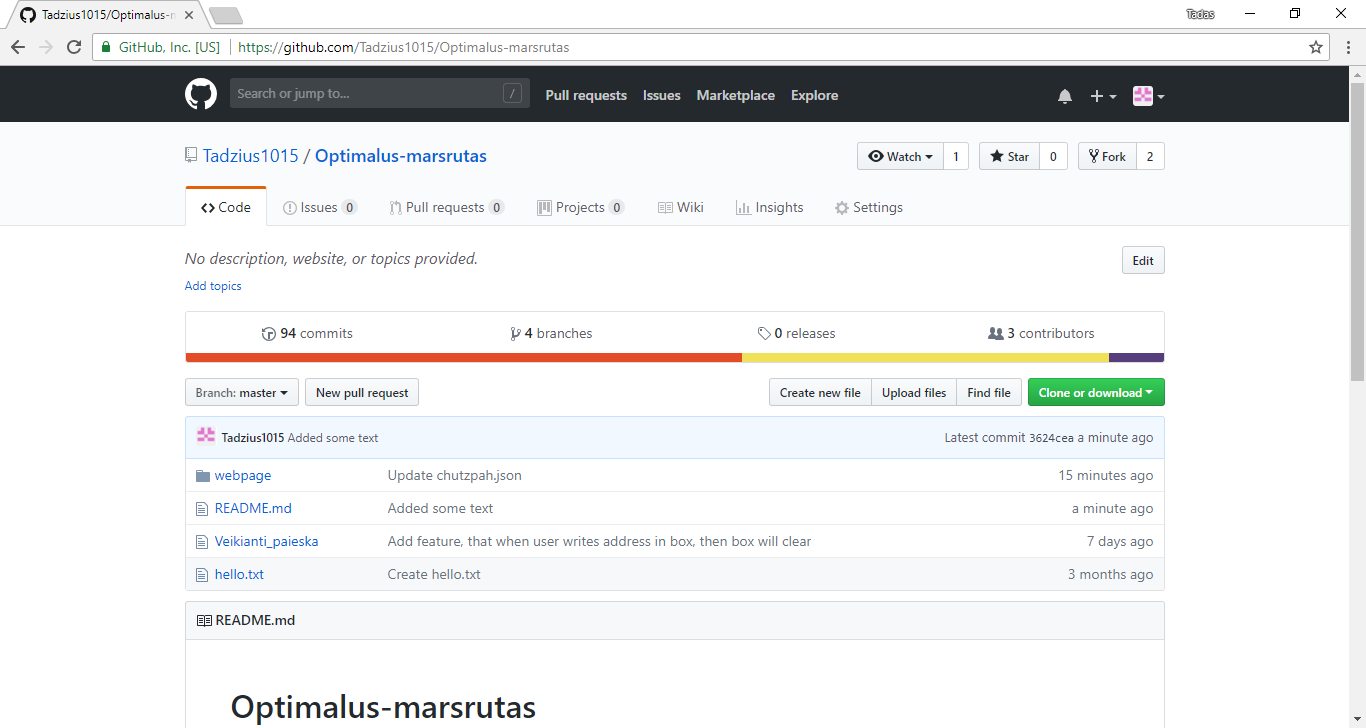
5 paveikslėlyje galime matyti burn-down ir burn-up grafikus, kurių vienas simbolizuoja, kiek po kiekvieno sprinto liko taškų atlikti iki projekto pabaigos, o kitame grafike raudona spalva rodo kiek mes per konkretų sprintą įvykdėme taškų, o mėlyna, kiek iš viso jų yra po kiekvieno sprinto. Iš antro grafiko galima spręsti kiek laiko panašiai dar liko užduočiai atlikti.



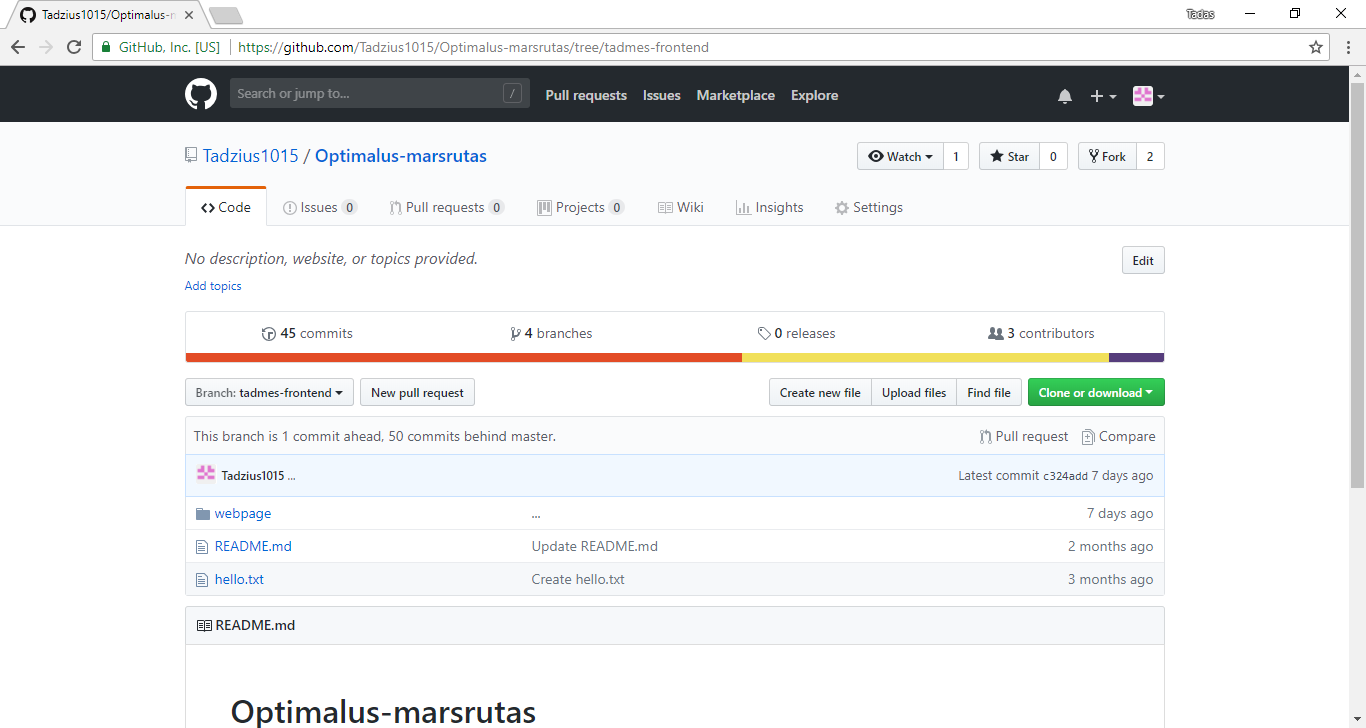
5 pav. Mūsų projekto statistika(burn-down ir burn-up grafikai)

