Kauno technologijos universitetas

Informatikos fakultetas

Keliavimo kartu platforma „JourneyFy“

Baigiamasis bakalauro studijų projektas

Lukas Kuzmickas

Projekto autorius

Asist. Dr. Mikas Binkis

Vadovas

Kaunas, 2025

Kauno technologijos universitetas

Informatikos fakultetas

Keliavimo kartu platforma „JourneyFy“

Baigiamasis bakalauro studijų projektas

Programų sistemos (6121BX012)

Lukas Kuzmickas

Projekto autorius

Asist. Dr. Mikas Binkis

Vadovas

Doc. Dr. Eglė Butkevičiūtė

Recenzentė

Kaunas, 2025

Kauno technologijos universitetas

Informatikos fakultetas

Lukas Kuzmickas

Keliavimo kartu platforma „JourneyFy“

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;

2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąraše;

3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);

4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Lukas Kuzmickas

Patvirtinta elektroniniu būdu

Kuzmickas Lukas. Keliavimo kartu platforma „JourneyFy“. Bakalauro studijų baigiamasis projektas Asist. Dr. Mikas Binkis; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos mokslai, Programų sistemos.

Reikšminiai žodžiai: kelionės, transportas, pavežėjas.

Kaunas, 2025. 83 p.

Santrauka

Šiame darbe pristatoma „JourneyFy“ – kelionių tematikos mobilioji aplikacija, sukurta naudojant React Native ir Firebase. Sistema skirta tiek vairuotojams, tiek keleiviams norintiems kartu dalintis kelionėmis, taip sprendžiama kelionės autostopu problema.

Pradiniame etape atlikta Lietuvos rinkoje egzistuojančių sprendimų analizė atskleidė, kad egzistuojančios kelionių platformos turi trūkumų - programėlės neturi papildomo funkcionalumo, vieningos platformos ar pakankamos naudotojų laisvės. Todėl „JourneyFy“ siūlo paprastą, bet galingą įrankį – kelionių platformą visiems.

Aplikacija parašyta TypeScript/Javascript kalba su React Native ir Expo Go, o duomenų bazės dalis - Firebase paslaugomis. Kodo kokybė patvirtinta vienetų, integraciniais, naudotojo sąsajos testais. Galutinis produktas sėkmingai veikia užsakovo lokalioje aplinkoje.

Kuzmickas Lukas. Carpooling platform “JourneyFy”. Bachelor's Final Degree Project / supervisor Asist. Dr. Mikas Binkis; Informatics Faculty, Kaunas University of Technology.

Study field and area (study field group): Computer Sciences, Software Systems.

Keywords: carpooling, travel, journey.

Kaunas, 2025. 83 p.

Summary

This paper presents JourneyFy, a travel-themed mobile application built using React Native and Firebase. The system is designed for both drivers and passengers who want to share rides together, thus solving the hitchhiking problem.

An initial analysis of existing solutions on the Lithuanian market revealed that existing travel platforms have shortcomings - apps lack additional functionality, a unified platform, or sufficient user freedom. Therefore, JourneyFy offers a simple but powerful tool - a travel platform for everyone.

The application is written in TypeScript/Javascript with React Native and Expo Go, and the database part is based on Firebase services. The code quality is confirmed by unit, integration, and user interface tests. The final product successfully works in the customer’s local environment.

Turinys

Lentelių sąrašas **7**
Paveikslų sąrašas **8**
Santrumpų ir terminų sąrašas **10**
Įvadas **12**
1. Analizė **14**
1.1. Techninis pasiūlymas 14
1.1.1. Sistemos apibrėžimas 14
1.1.2. Bendras veiklos tikslas 14
1.1.3. Sistemos pagrįstumas 14
1.1.4. Egzistuojančių sprendimų analizė 15
1.1.5. Prototipai ir pagalbinė informacija 19
1.2. Techninių galimybių analizė 20
2. Projektas **21**
2.1. Reikalavimų specifikacija 21
2.1.1. Komercinė specifikacija 21
2.1.2. Sistemos funkcijos 21
2.1.3. Naudotojo sąsajos specifikacija 43
2.1.4. Realizacijai keliami reikalavimai 45
2.1.5. Techninė specifikacija 46
2.2. Projektavimo metodai 46
2.2.1. Projektavimo valdymas ir eiga 46
2.2.2. Projektavimo technologija 47
2.2.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinė sistemos 47
2.3. Sistemos projektas 48
2.3.1. Statinis sistemos vaizdas 48
2.3.2. Dinaminis sistemos vaizdas 50
3. Testavimas **58**
3.1. Testavimo planas 58
3.2. Testavimo kriterijai 60
3.3. Vienetų testavimas 61
3.4. Integracinis testavimas 63
3.5. Naudotojo sąsajos testavimas 64
4. Dokumentacija naudotojui **66**
4.1. Apibendrintas sistemos galimybių aprašymas 66
4.2. Naudotojo vadovas 67
4.3. Diegimo vadovas 79
4.4. Administravimo vadovas 79
Rezultatai ir išvados **82**
Literatūros sąrašas **83**

Lentelių sąrašas

1 lentelė Egzistuojančių sprendimų analizė. 18
2 lentelė „Prisijungti“ PA specifikacija. 23
3 lentelė „Registruotis“ PA specifikacija. 24
4 lentelė „Priminti slaptažodį“ PA specifikacija. 24
5 lentelė „Peržiūrėti profilį“ PA specifikacija. 25
6 lentelė „Redaguoti profilį“ PA specifikacija. 26
7 lentelė „Peržiūrėti mašiną“ PA specifikacija. 26
8 lentelė „Ištrinti mašiną“ PA specifikacija. 27
9 lentelė „Pridėti mašiną“ PA specifikacija. 27
10 lentelė „Peržiūrėti kelionių istorijas“ PA specifikacija. 28
11 lentelė „Įvertinti kelionę“ PA specifikacija. 28
12 lentelė „Peržiūrėti nustatymus“ PA specifikacija. 29
13 lentelė „Pakeisti slaptažodį“ PA specifikacija. 29
14 lentelė „Pakeisti žemėlapio nustatymus“ PA specifikacija. 30
15 lentelė „Ištrinti paskyrą“ PA specifikacija. 30
16 lentelė „Atsijungti“ PA specifikacija. 31
17 lentelė „Peržiūrėti keliones“ PA specifikacija. 32
18 lentelė „Ištrinti kelionę“ PA specifikacija. 33
19 lentelė „Atšaukti kelionę“ PA specifikacija. 33
20 lentelė „Peržiūrėti kelionės žinutes“ PA specifikacija. 34
21 lentelė „Siųsti žinutę“ PA specifikacija. 34
22 lentelė „Peržiūrėti kelionių prašymus“ PA specifikacija. 35
23 lentelė „Priimti prašymą“ PA specifikacija. 35
24 lentelė „Atšaukti prašymą“ PA specifikacija. 36
25 lentelė „Pradėti kelionę“ PA specifikacija. 36
26 lentelė „Užbaigti kelionę“ PA specifikacija. 37
27 lentelė „Peržiūrėti piniginę“ PA specifikacija. 38
28 lentelė „Pridėti pinigus“ PA specifikacija. 39
29 lentelė „Siųsti pinigus“ PA specifikacija. 39
30 lentelė „Užpildyti kelionės paieškos kriterijus“ PA specifikacija. 40
31 lentelė „Ieškoti kelionių“ PA specifikacija. 41
32 lentelė „Peržiūrėti siūlomą kelionę“ PA specifikacija. 41
33 lentelė „Registruotis prie kelionės“ PA specifikacija. 42
34 lentelė „Pridėti kelionę“ PA specifikacija. 42

Paveikslų sąrašas

1 pav. „Bolt“ programėlės lango vaizdas. 15
2 pav. „BlaBlaCar“ sistemos lango vaizdas. 16
3 pav. „BananaCar“ programėlės lango vaizdas. 17
4 pav. Apibendrinta „JourneyFy“ panaudojimo atvejų diagrama. 22
5 pav. Prisijungimo posistemio diagrama. 23
6 pav. Profilio posistemio diagrama. 25
7 pav. Kelionės posistemio diagrama. 32
8 pav. Piniginės posistemio diagrama. 38
9 pav. Rezervacijos posistemio diagrama. 40
10 pav. Mašinos pridėjimo eskizas. 43
11 pav. Pradinio lango ir profilio eskizai. 44
12 pav. Piniginės ir nustatymų langų eskizai. 45
13 pav. Projekto valdymo įrankis Trello. 46
14 pav. „JourneyFy“ diegimo diagrama. 48
15 pav. „JourneyFy“ paketų diagrama. 49
16 pav. „JourneyFy“ esybių ryšių diagrama. 49
17 pav. „Peržiūrėti kelionės žinutes“ sekų diagrama. 50
18 pav. „Siųsti žinutę“ sekų diagrama. 51
19 pav. „Peržiūrėti keliones“ sekų diagrama. 52
20 pav. „Pridėti mašiną“ sekų diagrama - 1 dalis. 53
21 pav. „Pridėti mašiną“ sekų diagrama - 2 dalis. 54
22 pav. „Pridėti mašiną“ sekų diagrama - 3 dalis. 55
23 pav. „Užpildyti kelionės paieškos kriterijus“ sekų diagrama - 1 dalis. 56
24 pav. „Užpildyti kelionės paieškos kriterijus“ sekų diagrama - 2 dalis. 56
25 pav. „Užpildyti kelionės paieškos kriterijus“ sekų diagrama - 3 dalis. 57
26 pav. Testavimo kompiuterio specifikacija. 59
27 pav. „JourneyFy“ vienetų testai. 61
28 pav. „JourneyFy“ vienetų testų padengimas. 62
29 pav. „JourneyFy“ testų rezultatai - 1 dalis. 62
30 pav. „JourneyFy“ testų rezultatai - 2 dalis. 62
31 pav. „JourneyFy“ integraciniai testai. 63
32 pav. „JourneyFy“ integracinių testų padengimas. 63
33 pav. „JourneyFy“ naudotojo sąsajos testai. 64
34 pav. „JourneyFy“ naudotojo sąsajos testų padengimas. 65
35 pav. „JourneyFy“ logotipas. 67
36 pav. Prisijungimo posistemio langai. 68
37 pav. Profilio redagavimo langai. 70
38 pav. Mašinos pridėjimo langai. 70
39 pav. Kelionių istorijos ir įvertinimo langai. 71
40 pav. Nustatymų langai. 72
41 pav. Kelionės peržiūros ir prašymų langai. 73
42 pav. Kelionės rezervacijos ir pradžios langai. 74
43 pav. Kelionės pradžios ir pabaigos langai. 74
44 pav. Piniginės posistemio langai. 75

45 pav. Kelionės paieškos langai. 76
46 pav. Kelionės pridėjimo langai. 77
47 pav. Surastų kelionių langai. 77
48 pav. Pokalbių ir kelionės peržiūros langai. 78
49 pav. Firebase console puslapio vaizdas. 79
50 pav. Firebase siūlomi servisai. 80
51 pav. Naudotojų autentifikacijos duomenų valdymas. 80
52 pav. Firestore duomenų valdymas. 80
53 pav. Firebase storage duomenų valdymas. 81
54 pav. Firebase Realtime database valdymas. 81

Santrumpų ir terminų sąrašas

Santrumpos:

PA - panaudojimo atvejis.

Terminai:

API (angl. Application Programming Interface) – programos arba sistemos suteikiama sąsaja, leidžianti kūrėjams pasiekti jos funkcijas ir keistis duomenimis.

Android Studio - oficiali Google sukurta Android programėlių kūrimo integruota plėtojimo aplinka.

Android - atviro kodo mobiliųjų įrenginių operacinė sistema, sukurta Google – skirta išmaniesiems telefonams, planšetėms ir kitoms įrenginių klasėms.

Bolt – mobili paslauga, leidžianti per programėlę užsisakyti taksi ir kitas mikromobilumo priemones (pvz., elektrinius paspirtukus), jungia keleivius su vairuotojais.

BlaBlaCar – tarptautinė kelionių dalijimosi platforma, kuri jungia vairuotojus, siūlančius laisvas vietas savo automobilyje, su keleiviais, ieškančiais patogesnių ir pigesnių kelionių.

BananaCar – kelionių dalijimosi platforma, leidžianti vairuotojams kurti keliones su laisvomis vietomis savo automobilyje.

Balsamiq – įrankis, skirtas greitai kurti žemo detalumo naudotojo sąsajos eskizus.

Expo Go – nemokama mobili aplikacija, kuri leidžia iš karto atidaryti ir testuoti React Native projektus telefone be jokio papildomo (iOS/Android) kompiliavimo.

JS, JavaScript – JavaScript, scenarijų programavimo kalba.

Firebase - debesų kompiuterijos platforma mobiliosioms ir interneto aplikacijoms, teikianti duomenų bazę, autentifikavimą, analizę ir kitas paslaugas.

React Native – atvirojo kodo JavaScript karkasas, leidžiantis kurti gimtąsias mobiliąsias aplikacijas naudojant React.

TypeScript, TS – tai JavaScript plėtinys su tipų sistema, kuri užtikrina saugesnį ir patikimesnį kodą.

JSON (angl. JavaScript Object Notation) – atviro kodo formatas duomenų struktūroms aprašyti ir perduoti, kurį sudaro objektai su savybėmis ir jų reikšmėmis.

SQL (angl. Structured Query Language) – universali kalba, naudojama reliacinių duomenų bazių duomenų užklausoms rengti, aprašyti ir valdyti.

UML (angl. Unified Modeling Language) – vizualus modeliavimo ir specifikacijų kūrimo įrankis, skirtas objektinių programų sistemų struktūrai ir procesams dokumentuoti.

NPM (angl. Node Package Manager) – tai oficialus „Node.js“ paketų tvarkytuvas, leidžiantis lengvai rasti, diegti ir atnaujinti daugybę JavaScript bibliotekų bei įrankių. Naudojant NPM, projektui pridedami paketiniai modulių sprendimai, o jų priklausomybės valdomos automatiškai.

Kelionės kartu (angl. car-pooling) – tai vairuotojo siūloma kelionė automobiliu, kurioje jis atlaisvina laisvas vietas savo mašinoje, o keleiviai prisijungia ir kartu važiuoja, dalydamiesi kuro bei kitomis kelionės išlaidomis. Taip mažėja vienam žmogui tenkantys kaštai, eismo spūstys.

UI (naudotojo sąsaja): tai produkto vizualinė ir interaktyvi dalis – mygtukai, meniu, formos ir išdėstymas, su kuriais naudotojas tiesiogiai sąveikauja.

UX (naudotojo patirtis): tai bendras naudotojo įspūdis ir jausmai naudojantis produktu, apimantis patogumą, efektyvumą bei pasitenkinimą.

Visual Studio Code - nemokama, tarpplatforminė kodo redagavimo aplinka, kurią sukūrė Microsoft – ji palaiko daugybę programavimo kalbų, turi plėtinių sistemą, automatinio užbaigimo, sintaksės paryškinimą, derinimo įrankius ir integruotą Git palaikymą.

React Native Testing Library - JavaScript biblioteka, skirta React Native komponentų testavimui – leidžia atvaizduoti komponentus izoliuotai, užklausti jų turinio ir imituoti naudotojo sąveiką (pvz., paspaudimus).

Jest - JavaScript testų vykdymo sistema, sukurta Facebook – skirta vienetiniams ir integraciniams testams, palaiko snapshot testavimą, „watch“ režimą ir kodo aprėpties ir padengimo ataskaitas.

Github - debesų pagrindu veikianti platforma, leidžianti saugoti ir valdyti programinės įrangos kodą naudojant Git, bendradarbiauti per „pull request“ bei „issue“ sistemas, stebėti pakeitimus ir automatizuoti kūrimo procesus.

Google Play Store - oficiali „Android“ įrenginių programėlių, žaidimų, filmų, knygų ir muzikos parduotuvė – leidžia naudotojams naršyti, įdiegti bei atnaujinti turinį ir suteikia kūrėjams platformą savo produktams talpinti.

Lint - statinės kodo analizės įrankis – skenuoja šaltinio kodą, randa sintaksės klaidas, stiliaus neatitikimus ir galimas problemas prieš vykdymą.

IOS - Apple mobiliesiems įrenginiams skirta uždaro kodo operacinė sistema – veikia „iPhone“ ir „iPad“, leidžia diegti programas per App Store ir užtikrina griežtą saugumą bei reguliarius atnaujinimus.

Emuliatorius - programinė (arba aparatinė) priemonė, imituojanti kitos platformos ar įrenginio aplinką – leidžia kompiuteryje paleisti ir testuoti programėles taip, lyg jos veiktų tikrame įrenginyje.

MagicDraw – profesionali UML modeliavimui skirta programinė įranga, leidžianti kurti, dokumentuoti ir tikrinti UML diagramas bei modelius.

KYC - tai klientų tapatybės ir rizikos vertinimo procesas.

Over The Air - reiškia bekontakčius duomenų perdavimo ar atnaujinimo būdus.

Įvadas

„JourneyFy“ yra kelionių tematikos programėlė. Šiais laikais transportas tapo neatsiejama kasdieninio gyvenimo dalimi, todėl atsirado poreikis optimizuoti kelionių išlaidas ir rasti efektyvius ir patikimus transporto sprendimus. Programėlė suteikia patogią platformą tiek vairuotojams, tiek keleiviams susisiekti ir planuoti keliones. Šios kelionės remiasi autostopo principu, kelionės yra pilnai optimizuojamos. Taip sudaroma abipusė nauda pačiam vairuotojui ir keleiviui. Šie transporto dalijimosi sprendimai [1], padeda sumažinti kelionių kainų kaštus, dėl nuolat kintančių ekonominių sąlygų – didėjančių degalų kainų ir kitų transporto išlaidų. Tuo pačiu, programėlė suteikia keleiviams lankstumą, nes tradicinių transporto alternatyvų, kaip autobusai ir traukiniai nepatogumai, kaip dažni vėlavymai, griežtai nulemti grafikai, perpildytos kelionės, komplikuotos persėdimų eigos išnyksta. Kelionės yra planuojamos pačio keleivio. Tuo pačiu sprendžiama ir tvarumo problema – transporto dalijimosi sprendimai padeda sumažinti anglies dvideginio emisijas, mažinant transporto priemonių skaičių keliuose, kas prisideda prie tvaresnio vystymosi ir kovos su klimato kaita. „JourneyFy“, tampa kaip bendra platforma, kuri modernizuoja kelionių planavimą ir integruoja technologinius, ekonominius, socialinius ir ekologinius sprendimus į vieningą sistemą, atitinkančią šiuolaikinius naudotojų poreikius. Šiuolaikiniame pasaulyje, modernus transportas – tai raktas į patogesnį, ekologiškesnį ir ekonomiškai tvaresnį gyvenimą.

Darbo tikslas - sukurti šiuolaikišką ir efektyvią kelionių platformą „JourneyFy“, skirtą bendro važiavimo organizavimui bei transporto paslaugų dalijimuisi, kuri padėtų spręsti ekonominius, socialinius ir aplinkosaugos iššūkius, mažinant asmenines kelionių išlaidas ir anglies dvideginio emisijas. Siekiant šio tikslo, numatomi šie uždaviniai:

1. Atlikti esamų rinkoje veikiančių kelionių platformų ir bendro važiavimo (angl. ride-sharing) sistemų apžvalgą bei jų funkcionalumo palyginimą;
2. Sukurti detalią „JourneyFy“ sistemos architektūrą – parengti UML diagramas, suprojektuoti duomenų bazės struktūrą;
3. Įgyvendinti ir ištestuoti „JourneyFy“ pagrindinius sistemos funkcionalumus;
4. Paruošti galutinę naudotojo dokumentaciją.

Darbas susideda iš keturių pagrindinių etapų: analizės, projektavimo, testavimo ir dokumentavimo. Analizėje nustatomi sistemos reikalavimai, apibrėžiami jos tikslai ir pateikiamas jų pagrindimas bei apžvelgiami rinkoje egzistuojantys panašūs sprendimai. Projektavimo etape aprašomi funkcinių ir nefunkcinių reikalavimų detalės, pristatomos naudojamos kūrimo priemonės ir technologijos, taip pat iliustruojami sistemos statiniai ir dinaminiai modeliai. Testavimo skyriuje sudaromas testavimo planas, pristatomi taikomi testų tipai ir pateikiami jų rezultatai. Dokumentacijoje pateikiami naudotojo, diegimo ir administravimo vadovai. Darbo pabaigoje apibendrinami pasiekimai ir pateikiamos išvados.

„JourneyFy“ mobilioji aplikacija sukurta naudojant React Native (Expo Go) ir Firebase. Iš viso sukurta – apie 25 000 kodo eilučių. Čia apjungiama tiek kliento pusės logika (React Native komponentai, navigacija, UI elementai), tiek servisai kaip Firebase.

Pagrindinė projekto struktūra React Native aplikacijai:

- Pagrindiniai projekto UI elementai – ekranai naudojami programėlėje (angl. screens) - apie 10 000 kodo eilučių;
- Pagrindinės Firebase servisų funkcijos, kurios naudojamos programėlėje (angl. hooks) apie 3000 kodo eilučių;
- Pagrindiniai programėlės naudojami UI komponentai (angl. components) apie 10000 kodo eilučių;
- Pagrindinė programėlės navigacija ir navigacijos logika (angl. navigation) apie 1000 kodo eilučių;
- Pagrindiniai servisai (angl. services) apie 200 kodo eilučių;
- Pagrindinis programėlės dizainas ir spalvų paletės (angl. contants) apie 200 kodo eilučių.

1. Analizė

1.1. Techninis pasiūlymas

1.1.1. Sistemos apibrėžimas

„JourneyFy“ – pažangi kelionių tematikos programėlė, leidžianti naudotojams patogiai planuoti tiek kasdienes, tiek tarpmiestines keliones. Vairuotojai čia gali kurti savo siūlomus maršrutus: pasirinkti išvykimo ir atvykimo vietas, nustatyti datą, laiką, laisvų vietų skaičių. Keleiviai, savo ruožtu, gali peržiūrėti visų artimiausių kelionių sąrašą filtruodami pagal savo norimas paėmimo ir atsiėmimo vietas interaktyviame žemėlapyje, taikyti filtrus pagal maršrutą, išvykimo laiką, susisiekti su vairuotoju integruotoje žinučių sistemoje ir vos vienu mygtuko paspaudimu užsiregistruoti kelionei. Sistema siunčia pranešimus realiuoju laiku apie naujus pasiūlymus ir pakeitimus laisvų vietų skaičiuje, leidžia vertinti vairuotojus, o patogi naudotojo sąsaja užtikrina malonią patirtį įvairiomis sąlygomis.

1.1.2. Bendras veiklos tikslas

Bendras veiklos tikslas – sukurti „JourneyFy“ keliavimo kartu platformą, padėsiančią vairuotojams ir keleiviams greitai susirasti bendrakeleivius, suplanuoti maršrutus ir užpildyti laisvas vietas automobilyje. Ši platforma suteiks galimybę naudotojams paprastai derinti kelionės detales realiuoju laiku, dalytis kelionės išlaidomis ir taip mažinti anglies dvideginio emisijas bei skatinti socialinį bendruomeniškumą.