## Saugykla

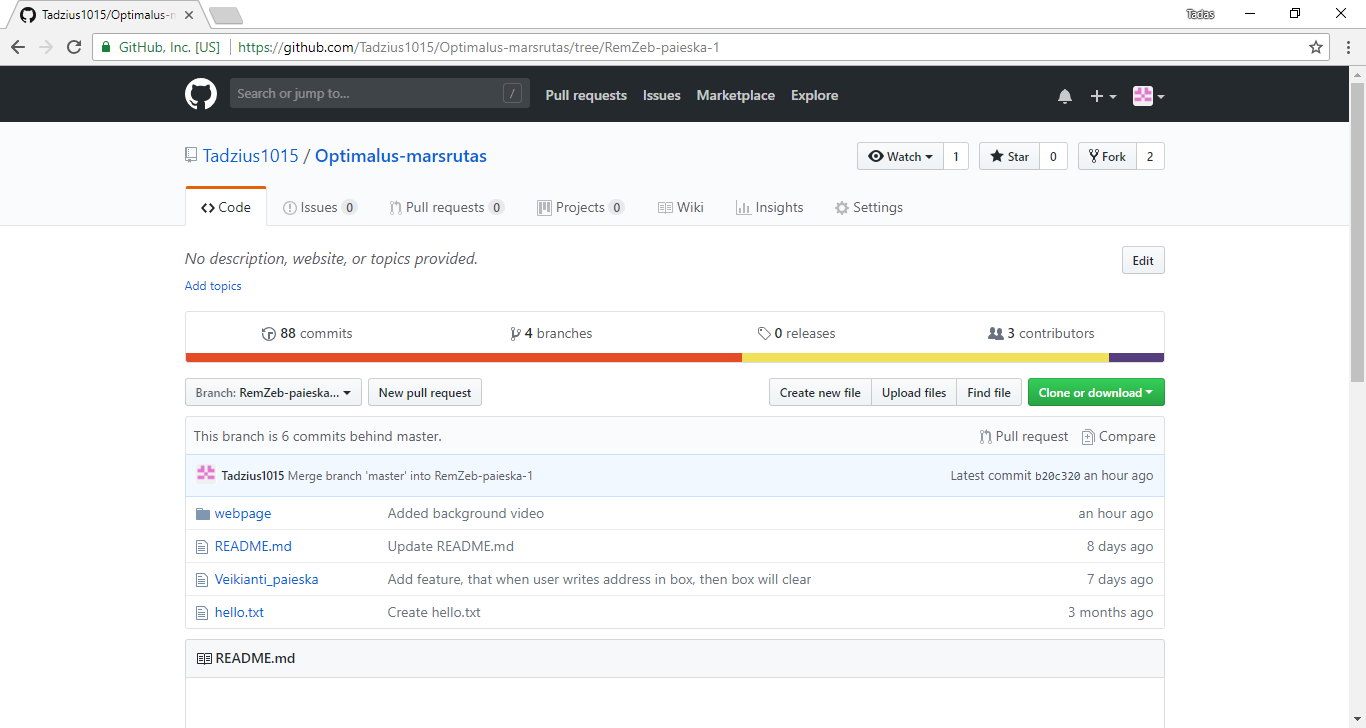
Savo projekte naudojome github repozitoriją. Saugykloje turime 4 šakas: master, Remigijaus, Tado ir Deivido. Jas pamatysite sekančiose nuotraukose. Remigijaus šakoje yra visas back-end kodas, nes jis buvo atsakingas už algoritmų realizavimą žemėlapyje. Toliau tiek Tado, tiek Deivido šakose galima rasti front-end kodo failus, kuriuos pagal galimybę abu kurdavome/redaguodavome.