Antrinis tikslas – sukurti vieningą ir prieinamą mobilią platformą, kurioje visi bendro važiavimo pasiūlymai būtų centralizuoti vienoje vietoje, taip atsisakant atskirų Facebook grupių, skelbimų portalų ar neaiškių pokalbių kanalų. Tai leis naudotojams lengvai ieškoti ir pasiūlyti keliones vienoje, aiškioje aplinkoje, užtikrinant greitą paiešką, skaidrią kainodarą ir saugią komunikaciją be trečiųjų šalių tarpininkų. Tokiu būdu „JourneyFy“ taps patikima kasdienio keliavimo kartu platforma.

1.1.3. Sistemos pagrįstumas

Daugeliui žmonių kasdienis susisiekimas tampa tikru iššūkiu: viešojo transporto grafikai ne visuomet dera prie darbo, studijų ar laisvalaikio poreikių, todėl tenka rinktis brangią ir neekonomišką alternatyvą – nuosavą automobilį. Automobilio eksploatacija vienam keleiviui yra neproporcingai brangi dėl kuro, mokesčių, stovėjimo vietos ir priežiūros išlaidų. Be to, miestuose susidariusios spūstys prailgina kelionės laiką, sukelia stresą, o nuolat kylantis taršos lygis kenkia aplinkai ir gyventojų sveikatai.

Nors rinkoje veikia įvairūs kelionių kartu sprendimai, dauguma jų turi keletą esminių trūkumų: nėra vieningos platformos, sujungsiančios vairuotojus ir keleivius skirtingose zonose, maršrutai dažniausiai ribojasi trumpais atstumais ir nėra pritaikyti tolimesnėms kelionėms, be to, jie reikalauja sudėtingo paieškos proceso ir ne visada užtikrina aiškų išlaidų pasidalijimą bei saugumą. Dėl to potencialūs naudotojai negauna patogaus ir greito sprendimo, o transporto sektoriuje lieka neišnaudotos galimybės optimizuoti kaštus, sumažinti spūstis ir anglies dvideginio emisijas.

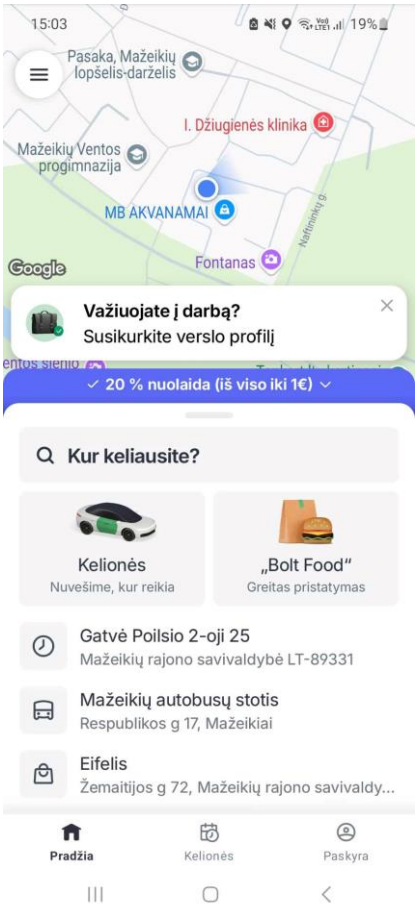
„JourneyFy“ tai vieninga kelionių dalijimosi platforma, kurioje vairuotojas gali akimirksniu paskelbti maršrutą, nurodyti išvykimo laiką ir kainą, o keleiviai – per kelias sekundes surasti ir užsiregistruoti

į kelionę. Sistema palengvina kelionės ir rezervacijos procesus. Tokiu būdu „JourneyFy“ ne tik padeda keliautojams sutaupyti pinigų ir laiko, bet ir reikšmingai prisideda prie anglies dvideginio emisijų mažinimo, mažindama automobilių skaičių keliuose ir skatindama tvarią bendro važiavimo kultūrą.

1.1.4. Egzistuojančių sprendimų analizė

Rinkoje veikia keletas bendro važiavimo ir transporto paslaugų platformų, tokių kaip Bolt, BlaBlaCar, BananaCar. Žemiau pateikiama jų funkcionalumo, naudotojų, kainodaros ir kitų svarbių kriterijų palyginimo santrauka su mūsų kuriamu sprendimu „JourneyFy“.

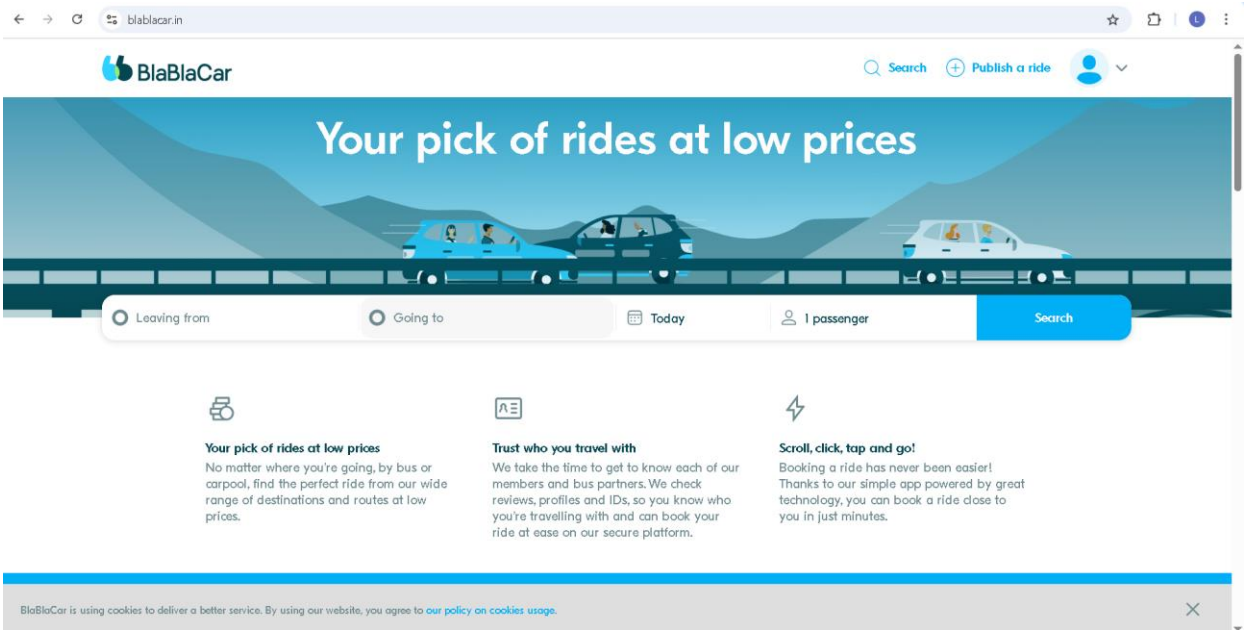
Bolt [2] yra viena iš populiariausių transporto paslaugų platformų, orientuotų į trumpas miesto keliones: vos kelių mygtukų paspaudimu keleivis gali užsisakyti taksi ar elektrinį paspirtuką ir realiu laiku matyti atvykstančio vairuotojo buvimo vietą bei numatomą atvykimo laiką. Dinaminė kainodara leidžia tarifus keisti pagal eismo sąlygas ir paklausą, o programėlė palaiko kelis mokėjimo būdus (banko korteles, mobilias pinigines). Be tradicinio taksi Bolt taip pat siūlo dviračių ir paspirtukų nuomą, integruotą maisto pristatymo paslaugą „Bolt Food“, o saugumą užtikrina keleivių ir vairuotojų vertinimų sistema, SOS funkcija bei draudimas. Dėl šio plataus funkcionalumo Bolt idealiai tinka greitam ir patogiam susisiekimui mieste (žr. 1 pav.).



1 pav. „Bolt“ programėlės lango vaizdas.

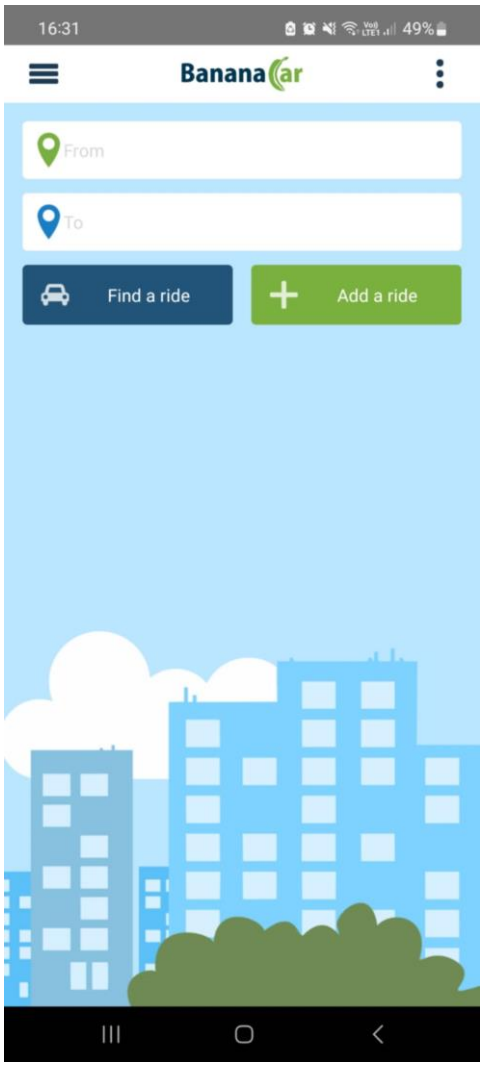
BlaBlaCar [3] yra tarptautinė bendro važiavimo (angl. carpooling) platforma, kuri leidžia vairuotojams skelbti ilgų nuotolių keliones ir dalytis laisvomis vietomis automobilyje, o keleiviams

– rezervuoti vietas. Naudotojai susikuria profilius, nurodo kelionės maršrutą, datą, laiką ir kainą, o kiti keliavimo entuziastai peržiūri siūlomus reisus, skaito vairuotojo atsiliepimus ir vienu paspaudimu užsiregistruoja. Sistema palaiko tiesioginį susirašinėjimą, leidžiantį derinti susitikimo vietą ir kitus kelionės niuansus, o integruota mokėjimo sistema užtikrina, kad kaštai būtų sąžiningai padalijami tarp visų keleivių. Dėl savo didelės bendruomenės – daugiau nei 90 milijonų naudotojų 22 šalyse – ir griežtos naudotojų patikros, BlaBlaCar garantuoja patikimą ir ekonomišką tarpmiestinį keliavimą (žr. 2 pav.).



2 pav. „BlaBlaCar“ sistemos lango vaizdas.

BananaCar [4] yra pakeleivių paieškos platforma. Platforma yra kelionių paieškos sistema. Pagrindiniai šios platformos funkcionalumai yra kelionių paieška, rezervacija, valdymas. Ši programėlė labiausiai atitinka „JourneyFy“ pagal visus galimus funkcionalumus. Ši programėlė yra palaikoma visose platformose (IOS, Android). Tai viena iš sistemų, kuri yra jau įdiegta ir išbandyta Lietuvoje (žr. 3 pav.).



3 pav. „BananaCar“ programėlės lango vaizdas.

Lentelėje **1 lentelė** pateikiamas „JourneyFy“ programėlės palyginimas pagal pasirinktus kriterijus.

1 lentelė Egzistuojančių sprendimų analizė.

Lyginimo kriterijai	**„Bolt“ programėlė**	**„BlaBlaCar“ programėlė**	**„BananaCar“ programėlė**	**Sukurta programėlė „JourneyFy“**
Pokalbių funkcionalumas	+	+	-	+
Žemėlapio integracija	+	+	-	+
Individuali kelionių paieška	-	+	+	+
Kelionės maršruto sudarymas	+	+	-	+
Operacinė sistema	Android, iOS	Android, iOS	Android, iOS	Android (iOS vėliau)
Naudotojų tapatybės ir KYC patikra	+	+	-	-
SOS mygtukas	+	-	-	-
Palaikomų arba aktyvių naudotojų kiekis	>200 mln.	>20 mln.	>1000	Kol kas rinkoje neišleista
Maršrutų atstumas	Trumpų miesto kelionių atstumai	Vidutinio ir ilgo nuotolio tarpmiestiniams reisams	Įvairaus tipo nuotolio atstumai	Įvairaus tipo nuotolio atstumai

Atlikę analizę, palyginome tris platformas – miesto mikrokelionėms skirtą Bolt, ilgų nuotolių kelionių kartu paslaugą BlaBlaCar ir kelionių paieškos platformą BananaCar – su mūsų „JourneyFy“ sprendimu. Bolt pasižymi greitu taksi ar e-paspirtukų užsakymu, dinamine kainodara, keliomis mokėjimo galimybėmis bei saugumo priemonėmis, tačiau tinka tik trumpoms miesto kelionėms. BlaBlaCar sujungia vairuotojus ir keleivius per ilgų maršrutų skelbimus, siūlo atsakingą bendruomenės patikros sistemą ir integruotą mokėjimą, bet platforma labiau orientuota į tarpmiestinį keliavimą. BananaCar pasižymi panašiomis funkcijomis, kaip BlaBlaCar, tačiau šitoje platformoje trūksta pokalbių, žemėlapio, kelionės maršruto sudarymo galimybių. Mūsų „JourneyFy“ siekis – apjungti šių platformų privalumus ir sutvarkyti jų trūkumus: pasiūlyti tiek trumpoms, tiek ilgoms kelionėms tinkančią platformą.

Palyginimo kriterijų paaiškinimai:

- Pokalbių funkcionalumas – pakeleivis gali bendrauti su vairuotoju, uždavinėti jam aktualius klausimus;
- Žemėlapio integracija – naudotojas gali matyti kelionės žemėlapį, juo naudotis;
- Individuali kelionių paieška – naudotojas gali individualiai ieškoti kelionės, kuri jam tinkama;
- Kelionės maršruto sudarymas – vairuotojui sudaromas kelionės maršrutas, kai kelionė yra pradedama arba ją planuojama atlikti;
- Operacinė sistema – programėlės palaikomos operacinės sistemos;
- Naudotojų tapatybės ir KYC patikra – užtikrinimas keleivių saugumas, vairuotojai privalo patvirtinti savo tapatybę ir praeinama mašinos patikra;
- SOS mygtukas – atsitikus nelaimei yra galimybė išsikviesti pagalbą kelyje;

- Palaikomų arba aktyvių naudotojų kiekis – taip sudaromas vaizdas, kiek programėlėje yra aktyvių naudotojų;
- Maršrutų atstumas – kuris kelionių atstumas yra populiarus arba dažniausiai naudojamas programėlėse.

Šie kriterijai buvo pasirinkti, nes geriausiai pavaizduoja esančius trūkumus keliavimo kartu programėlių rinkoje. Daugumoje sprendimų trūksta įvairaus funkcionalumo, kuris prisidėtų prie platformos apjungimo ir centralizavimo.

Galime teigti, kad artimiausi egzistuojantys sprendimai „JourneyFy“ platformai yra BlaBlaCar ir BananaCar. BlaBlaCar siūlo panašias keliavimo kartu paslaugas, tačiau dažniausiai orientuojasi tik į vieną kelionių sritį (miesto arba tarpmiestinius reisus). Bolt programėlė yra labiau orientuotą į platesnę transporto paslaugų sferą, ilgesnės kelionės tampa per brangios. BananaCar platforma pasiūlo naudotojui per mažai funkcionalumo. „JourneyFy“ kelionių platforma išsiskiria iš visų savo įvairaus tipo atstumo kelionėmis ir papildomu funkcionalumu.

1.1.5. Prototipai ir pagalbinė informacija

„JourneyFy“ programėlės naudotojo sąsajai, buvo panaudotas „GoPoolar Car APP UI“ [5] dizaino šablonas, iš kurio perimta ir pritaikyta spalvų paletė, komponentai bei navigacijos logika ir nuoseklumas. Šie komponentai buvo pritaikyti pagal programėlės poreikius. Šis UI/UX dizaino šablonas leido sukurti vientisą ir modernų dizainą.

Kuriant „JourneyFy“ programėlės naudotojo sąsają, nuo pat pradžių siekta sukurti ne tik vizualiai patrauklią, bet ir funkcionaliai tvarią, sklandžią bei intuityvią naudotojo patirtį.

Įkvėpimo buvo imtasi ir iš BlaBlaCar. Remiantis jau egzistuojančia sistema buvo paimtas pavyzdys iš kelionių paieškos ir užklausų logikos, pritaikytas naudotojų atsiliepimų funkcionalumas. BlaBlaCar padėjo sudaryti bendrą vystymo planą visiem kitiem „JourneyFy“ programėlės funkcionalumam. Reikėjo vienos unikalios kelionių platformos vairuotojam ir keleiviam Lietuvoje. Tai panaikina kitų alternatyvų prasmę: Facebook ar kitų socialinių medijų alternatyvos. Socialinėse medijose yra sudaromas tam tikras nepatogumas organizuoti keliones (dažniausiai už jas pilnai atsakingas vairuotojas). Nėra unikalaus sprendimo, kaip keliones rezervuoti arba atšaukti. Šie dideli nepatogumai ir sudarė visą „JourneyFy“ platformos vystymosi pradžią.

1.2. Techninių galimybių analizė

„JourneyFy“ mobilioji aplikacija sukurta remiantis JavaScript/TypeScript programavimo kalbomis: kliento pusė – React Native karkasas, paleidžiamas per Expo Go, o visos duomenų bazės funkcijos – Firebase serviso paslaugos (Firestore duomenų bazė, Authentication autentifikacijai, Storage). Toks sprendimas leidžia naudoti tą pačią kalbą (JavaScript/TypeScript) ir UI komponentams, ir serverio logikai, todėl nebereikia atskiros administracinės sistemos ar sudėtingo serverių valdymo. Expo Go suteikia greitą „hot reload“ keitimų peržiūrą mobiliuosiuose įrenginiuose bei galimybę vykdyti Over-The-Air atnaujinimus, tad naujos funkcijos pasiekiamos akimirksniu. Firestore automatiškai sinchronizuoja duomenis realiu laiku ir palaiko offline režimą. Firebase autentifikacijos integracija užtikrina paprastą, saugų prisijungimą (el. pašto šablonas), o Firebase saugos taisyklės leidžia valdyti duomenų prieigą. Be to, Firebase automatiškai didina paslaugų kaštus augant naudotojų skaičiui, todėl infrastruktūra neapsunkina vystymo darbo. Nors React Native programėlė naudoja šiek tiek daugiau įrenginio atminties ir procesoriaus nei gimtąja kalba rašytos aplikacijos, o senesnių Android/iOS versijų suderinamumas kartais reikalauja papildomų pataisymų, šie niuansai paprastai išsprendžiami optimizacijomis ir NPM bibliotekų atnaujinimais.

2. Projektas

2.1. Reikalavimų specifikacija

2.1.1. Komercinė specifikacija

„JourneyFy“ projekto užsakovas – transporto rinkoje dirbantis studentas, kuris nagrinėja kelionių autostopu problemą. „JourneyFy“ projekto vykdytojas – KTU ketvirto kurso studentas. Numatyta projekto pristatymo data – 2025 m. balandžio 20 d. Iki to laiko sistema turi būti visiškai suprogramuota, ištestuota, įdiegta lokalioje aplinkoje ir parengta naudoti. Komercializavimo galimybių ratas – platus: „JourneyFy“ gali tapti tarpininku tarp vairuotojo ir keleivio, imdamas nedidelį komisinį mokestį nuo kiekvienos sėkmingos kelionės (pvz., 5 %–10 % nuo bilieto kainos). Be to, galima pasiūlyti prenumeratos modelį aktyviems naudotojams – mėnesinį arba metinį planą su mažesniais mokesčiais už paslaugą, VIP palaikymu ar išplėstomis paieškos filtravimo galimybėmis. Reklamos ir partnerystės programos su degalinių, kavinių ar stovėjimo aikštelių tinklais leistų papildomai generuoti pajamas. Ateityje – bendradarbiavimas su transporto draudimo bendrovėmis, siūlant kelionei draudimą tiesiogiai per programėlę. Tokiu būdu „JourneyFy“ užtikrintų tvarią verslo modelio plėtrą ir platesnį naudotojų pasitenkinimą.

2.1.2. Sistemos funkcijos

„JourneyFy“ sistemos funkcinius reikalavimus iliustruojame UML panaudojimo diagramomis ir lentelėmis, 4 pav. pateikiama visos sistemos panaudojimo atvejų diagrama.

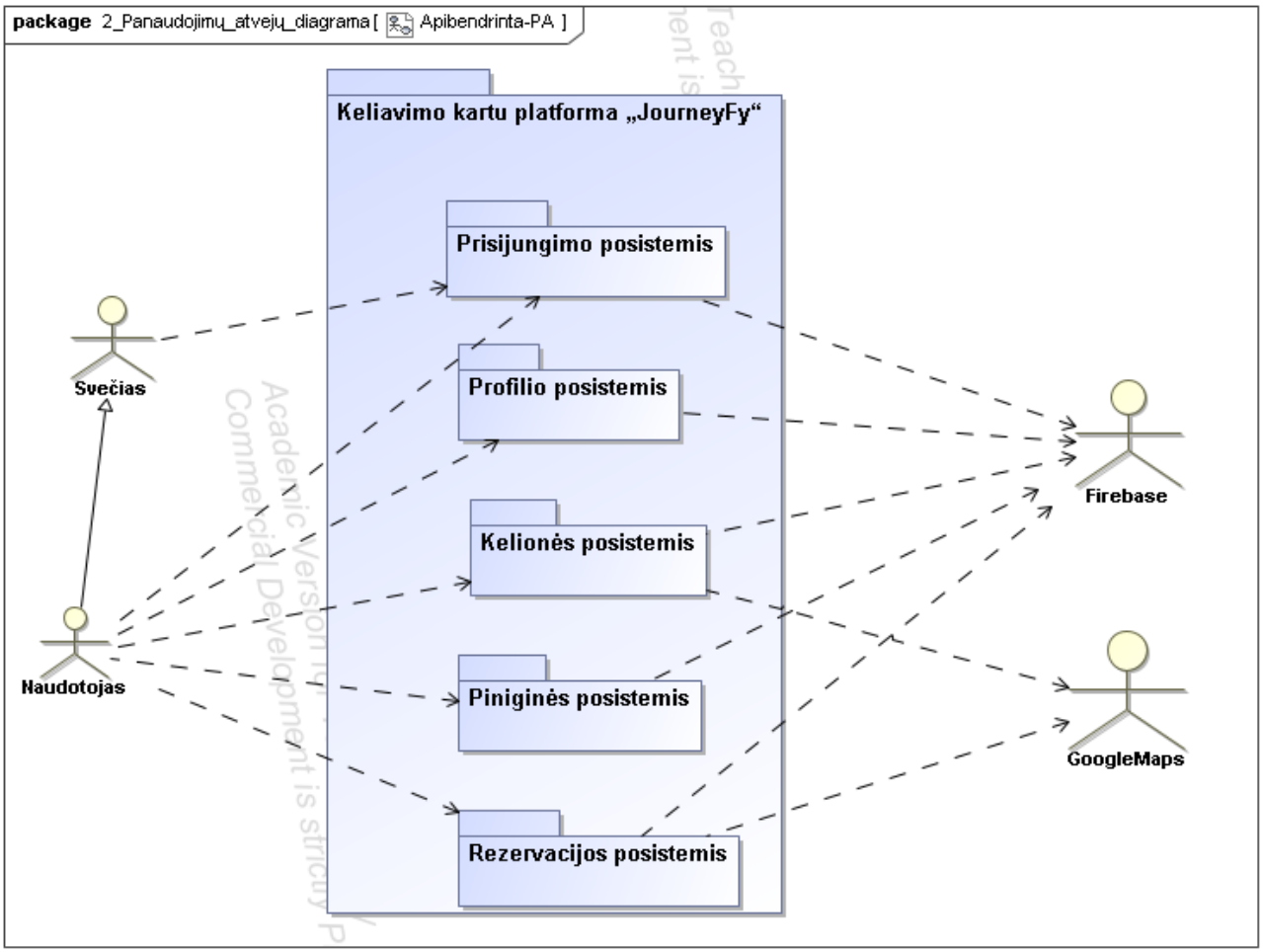
Šie funkcionalumai suskirstyti į kelias posistemes:

- Prisijungimo posistemis (posistemis atsakingas už svečio bei naudotojo registraciją, prisijungimą, slaptažodžio priminimą);
- Profilio posistemis (posistemis atsakingas už naudotojo profilio, mašinos, kelionių istorijos funkcionalumą, nustatymų funkcionalumą);
- Kelionių posistemis (posistemis atsakingas už naudotojų kelionių peržiūrą, kelionių pradžią ir užbaigimą);
- Piniginės posistemis (posistemis atsakingas už piniginės valdymą – naudotojo piniginės balansą);
- Rezervacijos posistemis (posistemis atsakingas už kelionių rezervaciją ir kelionių kūrimą bei paiešką).

Pagrindiniai sistemos aktoriai:

- Svečias – pradinis sistemos naudotojas, kuris gali registruotis;
- Naudotojas – jau registruotas naudotojas, kuris paveldi visus svečio funkcionalumus, tuo pačiu gali atlikti visus kitų posistemių funkcionalumus.

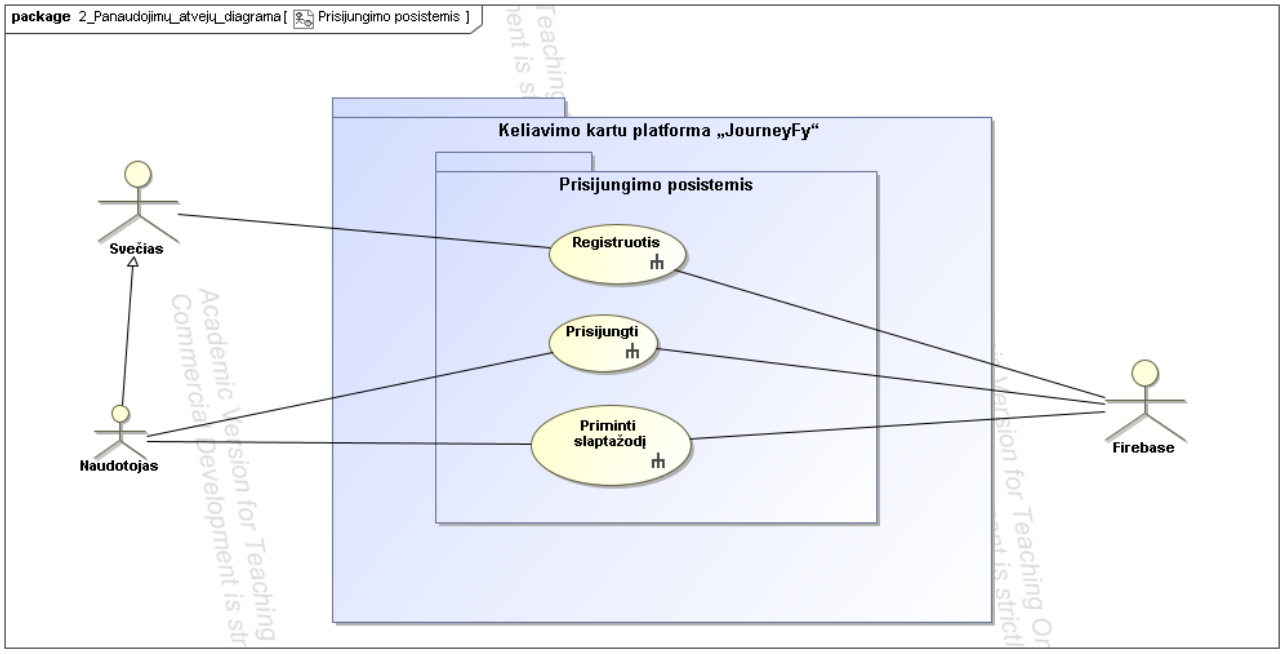
Administratoriaus posistemis egzistuoja, tačiau jis valdoma debesyje (Firebase Admin Console sąsaja). Tai administratoriaus, kaip sistemos aktoriaus nevaizduojame.



4 pav. Apibendrinta „JourneyFy“ panaudojimo atvejų diagrama.

Prisijungimo posistemis

Prisijungimo posistemis yra skirtas pagrindiniams autentifikacijos funkcionalumams. Čia svečias arba naudotojas gali susikurti paskyrą prisiregistruodamas prie sistemos, prisijungti prie sistemos ir prisiminti savo slaptažodį (žr. 5 pav.).



5 pav. Prisijungimo posistemio diagrama.

PA „Prisijungti“ specifikacija pateikiama lentelėje **2 lentelė**. Naudotojas norėdamas prisijungti ir pradėtis naudotis programėle privalo prisijungti su savo registruotu el. paštu (kuris privalo būti patvirtintas) ir slaptažodžiu.

2 lentelė „Prisijungti“ PA specifikacija.

PA. Prisijungti		
Tikslas. Leisti naudotojui prisijungti prie sistemos „JourneyFy“		
Aprašymas. Registruotas programėlės naudotojas gali prisijungti prie sistemos, atliekamas prisijungimo veiksmas – užpildomas registruoto el. pašto ir slaptažodžio laukai. Norėdamas pilnai prisijungti prie sistemos naudotojas privalo būti patvirtinęs savo paskyros el. paštą iš gautos sisteminės žinutės.		
Prieš sąlyga	Naudotojas neprisijungęs prie sistemos	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojas prijungiamas prie sistemos	

PA „Registruotis“ specifikacija pateikiama lentelėje **3 lentelė**. Naudotojas arba svečias, kuris nori susikurti paskyrą, privalo užpildyti formą su savo el. paštu ir norimu slaptažodžiu.

3 lentelė „Registruotis“ PA specifikacija.

PA. Registruotis		
Tikslas. Leisti svečiui prisiregistruoti prie sistemos		
Aprašymas. Svečias kuris nori pradėti naudotis programėle, turi pilnai atlikti registracijos veiksmą – užpildyti savo naujos paskyros el. paštą ir slaptažodį. Po to svečias privalo patvirtinti el. paštą su gauta užklausa, kad galėtų sėkmingai prisijungti prie sistemos.		
Prieš sąlyga	Svečias neprisijungęs prie sistemos	
Aktoriai	Svečias	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Sukuriama naudotojo paskyra duomenų bazėje ir išsiunčiama el. pašto patvirtinimo užklausa.	

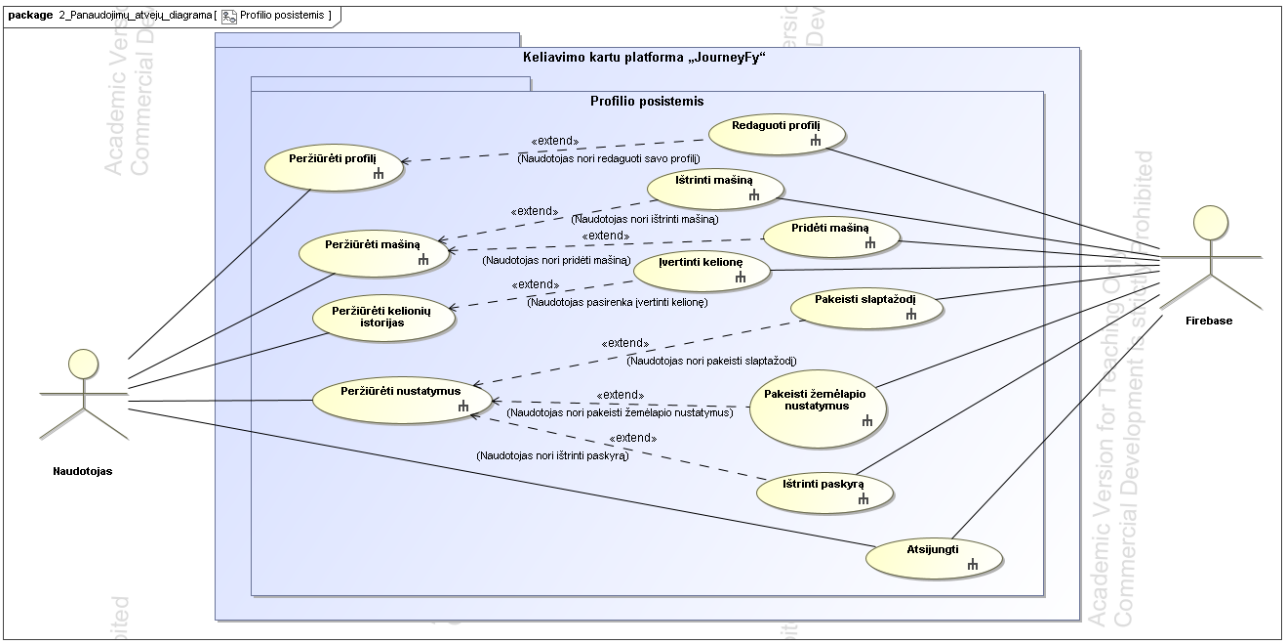
PA „Priminti slaptažodį“ specifikacija pateikiama lentelėje **4 lentelė**. Naudotojas, kuris užmiršo savo slaptažodį, gali pateikti ir gauti slaptažodžio keitimo užklausą į savo registruotą el. paštą.

4 lentelė „Priminti slaptažodį“ PA specifikacija.

PA. Priminti slaptažodį		
Tikslas. Leisti naudotojui pasikeisti savo slaptažodį, jeigu jis negali prisijungti prie sistemos, bet žino savo el. paštą ir turi galimybę ten peržiūrėti laiškus.		
Aprašymas. Naudotojas norintis prisiminti savo slaptažodį privalo atlikti slaptažodžio priminimo veiksmą – užpildomas registruotos paskyros el. paštas. Po šio užpildymo naudotojas gauną slaptažodžio keitimo užklausą į savo registruotą el. paštą. Užpildydami ir pateikdami užklausą galime pasikeisti savo slaptažodį.		
Prieš sąlyga	Naudotojas neprisijungęs prie sistemos	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojas gauna slaptažodžio pakeitimo užklausą į savo registruotos paskyros el. paštą	

Profilio posistemis

Profilio posistemis yra skirtas pagrindiniams naudotojo profilio funkcionalumams. Čia naudotojas gali peržiūrėti ir redaguoti savo profilio informaciją, pridėti arba ištrinti mašiną, peržiūrėti savo kelionių istorijas bei keisti nustatymus ir atsijungti (žr. 6 pav.).