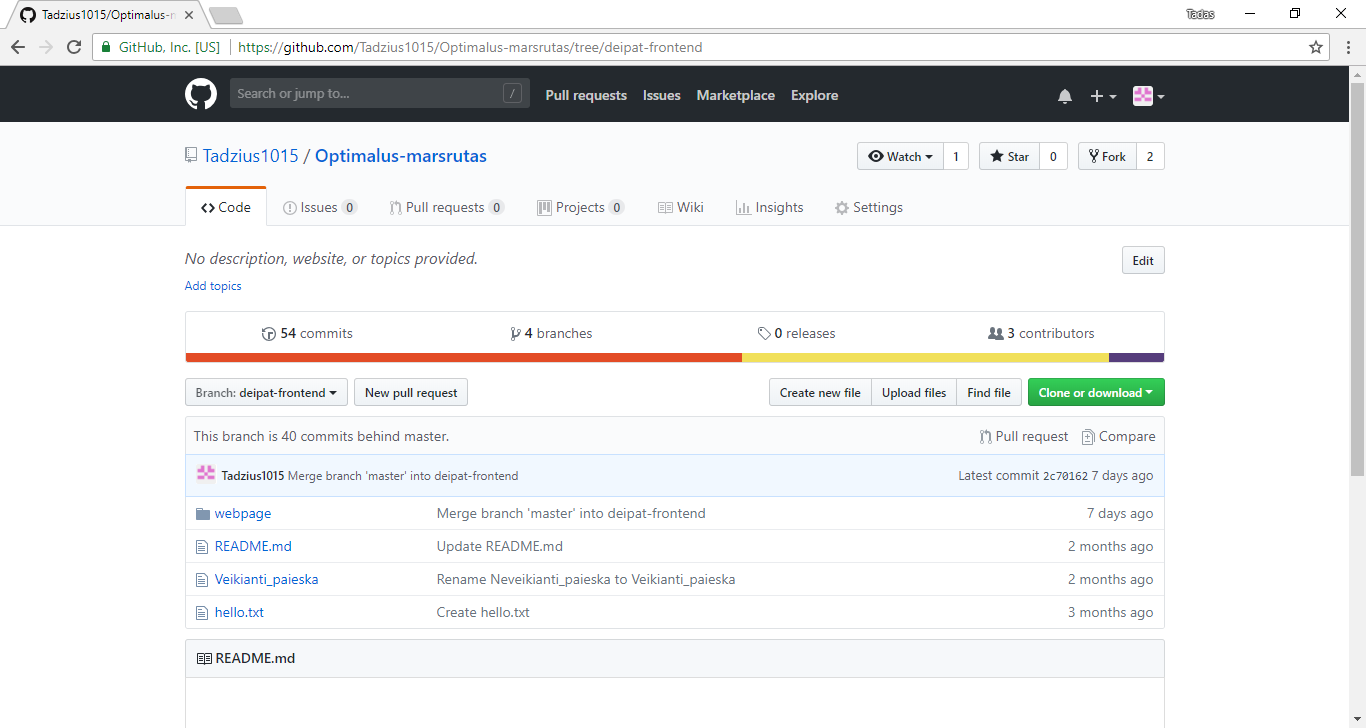
6 pav. Master šaka



7 pav. Tado šaka



8 pav. Remigijaus šaka

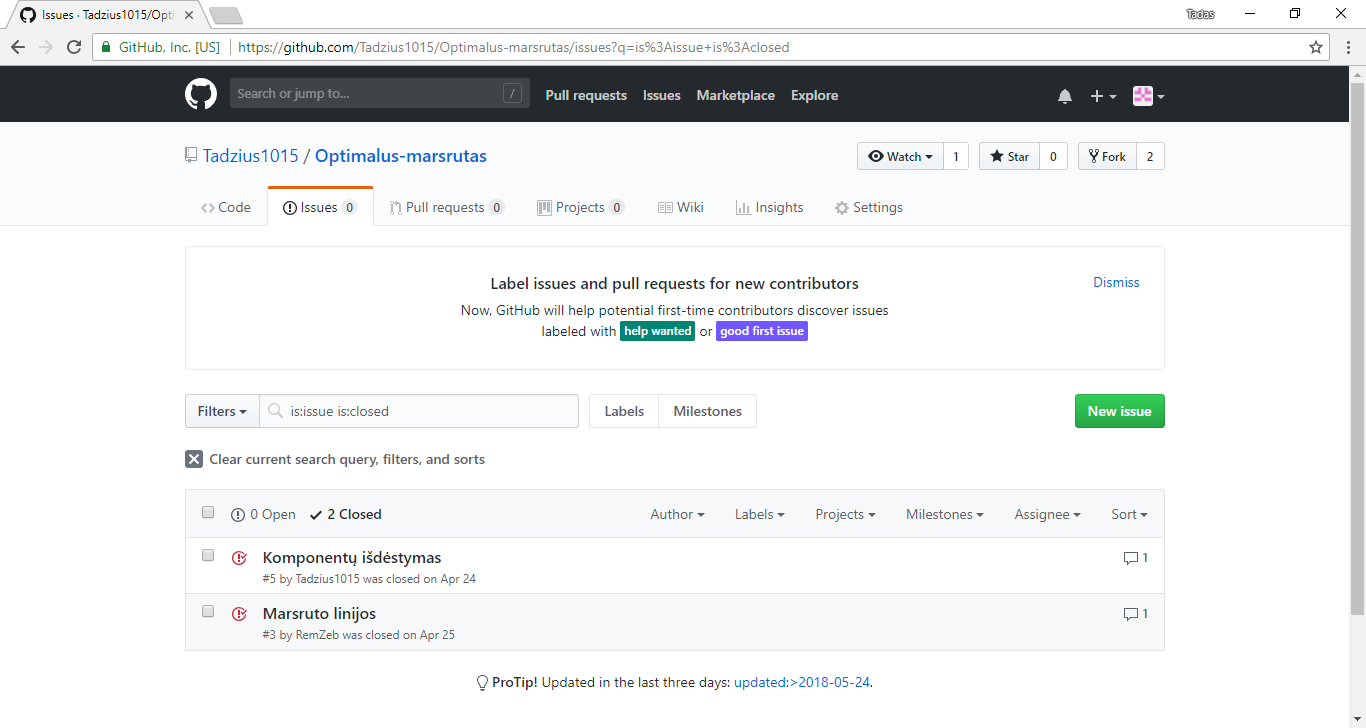


9 pav. Deivido šaka

## 

## Klaidų, darbų, idėjų sekimo įrankis

Šį įrankį naudojome github repozitorijoje. Atlikdami darbą sukūrėme keletą iššūkių užklausų bei keletą merge‘nimo užklausų. Jas galite pamatyti sekančiose nuotraukose:



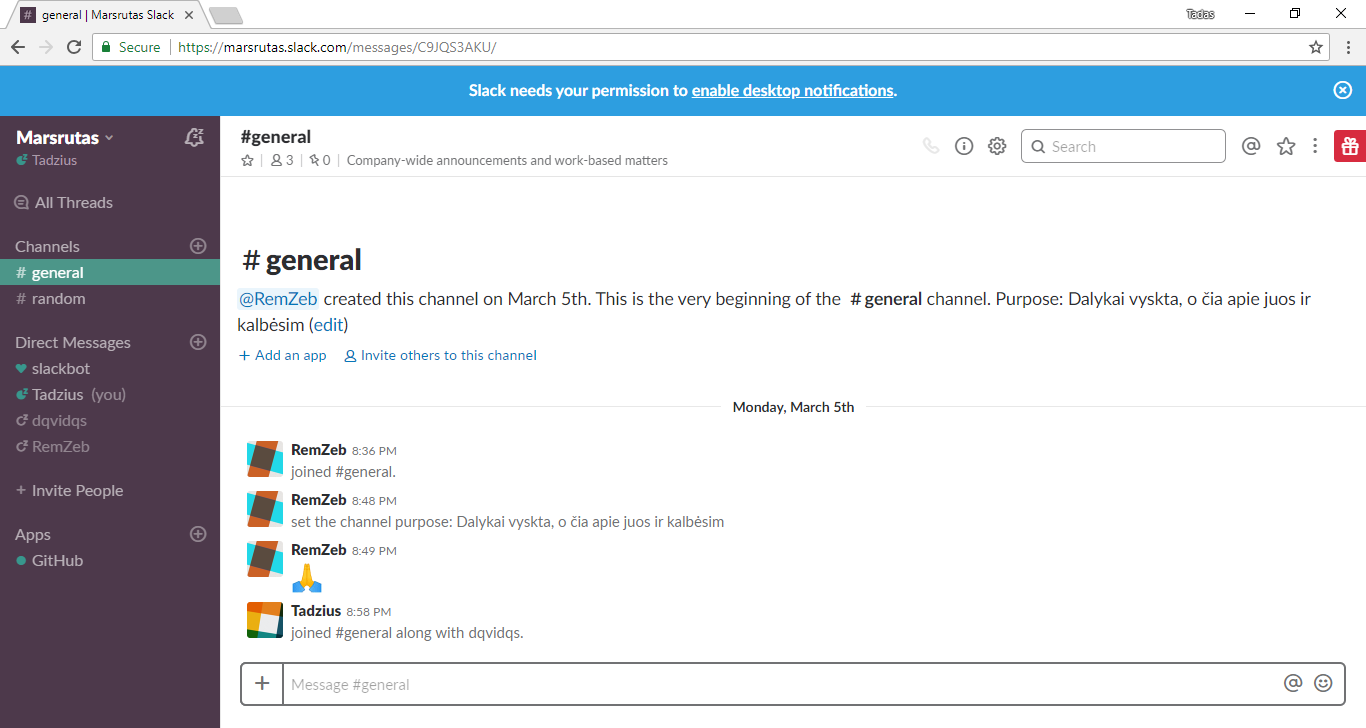
10 pav. Iššukių užklausos.

## 

11 pav. Suliejimo užklausos.

## Bendradarbiavimo įrankis

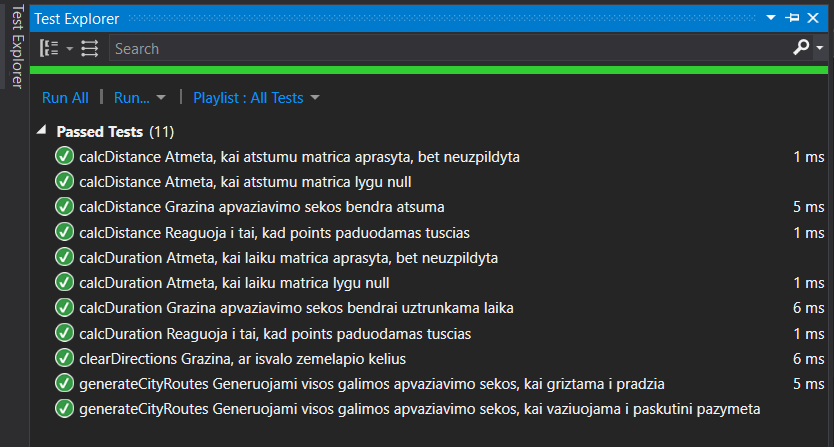
Kadangi visi komandos nariai gyvename viename bendrabutyje, todėl mes ir nenaudojome jokio bendradarbiavimo įrankio. Žinoma, mes esame susikūrę slack kanalą, tačiau jame pokalbiai nevyksta. Jeigu ko nors reikia mums, tai tiesiog susitinkam pas kurį nors kambaryje ir apatariame iškilusias bėdas, padarytus darbus ar dalinamės sugeneruotomis idėjomis.



12 pav. Mūsų komandos bendradarbiavimo įrankis

## Testavimo įrankiai/karkasai

Testavimui naudojome per visual studio esančią biblioteką pavadinimu chutzpah, kuri palaiko mocha, jasmine ir kitokius testavimo įrankius. Mes patys naudojomės jasmine testavimo įrankiu. Kadangi daug mūsų funkcijų yra tiesiogiai surištos su google api, tai todėl mes testavome tik keleta savo funkcijų, kaip atstumo radimas, žemėlapio išvalymas ir pan. Iš viso mes parašėme vienuolika testų, kurie atspindi, kaip mūsų tam tikros funkcijos veikia su tam tikrais duodamais duomenimis ir kokie rezultatai turėtų grįžti vartotojui.



13 pav. Atlikto testavimo rezultatai

# Dokumentacija naudotojui

Labas, ačiū, kad nepasikuklinai ir pradėjai naudoti mūsų sukurtą kelių paieškos sistemą. Mūsų projekto komanda tikisi, jog naudodamas šį projektą, atrasi savo reikiamą maršrutą ir taip sutaupysi savo Labas, ačiū, kad nepasikuklinai ir pradėjai naudoti mūsų sukurtą kelių paieškos sistemą. Mūsų projekto komanda tikisi, jog naudodamas šį projektą, atrasi savo reikiamą maršrutą ir taip sutaupysi savo laiką bei automobilio degalų sanaudas ir taip prisidėsi prie švaresnės aplinkos ne tik sau, bet ir savo šeimos nariams, giminaičiams, draugams, pažįstamiems bei visiems kitiems Lietuvos(ir ne tik) gyventojams. Mes manome, jog tai yra puikus dalykas 😊 Tad nelaukdami, pradedame pažintį su mūsų sistema.