6 pav. Profilio posistemio diagrama.

PA „Peržiūrėti profilį“ specifikacija pateikiama lentelėje **5 lentelė**. Naudotojas gali peržiūrėti savo pagrindinę profilio informaciją.

5 lentelė „Peržiūrėti profilį“ PA specifikacija.

PA. Peržiūrėti profilį		
Tikslas. Naudotojas gali peržiūrėti savo pagrindinę profilio informaciją, kaip vardas, pavardė, telefono numeris ir profilio nuotrauka.		
Aprašymas. Naudotojas, kuris nori peržiūrėti savo profilį gali paspausti ant profilio skilties mygtuko. Vėliau užkraunama šito naudotojo profilio informacija, kaip vardas, pavardė, telefono numeris.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra profilio skiltyje.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	Redaguoti profilį
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojas mato savo profilio informacijos langą	

PA „Redaguoti profilį“ specifikacija pateikiama lentelėje **6 lentelė**. Naudotojas gali redaguoti savo profilio informaciją.

6 lentelė „Redaguoti profilį“ PA specifikacija.

PA. Redaguoti profilį		
Tikslas. Naudotojas gali redaguoti savo pagrindinę profilio informaciją, kaip vardas, pavardė, telefono numeris bei profilio nuotrauka. Jeigu naudotojas yra naujai registruotas – jis vieną kartą gali užpildyti savo gimimo datą, vėliau šito lauko koreguoti nėra leidžiama.		
Aprašymas. Naudotojas, kuris yra profilio peržiūros lange gali redaguoti savo pagrindinę profilio informaciją, kaip vardas, pavardė, telefono numeris ir profilio nuotrauka. Po šio redagavimo atnaujinami profilio duomenys duomenų bazėje.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra profilio peržiūros lange.	
Aktoriai	Naudotojas.	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojo profilio informacija sėkmingai atnaujinama duomenų bazėje ir pateikiamas sėkmės pranešimas.	

PA „Peržiūrėti mašiną“ specifikacija pateikiama lentelėje **7 lentelė**. Naudotojas gali peržiūrėti savo paskyros mašiną.

7 lentelė „Peržiūrėti mašiną“ PA specifikacija.

PA. Peržiūrėti mašiną		
Tikslas. Naudotojas gali peržiūrėti savo registruotą mašiną sistemoje.		
Aprašymas. Naudotojas kuris yra profilio skiltyje gali peržiūrėti savo mašiną, paspausdamas ant mašinos navigacijos skilties. Kai naudotojas naviguoja link mašinos dalies yra užkraunama jo pridėta mašina iš duomenų bazės.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra profilio skiltyje.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	Ištrinti mašiną Pridėti mašiną
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojas mato savo mašinos peržiūros langą.	

PA „Ištrinti mašiną“ specifikacija pateikiama lentelėje **8 lentelė**. Naudotojas gali ištrinti savo mašiną iš paskyros.

8 lentelė „Ištrinti mašiną“ PA specifikacija.

PA. Ištrinti mašiną		
Tikslas. Naudotojas gali ištrinti savo pridėtą mašiną iš duomenų bazės.		
Aprašymas. Naudotojas, kuris yra mašinos peržiūros lange gali ištrinti savo mašiną.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra mašinos peržiūros lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Ištrinama naudotojo mašina iš duomenų bazės ir pateikiamas sėkmės pranešimas.	

PA „Pridėti mašiną“ specifikacija pateikiama lentelėje **9 lentelė**. Naudotojas gali pridėti mašiną prie savo paskyros, kuri privalo būti patvirtinta administratoriaus.

9 lentelė „Pridėti mašiną“ PA specifikacija.

PA. Pridėti mašiną		
Tikslas. Naudotojas gali pridėti vieną mašiną prie savo paskyros, kad galėtų sėkmingai kurti keliones.		
Aprašymas. Naudotojas, kuris yra mašinos peržiūros lange, gali pridėti mašiną paspausdamas pridėjimo mygtuką ir užpildydamas mašinos pridėjimo formą. Šiuo metu, dėl mašinos patvirtinimo problemų palaikomas tik vienos mašinos pridėjimas vienam naudotojui.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra mašinos peržiūros lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojo mašina pridedama prie duomenų bazės ir yra pateikiamas sėkmės pranešimas, perkraunamas mašinos peržiūros langas.	

PA „Peržiūrėti kelionių istorijas“ specifikacija pateikiama lentelėje **10 lentelė**. Naudotojas gali peržiūrėti savo buvusių ir užbaigtų kelionių aprašymus.

10 lentelė „Peržiūrėti kelionių istorijas“ PA specifikacija.

PA. Peržiūrėti kelionių istorijas		
Tikslas. Naudotojas gali peržiūrėti savo buvusių kelionių istorijas.		
Aprašymas. Naudotojas, kuris yra kelionių istorijų lange, gali pasirinkti vieną iš kelionių ir peržiūrėti informaciją apie šią kelionę kaip keleivis arba vairuotojas.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra kelionių istorijų lange bei pasirenka vieną iš buvusių kelionių.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	Įvertinti kelionę
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojui rodomas buvusios kelionės peržiūros langas.	

PA „Įvertinti kelionę“ specifikacija pateikiama lentelėje **11 lentelė**. Naudotojas gali įvertinti savo buvusią kelionę.

11 lentelė „Įvertinti kelionę“ PA specifikacija.

PA. Įvertinti kelionę		
Tikslas. Naudotojas gali įvertinti savo buvusią kelionę. Taip kitiems naudotojams sudaromas įvaizdis apie kelionės vairuotoją.		
Aprašymas. Naudotojas, kuris yra buvusios kelionės peržiūros lange gali įvertinti savo kelionę nuo 1-5 žvaigždučių bei pateikdamas komentarą. Pačio vairuotojo įvertinimai sau nesiskaito.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra buvusios kelionės peržiūros lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojas užpildo įvertinimo formą, kuri išsaugoma ir pateikiamas sėkmės pranešimas.	

PA „Peržiūrėti nustatymus“ specifikacija pateikiama lentelėje **12 lentelė**. Naudotojas gali peržiūrėti savo nustatymus.

12 lentelė „Peržiūrėti nustatymus“ PA specifikacija.

PA. Peržiūrėti nustatymus		
Tikslas. Naudotojas gali peržiūrėti nustatymų dalies langą, kur gali koreguoti savo nustatymus.		
Aprašymas. Naudotojas, kuris yra profilio lange gali pasirinkti nustatymų dalies skiltį ir peržiūrėti visus nustatymus.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra profilio lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	Pakeisti slaptažodį Pakeisti žemėlapio nustatymus Ištrinti paskyrą
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojui yra užkraunamas nustatymų peržiūros langas.	

PA „Pakeisti slaptažodį“ specifikacija pateikiama lentelėje **13 lentelė**. Naudotojas gali pakeisti savo paskyros slaptažodį.

13 lentelė „Pakeisti slaptažodį“ PA specifikacija.

PA. Pakeisti slaptažodį		
Tikslas. Naudotojas gali pakeisti savo slaptažodį, kuris naudojamas norint prisijungti prie programėlės.		
Aprašymas. Naudotojas, kuris yra nustatymų peržiūros lange gali pasirinkti slaptažodžio pakeitimo skiltį norėdamas pasikeisti savo paskyros slaptažodį. Yra užpildoma slaptažodžio pakeitimo forma, šie slaptažodžiai privalo būti vienodi.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra nustatymų peržiūros lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojo slaptažodis pakeičiamas duomenų bazėje ir pateikiamas sėkmės pranešimas.	

PA „Pakeisti žemėlapio nustatymus“ specifikacija pateikiama lentelėje **14 lentelė**. Naudotojas gali pakeisti savo žemėlapio nustatymus.

14 lentelė „Pakeisti žemėlapio nustatymus“ PA specifikacija.

PA. Pakeisti žemėlapio nustatymus		
Tikslas. Naudotojas gali pakeisti savo žemėlapio nustatymus.		
Aprašymas. Naudotojas, kuris yra nustatymų peržiūros lange gali pasirinkti žemėlapio nustatymų skiltį ir pakeisti pagrindinius žemėlapio nustatymus kaip sustojimų skaičius ir paskirties vietos spindulys. Šios reikšmės padės geriau ieškoti kelionių ateityje.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra nustatymų peržiūros lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojas pakoreguoja savo žemėlapio nustatymus ir jie atnaujinami duomenų bazėje, pateikiamas sėkmės pranešimas.	

PA „Ištrinti paskyrą“ specifikacija pateikiama lentelėje **15 lentelė**. Naudotojas gali ištrinti savo paskyrą.

15 lentelė „Ištrinti paskyrą“ PA specifikacija.

PA. Ištrinti paskyrą		
Tikslas. Naudotojas gali ištrinti savo paskyrą.		
Aprašymas. Naudotojas, kuris yra nustatymų peržiūros lange ir pasirenka paskyros ištrinimo skiltį, gali ištrinti savo paskyrą visam laikui. Tačiau tam reikalingas mygtuko patvirtinimas. Po paskyros ištrinimo jau esantis naudotojas tampa svečiu ir grįžtą į prisijungimo langą.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra nustatymų peržiūros lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Ištrinamas naudotojas iš duomenų bazės ir yra grįžtamą į prisijungimo langą.	

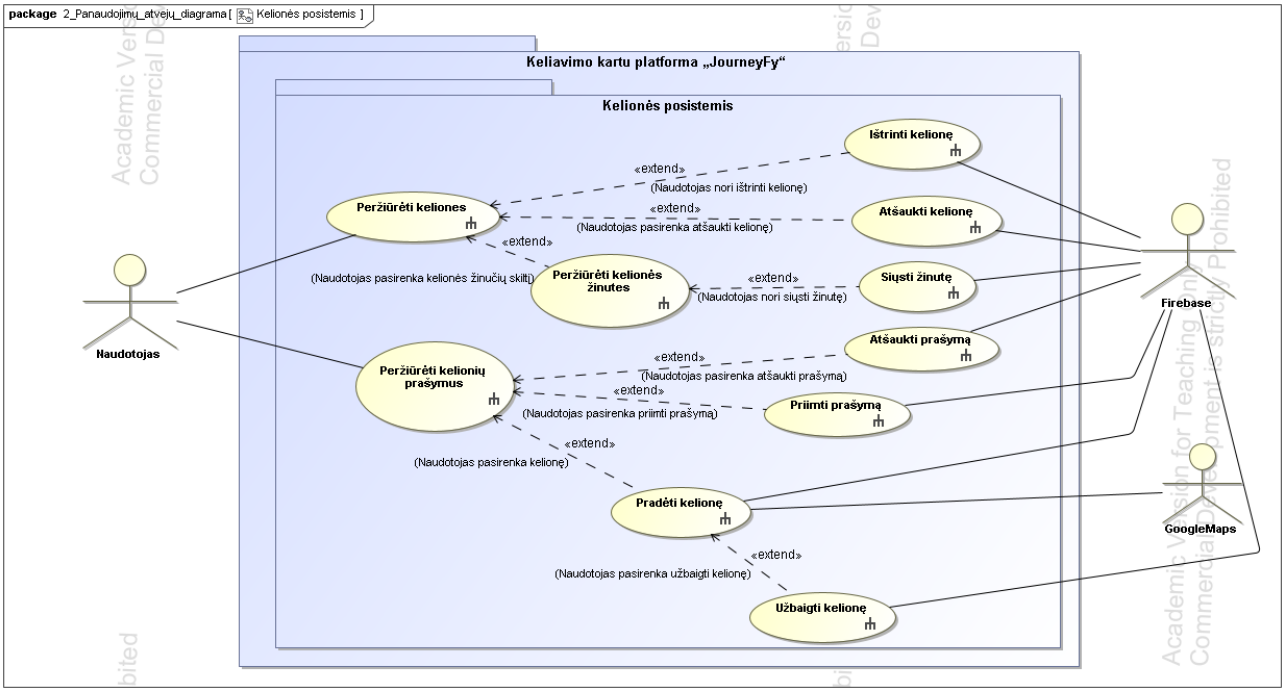
PA „Atsijungti“ specifikacija pateikiama lentelėje **16 lentelė**. Naudotojas gali atsijungti nuo savo paskyros.