Pirmiausia, nuėjęs į savo interneto naršyklės programą, suvesk mūsų sistemos puslapio adresą, kuris yra: [www.zemelapis.we2host.lt](http://www.zemelapis.we2host.lt). Atsidarius svetainei, turėtų atsirasti toks langas:



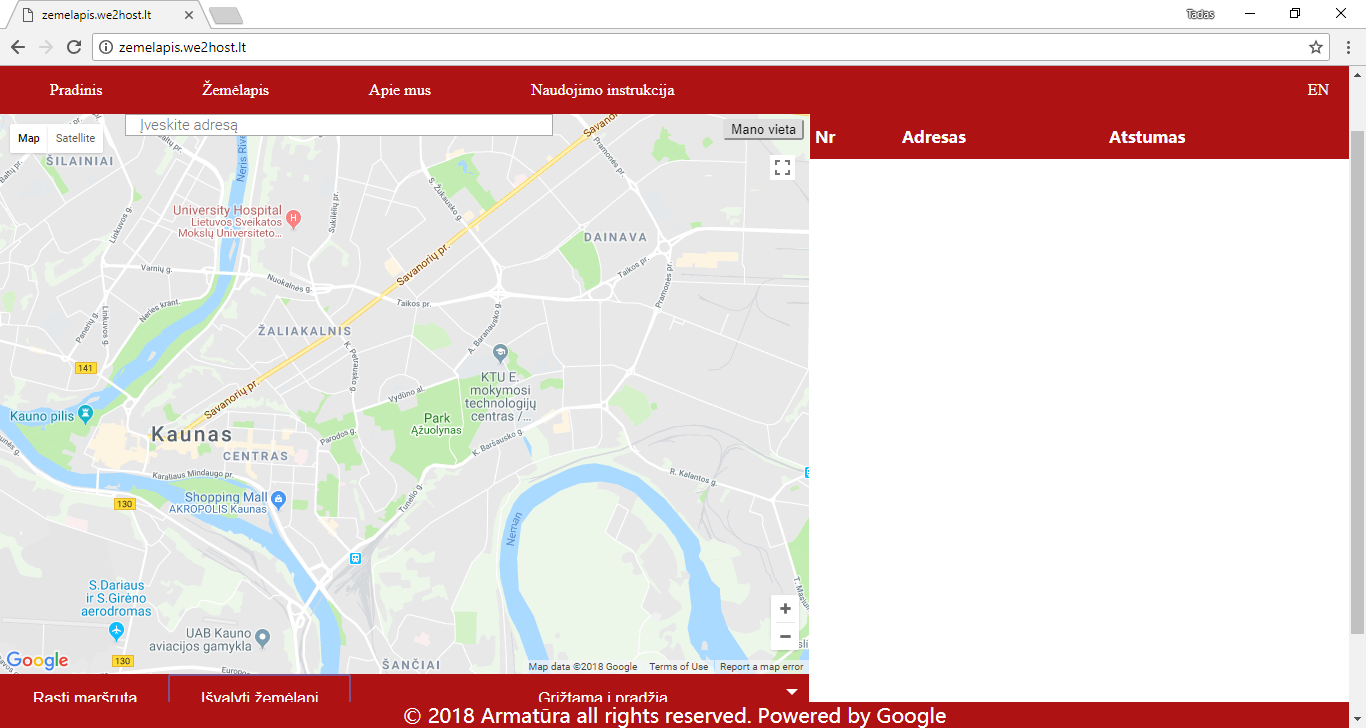
14 pav. pradinis puslapio langas

Paspaudus mygtuką Pradinis, atsiranda pagrindinis puslapio langas, kuriame pradeda eiti video apie google street view sistemą, parodant žymias pasaulio vietoves/miestus. Pirmame šio puslapio plane galime rasti pasisveikinimo bei palinkėjimo tekstus ir patarimą, jog prieš naudojantis puslapio funkcijomis, reikėtų perskaityti naudojimo instrukciją. Sekančioje nuotraukoje galime pamatyti, kaip atrodo pagrindinis langas:

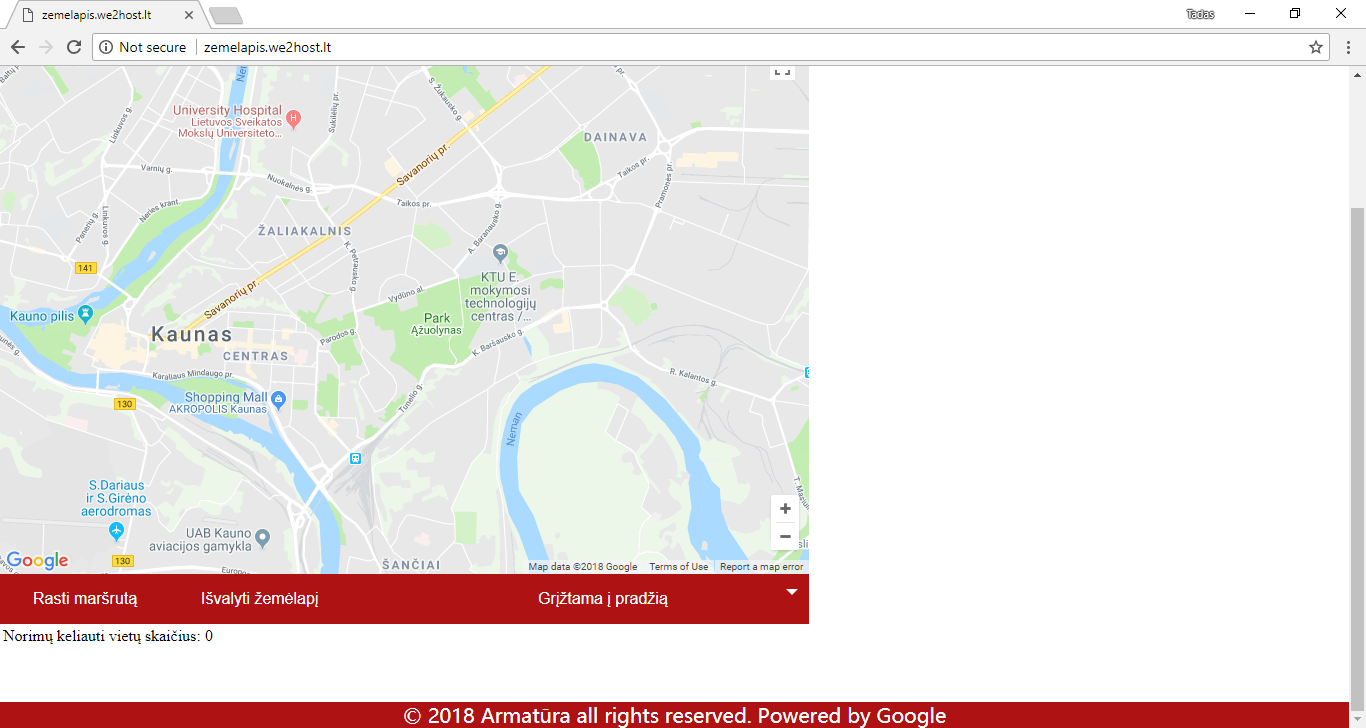


15 pav. puslapio pagrindinis langas

Toliau einant per puslapį galima rasti mygtuką žemėlapis. Paspaudus jį, atsiras toks vaizdas:

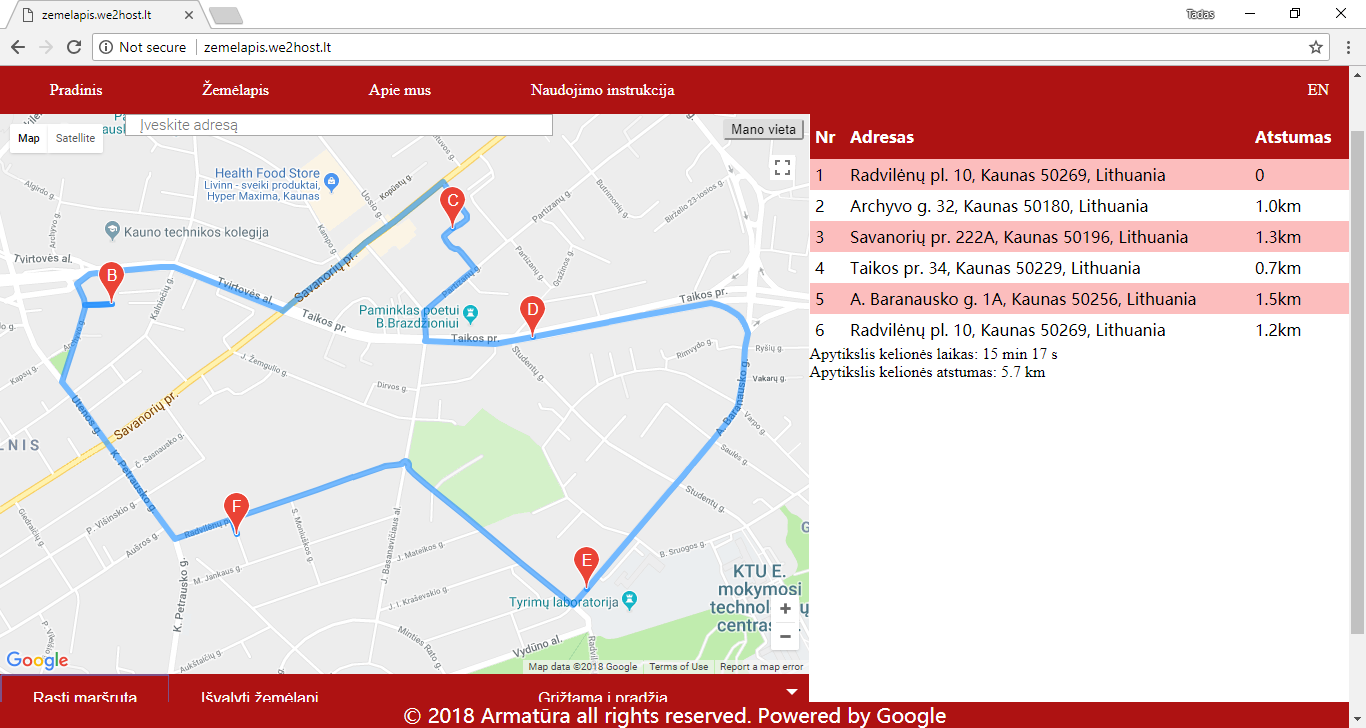


16 pav. žemėlapio puslapis 1 paveikslėlis

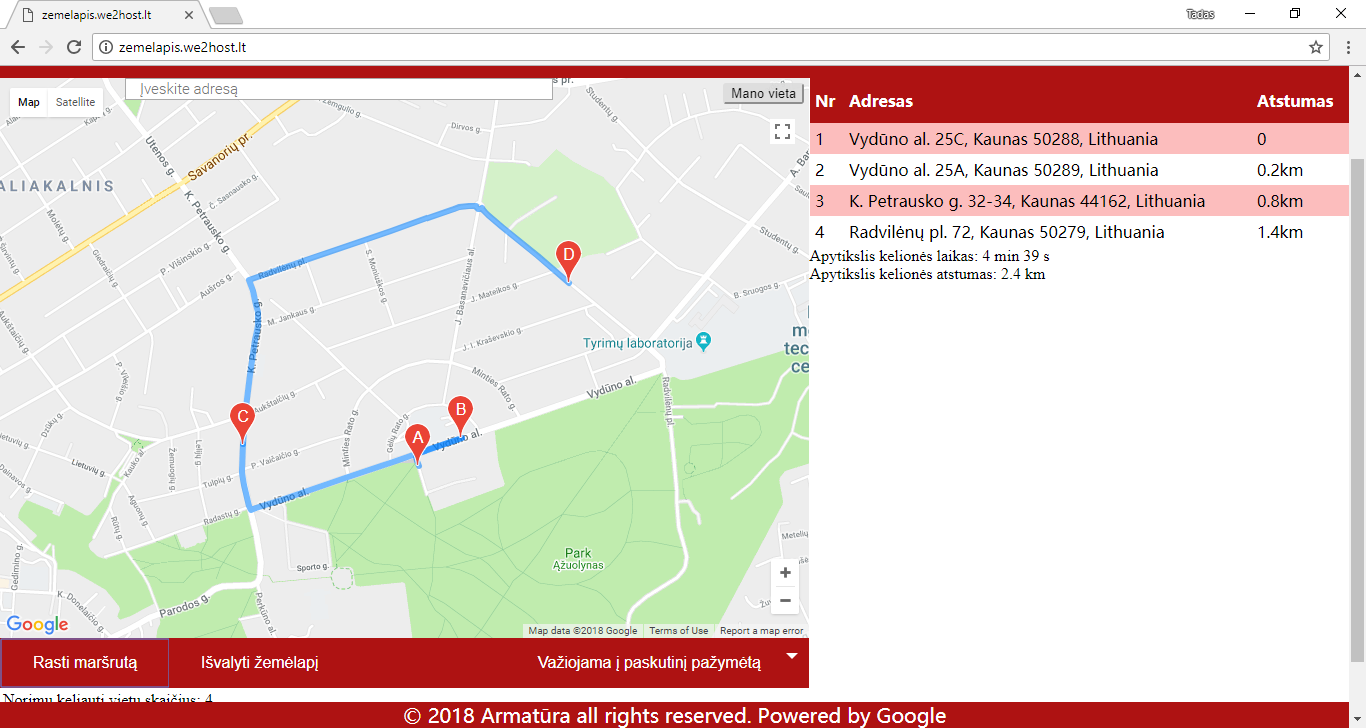


17 pav. žemėlapio puslapis 2 paveikslėlis

Kaip matote, šiame puslapio dalyje yra visa pagrindinė esmė – maršruto paieška. Norint rasti maršrutą, reikia funkcijai duoti adresus. Juos galima duoti trimis būdais: juos įvedant, pažymint žemėlapyje norimą vietą ar miksuotai, t.y. tiek įvedant adresus, tiek juos pažymint. Norint įvesti adresą, reikia žemėlapio viršuje susirasti teksto laukelį, kuriame parašyta searchbox ir ant jo paspaudus vesti norimą adresą. Užtenka įvesti nors vieną raidę ir laukelis išmeta pasiūlymus su adresais, tačiau, jei netenkina pasirodę adresai, toliau patartina vesti dar po vieną raidę/skaičių kol pasirodys norima vieta. Pasirodžius tinkamam adresui, reikia tada ant jo paspausti ir žemėlapyje atsiras žymeklis, žymintis vietovę. Po sėkmingo įvedimo, teksto laukelis išsivalys ir tada bus galima vesti naują adresą ir tai galėsite daryti kiek tik norėsite kartų(bet ne daugiau kaip 10). Norint pažymėti vietą žemėlapyje, reikia jį slinkti su paspaustu kairiuoju pelės mygtuku ir susiradus tinkamą vietą, paspausti kairijį pelės mygtuką ant jos ir tada žemėlapyje jinai pasižymės automatiškai. Darant mixu, galima tiek vesti, tiek žymėti vietas(ne daugiau 10). Pastaba: puslapio apačioje, po mygtukais ,,rasti maršrutą‘‘ ir ,,išvalyti žemėlapį‘‘ galime rasti tekstą su atsinaujinančiu skaičiumi, kuris nurodo, kiek adresų buvo pažymėta/įvesta žemėlapyje. Susivedus norimus adresus, tada galima ieškoti optimalus maršruto, tačiau prieš pradedant paiešką, reikia pasirinkti, ar norima grįžti į maršruto pradžią, t.y. iš paskutinio taško važiuoti atgal į pradžią, ar norima važiuoti tik į paskutinį pažymėtą/įvestą tašką. Pasirinkus maršruto tipą, galime ieškoti optimalus maršruto. Norint tai rasti, reikia paspausti mygtuką ,,rasti