16 lentelė „Atsijungti“ PA specifikacija.

<table>
<tr><td colspan="3">PA. Atsijungti</td></tr>
<tr><td colspan="3">Tikslas. Naudotojas gali atsijungti iš sistemos.</td></tr>
<tr><td colspan="3">Aprašymas. Naudotojas, kuris yra profilio lange ir pasirenka atsijungti skiltį, gali su mygtuko patvirtinimu atsijungti nuo programėlės. Po atsijungimo grįžtamą į prisijungimo langą.</td></tr>
<tr><td colspan="2">Prieš sąlyga</td><td>Naudotojas yra prisijungęs ir yra profilio lange.</td></tr>
<tr><td colspan="2">Aktoriai</td><td>Naudotojas</td></tr>
<tr><td rowspan="3">Susiję panaudojimo atvejai</td><td>Išplečiantys PA</td><td></td></tr>
<tr><td>Apimami PA</td><td></td></tr>
<tr><td>Specializuojami PA</td><td></td></tr>
<tr><td colspan="2">Po sąlyga</td><td>Naudotojas yra atjungiamas nuo sistemos ir yra grįžtamą į prisijungimo langą.</td></tr>
</table>

Kelionės posistemis.

Kelionės posistemis yra skirtas pagrindiniams naudotojo kelionių tematikos funkcionalumams. Čia naudotojas gali peržiūrėti savo būsimų arba pridėtų kelionių aprašymus, bendrauti žinutėmis su kitais, priimti savo kelionių prašymus bei juos atmesti ir pradėti arba užbaigti keliones (žr. 7 pav.).



7 pav. Kelionės posistemio diagrama.

PA „Peržiūrėti keliones“ specifikacija pateikiama lentelėje **17 lentelė**. Naudotojas gali peržiūrėti savo pridėtas arba registruotas keliones.

17 lentelė „Peržiūrėti keliones“ PA specifikacija.

PA. Peržiūrėti keliones		
Tikslas. Naudotojas gali peržiūrėti savo keliones, kaip keleivis arba vairuotojas.		
Aprašymas. Naudotojas, kuris yra kelionių lange gali pasirinkti vieną iš kelionių savo sąraše ir peržiūrėti ją. Čia rodomos visos kelionės, prie kurių naudotojas yra prisiregistravęs kaip keleivis arba yra pačios kelionės vairuotojas.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra kelionių peržiūros lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	Ištrinti kelionę Atšaukti kelionę Peržiūrėti kelionės žinutes
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojui yra užkraunamas kelionių peržiūros langas.	

PA „Ištrinti kelionę“ specifikacija pateikiama lentelėje **18 lentelė**. Naudotojas gali ištrinti savo pridėtą kelionę.

18 lentelė „Ištrinti kelionę“ PA specifikacija.

PA. Ištrinti kelionę		
Tikslas. Naudotojas gali ištrinti savo sukurtą kelionę.		
Aprašymas. Naudotojas, kuris yra kelionių peržiūros lange gali ištrinti savo kelionę, jei jis yra šios kelionės vairuotojas.		
Prieš sąlyga	Naudotojas yra prisijungęs bei yra kelionės vairuotojas ir yra kelionių peržiūros lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Ištrinama naudotojo kelionė iš duomenų bazės ir pateikiamas sėkmės pranešimas.	

PA „Atšaukti kelionę“ specifikacija pateikiama lentelėje **19 lentelė**. Naudotojas gali atšaukti savo kelionės rezervacijos prašymą.

19 lentelė „Atšaukti kelionę“ PA specifikacija.

PA. Atšaukti kelionę		
Tikslas. Naudotojas gali atšaukti savo registraciją.		
Aprašymas. Naudotojas, kuris yra prisiregistravęs gali atšaukti savo registraciją nuo kelionės.		
Prieš sąlyga	Naudotojas yra prisijungęs bei yra prisiregistravęs prie kelionės ir yra kelionės peržiūros lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Atšaukiama naudotojo registracija nuo kelionės ir pateikiamas sėkmės pranešimas.	

PA „Peržiūrėti kelionės žinutes“ specifikacija pateikiama lentelėje **20 lentelė**. Naudotojas gali peržiūrėti kelionės pokalbio žinutes.

20 lentelė „Peržiūrėti kelionės žinutes“ PA specifikacija.

PA. Peržiūrėti kelionės žinutes		
Tikslas. Naudotojas gali peržiūrėti kelionės žinutes.		
Aprašymas. Naudotojas, kuris yra kelionių peržiūros lange gali pasirinkti žinučių skiltį ir peržiūrėti kelionės pokalbių langą.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra kelionės peržiūros lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	Siųsti žinutę
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Užkraunamas kelionės žinučių peržiūros langas.	

PA „Siųsti žinutę“ specifikacija pateikiama lentelėje **21 lentelė**. Naudotojas gali siųsti žinutes kelionės pokalbio lange.

21 lentelė „Siųsti žinutę“ PA specifikacija.

PA. Siųsti žinutę		
Tikslas. Naudotojas gali siųsti žinutes, žinučių peržiūros lange.		
Aprašymas. Naudotojas, kuris yra žinučių peržiūros lange, gali bendrauti su kitais naudotojais siųsdamas žinutes. Matomas laikas kada žinutė yra išsiųsta ir būtent kokio naudotojo - nuotrauka ir vardas bei pavardė.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra žinučių peržiūros lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojas išsiunčią žinutę, kuri atnaujinama ir matoma žinučių peržiūros lange.	

PA „Peržiūrėti kelionių prašymus“ specifikacija pateikiama lentelėje **22 lentelė**. Naudotojas gali peržiūrėti savo pridėtų kelionių rezervacijos prašymus.

22 lentelė „Peržiūrėti kelionių prašymus“ PA specifikacija.

PA. Peržiūrėti kelionių prašymus		
Tikslas. Naudotojas gali peržiūrėti savo kelionių keleivių prašymus		
Aprašymas. Naudotojas, kuris yra kelionių lange gali pasirinkti prašymų skiltį ir peržiūrėti visus kelionių prašymus.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra kelionių lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	Priimti prašymą Atšaukti prašymą Pradėti kelionę
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojui užkraunamas kelionių prašymų peržiūros langas.	

PA „Priimti prašymą“ specifikacija pateikiama lentelėje **23 lentelė**. Naudotojas gali priimti keleivio rezervacijos prašymą.

23 lentelė „Priimti prašymą“ PA specifikacija.

PA. Priimti prašymą		
Tikslas. Naudotojas gali priimti keleivio rezervacijos prašymą		
Aprašymas. Naudotojas, kuris yra kelionių prašymų peržiūros lange gali priimti keleivių rezervacijos prašymus.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra kelionių prašymų peržiūros lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Keleivis yra pridedamas prie kelionės ir pateikiamas sėkmės pranešimas	

PA „Atšaukti prašymą“ specifikacija pateikiama lentelėje **24 lentelė**. Naudotojas gali atšaukti keleivio rezervacijos prašymą.

24 lentelė „Atšaukti prašymą“ PA specifikacija.

PA. Atšaukti prašymą		
Tikslas. Naudotojas gali atšaukti keleivio rezervacijos prašymą		
Aprašymas. Naudotojas, kuris yra kelionių prašymų peržiūros lange gali atšaukti keleivių rezervacijos prašymus.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra kelionių prašymų peržiūros lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Keleivio rezervacijos prašymas yra atšaukiamas ir pateikiamas sėkmės pranešimas.	

PA „Pradėti kelionę“ specifikacija pateikiama lentelėje **25 lentelė**. Naudotojas gali pradėti savo kelionę.

25 lentelė „Pradėti kelionę“ PA specifikacija.

PA. Pradėti kelionę		
Tikslas. Naudotojas gali pradėti kelionę.		
Aprašymas. Naudotojas pasirinkęs kelionę ir paspaudęs pradėjimo mygtuką, gali pradėti kelionę. Naudotojui yra sudaromas kelionės maršrutas.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra pasirinkęs kelionę iš kelionių prašymo peržiūros lango.	
Aktoriai	Naudotojas, Google Maps API	
Susiję panaudojimo atvejai	**Išplečiantys PA**	Užbaigti kelionę
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojas pradeda kelionę ir jam sudaromas kelionės maršrutas. Kelionės statusas atnaujinamas duomenų bazėje, parodomas sėkmės pranešimas.	

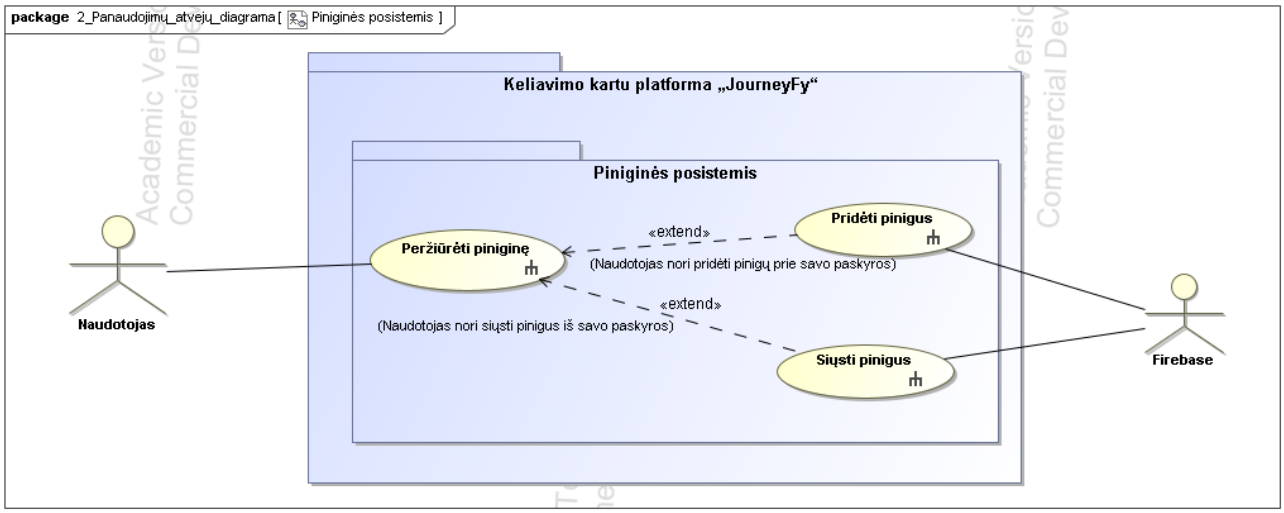
PA „Užbaigti kelionę“ specifikacija pateikiama lentelėje **26 lentelė**. Naudotojas gali užbaigti savo pradėtą kelionę.

26 lentelė „Užbaigti kelionę“ PA specifikacija.