maršrutą‘‘ ir tada dešinėje puslapio pusėje iššoks lentelė su eilės numeriu, kuris nurodo kelintą tą tašką turėsi aplankyti. Toliau dar yra adreso pavadinimas bei atstumas, kuris yra nurodomas nuo prieš tai esančio taško iki esamo taško. Žemėlapyje galima matyti virš pažymėtų taškų raides, kurios yra surikiuotos abėcėlės tvarka didėjančiai ir nurodo į kurią vietą vairuotojas važiuos pirmiau. Pvz A nurodo, kad vairuotojas važiuos pirma į tą vietą, B nurodo, kad po A, jis važiuos į B tašką, po to į C ir t.t., kol baigsis maršrutas. Lentelės apačioje galima pamatyti apytikslį keliavimo laiką bei atstumą. Be to, žemėlapyje tarp taškų yra nupiešiamas optimalus maršruto kelias, kurį priartinus, galima pamatyti konkrečiau per kokias gatves/vietoves bus vykstama. Norint ieškoti naują maršrutą, reikia išvalyti žemėlapį bei lentelę, tai padaryti galima, paspaudus mygtuką ,,išvalyti žemėlapį“ ir šie komponentai išsivalys. Tada vėl bus galima daryti tai ką minėjau aukščiau.



18 pav. optimalus maršruto paieškos pavyzdys, naudojant paiešką su grįžimu atgal.



19 pav. optimalus maršruto paieškos pavyzdys, naudojant paiešką su važiavimu į paskutinį tašką

Toliau einant per puslapį, galime rasti mygtuką ,,projekto komanda’’. Jį paspaudus pasirodys visų komandos narių vardai, pavardės, kontaktinė informacija bei nuotraukos:



20 pav. projekto komanda

Jeigu naudojantis mūsų puslapiu, randate tam tikrą neatitikimą, neveikimą, ar išvis norite paklausti klausimų apie puslapį ir jo veikimą, nesiskuklinkite ir rašykite bet kuriam komandos nariui, mielai jus išklausysime ir atsakysime į norimus klausimus ar sutvarkysime kas neveikia mūsų puslapyje. 😊

Na ir pabaigai, pas mus puslapyje galite rasti trumpą vartotojo instrukcija, kurioje yra surašyti pagrindiniai dalykai, norint naudotis mūsų sukurta sistema. Tačiau prieš naudojant, patariame perskaityti šią(ilgąją) naudojimo instrukciją.

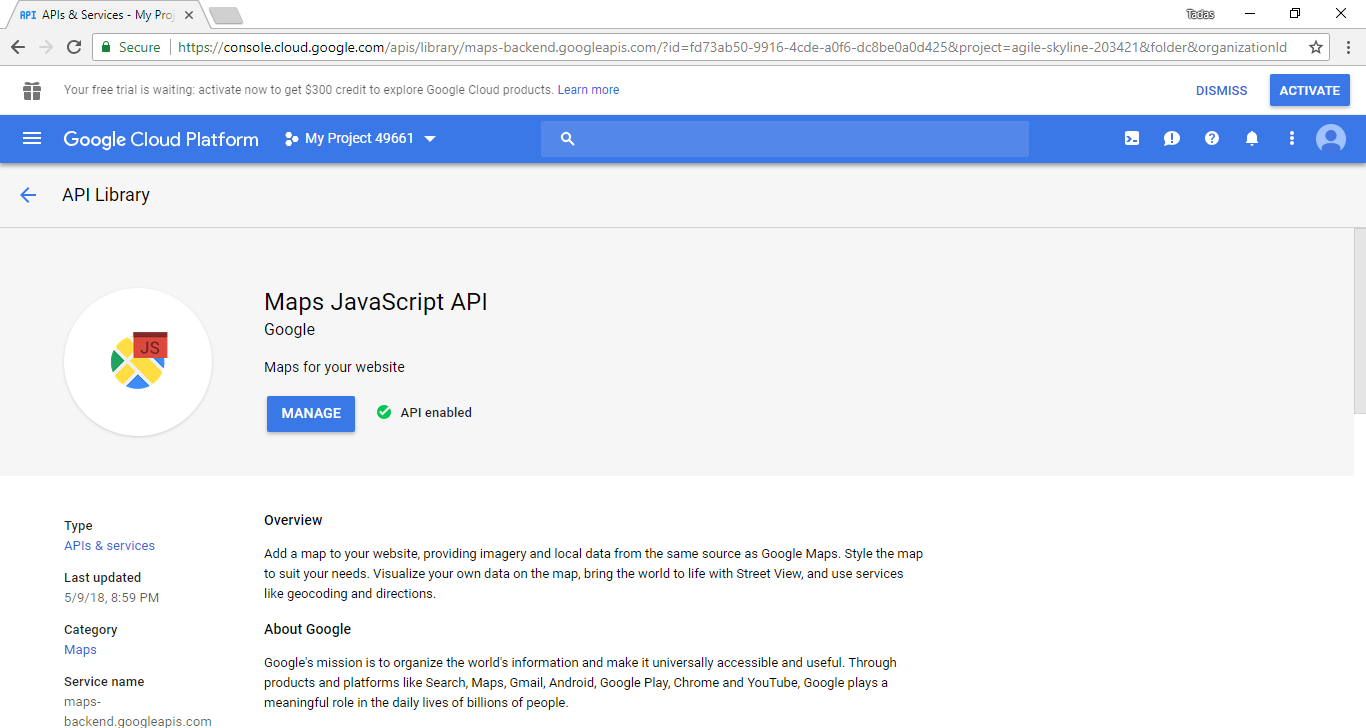


21 pav. trumpa naudojimo instrukcija

P.S puslapis taip pat turi anglišką versiją. Ten viskas yra taip pat, kaip ir lietuviškoje, tačiau visi užrašai pateikiami anglų kalba. Todėl angliška vartotojo instrukcija bus padaryta kažkada… ☹

# Rezultatų apibendrinimas ir išvados

Atlikę numatytus projekto darbus, gavome maršruto planavimo sistemą, kurios pagalba galime surasti gana optimalų maršrutą ne daugiau kaip 10-imt taškų. Atlikdami šią užduotį, visi komandos nariai labiau pagilino žinias apie html programavimą, susipažino su JavaScript programavimo kalba, jos įpatumais, išmokome jos pagrindus bei pritaikėme savo projekte. Toliau sužinojome apie google api sistemą, kurios keletą komponentų taip pat panaudojome savo darbe.



22 pav. Komponento įrodymas

Manome, jog mūsų sukurtas puslapis turės tam tikrą išliekamąją vertę, nes kiek ieškojome analogų, tai jų mes neradome, kalbant apie Lietuvos rinką. Reikia gerai išreklamuoti savo puslapį ir mūsų nuomone, šį puslapį pamėgs vis daugiau suinteresuotų žmonių.

Problemos su kuriomis susidūrėme projekto darymo metu:

* + - 1. Naršyklių nesuderinamumas.
      2. Funkcijų neveikimas skirtingose naršyklėse.
      3. Dizainas(sukurti tokį dizainą, kuris įtiktų kiekvieno skoniui).
      4. Komponentų išdėstymo tvarka(aiškiai ir suprantamai).
      5. SSL saugos sertifikato neturėjimas, nes tiek google chrome, tiek mozila firefox neleidžia nustatyti esamos vietos, jei nėra https standarto.

# Literatūra

|  |  |
| --- | --- |
| [1] | „JavaScript Promises: an Introdction,“ by Jake Archibald. Available: https://developers.google.com/web/fundamentals/primers/promises |
| [2] | „Places serach box,“ by Google. Last update in: May 10th, 2018 Available: https://developers.google.com/maps/documentation/javascript/examples/places-searchbox   |  |  | | --- | --- | | [3] | „Async functions - making promises friendly,“ by Jake Archibald. Available: https://developers.google.com/web/fundamentals/primers/async-functions | | [4] | „Distance matrix API,“ by Google. Last updated in: May 10th, 2018 Available: https://developers.google.com/maps/documentation/distance-matrix/intro | | [5] | „Geolocation API,“ by Google. Last updated in: May 10th, 2018 Available: https://developers.google.com/maps/documentation/geolocation/intro | | [6] | „Genetic algorithm: an implement using JavaScript and HTML5,“ by Prajan Das. Published in: Sep 12th, 2016 Available: https://www.codeproject.com/Articles/1127321/Genetic-algorithm-an-implementation-using-JavaScri | | [7] | „Machine Learning: Introduction to Genetic Algorithms,“ by Burak Kanber. Published in: Sep 4th, 2012 Available: https://burakkanber.com/blog/machine-learning-genetic-algorithms-part-1-javascript/ | |
|  | |  |  |  |  | | --- | --- | --- | --- | | [8] | „Coding Challenge #35.1: Traveling Salesperson,“ by The Coding Train. Uploaded in: Aug 24th, 2016 Available: https://www.youtube.com/watch?v=BAejnwN4Ccw | „Coding Challenge #35.1: Traveling Salesperson,“ by The Coding Train. Uploaded in: Aug 24th, 2016 Available: https://www.youtube.com/watch?v=BAejnwN4Ccw | „Coding Challenge #35.1: Traveling Salesperson,“ by The Coding Train. Uploaded in: Aug 24th, 2016 Available: https://www.youtube.com/watch?v=BAejnwN4Ccw | |