PA. Užbaigti kelionę		
Tikslas. Naudotojas gali užbaigti jau pradėtą kelionę		
Aprašymas. Naudotojas, kuris yra kelionės pradžios lange, gali užbaigti šią kelionę.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra kelionės pradžios lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Kelionė užbaigiama, atnaujinamas šios kelionės statusas, parodomas sėkmės pranešimas.	

Piniginės posistemis

Piniginės posistemis yra skirtas naudotojo piniginės balanso valdymui – galima siųsti ir pridėti pinigus prie savo piniginės balanso (žr. 8 pav.).



8 pav. Piniginės posistemio diagrama.

PA „Peržiūrėti piniginę“ specifikacija pateikiama lentelėje **27 lentelė**. Naudotojas gali peržiūrėti savo piniginės balansą.

27 lentelė „Peržiūrėti piniginę“ PA specifikacija.

PA. Peržiūrėti piniginę		
Tikslas. Naudotojas gali peržiūrėti savo piniginę		
Aprašymas. Naudotojas gali peržiūrėti savo piniginės likutį.		
Prieš sąlyga	Naudotojas yra prisijungęs	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	Pridėti pinigus Siųsti pinigus
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojui rodomas piniginės peržiūros langas.	

PA „Pridėti pinigus“ specifikacija pateikiama lentelėje **28 lentelė**. Naudotojas gali pridėti pinigų prie savo piniginės balanso.

28 lentelė „Pridėti pinigus“ PA specifikacija.

PA. Pridėti pinigus		
Tikslas. Naudotojas gali pridėti pinigų prie savo piniginės		
Aprašymas. Naudotojas, esantis piniginės peržiūros lange gali pasirinkti pinigų pridėjimo skiltį ir užpildydamas visas reikalingas formas gali pridėti pinigų prie savo piniginės.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra piniginės peržiūros lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojo piniginės likutis atnaujinamas ir pateikiamas sėkmės pranešimas.	

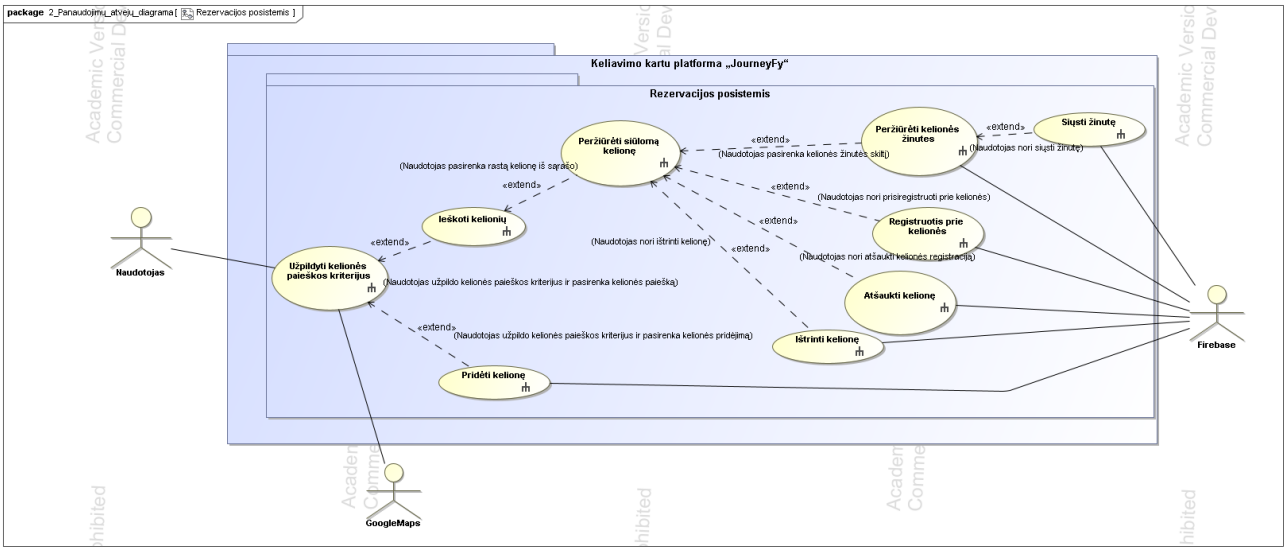
PA „Siųsti pinigus“ specifikacija pateikiama lentelėje **29 lentelė**. Naudotojas gali užbaigti savo pradėtą kelionę.

29 lentelė „Siųsti pinigus“ PA specifikacija.

PA. Siųsti pinigus		
Tikslas. Naudotojas gali siųsti pinigus iš savo piniginės		
Aprašymas. Naudotojas, esantis piniginės peržiūros lange gali pasirinkti pinigų siuntimo skiltį ir užpildydamas visas reikalingas formas gali siųsti pinigus iš savo piniginės.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra piniginės peržiūros lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojo piniginės likutis atnaujinamas ir pateikiamas sėkmės pranešimas.	

Rezervacijos posistemis

Rezervacijos posistemis yra skirtas naudotojo kelionių rezervacijos funkcionalumams – kelionių paieška, pridėjimas, rezervacija, atšaukimas (žr. 9 pav.). Tuo pačiu paveldimi keli panaudojimo atvejai iš Kelionių posistemio, šie atvejai pateikiami lentelėse (**18 lentelė**, **19 lentelė**, **20 lentelė**, **21 lentelė**).



9 pav. Rezervacijos posistemio diagrama.

PA „Užpildyti kelionės paieškos kriterijus“ specifikacija pateikiama lentelėje **30 lentelė**. Naudotojas gali pasirinkti pagrindinius kelionės paieškos kriterijus – paėmimo ir paskirties vietas, vietų skaičių, pradžios laiką.

30 lentelė „Užpildyti kelionės paieškos kriterijus“ PA specifikacija.

PA. Užpildyti kelionės paieškos kriterijus		
Tikslas. Naudotojas gali užpildyti kelionės paieškos kriterijus.		
Aprašymas. Naudotojas gali pasirinkti savo kelionės paieškos kriterijus - paėmimo tašką, galutinė kelionės tašką, vietų skaičių, kelionės pradžios laiką.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra pagrindiniame lange.	
Aktoriai	Naudotojas, Google Maps	
Susiję panaudojimo atvejai	**Išplečiantys PA**	Ieškoti kelionių Pridėti kelionę
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojas užpildo kriterijus	

PA „Ieškoti kelionių“ specifikacija pateikiama lentelėje **31 lentelė**. Naudotojas ieškoti kelionių pagal savo pasirinktus kriterijus.

31 lentelė „Ieškoti kelionių“ PA specifikacija.

PA. Ieškoti kelionių		
Tikslas. Naudotojas gali ieškoti kelionių pagal savo užpildytus kelionės kriterijus		
Aprašymas. Po kelionių kriterijų užpildymo galima pasirinkti kelionės paieškos skiltį, naudotojui atrenkamos kelionės pagal jo kriterijus.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra pagrindiniame lange su užpildytais kelionės kriterijais.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	Peržiūrėti siūlomą kelionę
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojui pateikiamas visų surastų kelionių sąrašas su pranešimu.	

PA „Peržiūrėti siūlomą kelionę“ specifikacija pateikiama lentelėje **32 lentelė**. Naudotojas gali peržiūrėti siūlomą kelionę, kuri buvo rasta pagal kriterijus.

32 lentelė „Peržiūrėti siūlomą kelionę“ PA specifikacija.

PA. Peržiūrėti siūlomą kelionę		
Tikslas. Naudotojas gali peržiūrėti siūlomą kelionę iš surastų sąrašo.		
Aprašymas. Naudotojas esantis surastų kelionių sąraše gali pasirinkti vieną kelionę iš sąrašo ir peržiūrėti ją.		
Prieš sąlyga	Naudotojas yra prisijungęs ir yra surastų kelionių sąrašo lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	Peržiūrėti kelionės žinutes Registruotis prie kelionės Atšaukti kelionę Ištrinti kelionę
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojui rodomas kelionės peržiūros langas	

PA „Registruotis prie kelionės“ specifikacija pateikiama lentelėje **33 lentelė**. Naudotojas gali registruotis prie kelionės.

33 lentelė „Registruotis prie kelionės“ PA specifikacija.

PA. Registruotis prie kelionės		
Tikslas. Naudotojas gali prisiregistruoti prie kelionės		
Aprašymas. Naudotojas esantis ne kelionės vairuotojas gali prisiregistruoti prie kelionės jeigu kelionės vietų skaičius nėra lygus nuliui. Po registracijos yra laukiama vairuotojo patvirtinimo.		
Prieš sąlyga	Naudotojas yra prisijungęs ir nėra kelionės vairuotojas bei yra kelionės peržiūros lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojas yra užregistruojamas prie kelionės ir išsiunčiamas prašymas kelionės vairuotojui.	

PA „Pridėti kelionę“ specifikacija pateikiama lentelėje **34 lentelė**. Naudotojas gali pridėti kelionę pagal pasirinktus kriterijus.

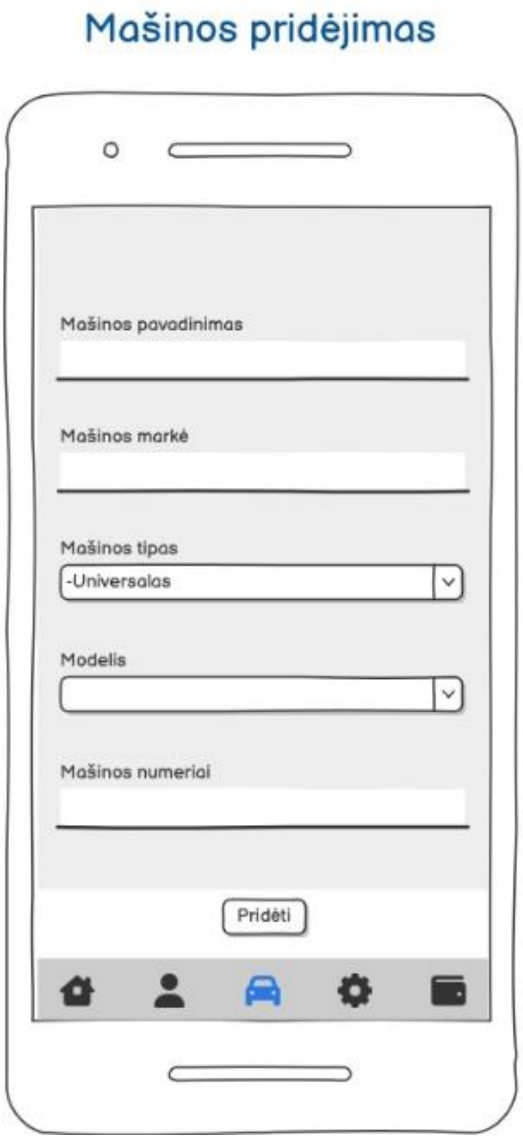
34 lentelė „Pridėti kelionę“ PA specifikacija.

PA. Pridėti kelionę		
Tikslas. Naudotojas gali pridėti kelionę.		
Aprašymas. Naudotojas po kelionės kriterijų užpildymo gali pasirinkti kelionės pridėjimo skiltį ir toliau užpildydamas kelionės pridėjimo formą gali pridėti kelionę. Svarbu paminėti, kad norint pilnai sukurti kelionę būtina turėti administratoriaus patvirtiną automobilį.		
Prieš sąlyga	Naudotojas yra prisijungęs bei yra užpildęs kelionės kriterijus ir yra pagrindiniame lange.	
Aktoriai	Naudotojas	
Susiję panaudojimo atvejai	**Išplečiantys PA**	
	Apimami PA	
	Specializuojami PA	
Po sąlyga	Naudotojo kelionė pridedama ir parodomas sėkmės pranešimas.	

2.1.3. Naudotojo sąsajos specifikacija

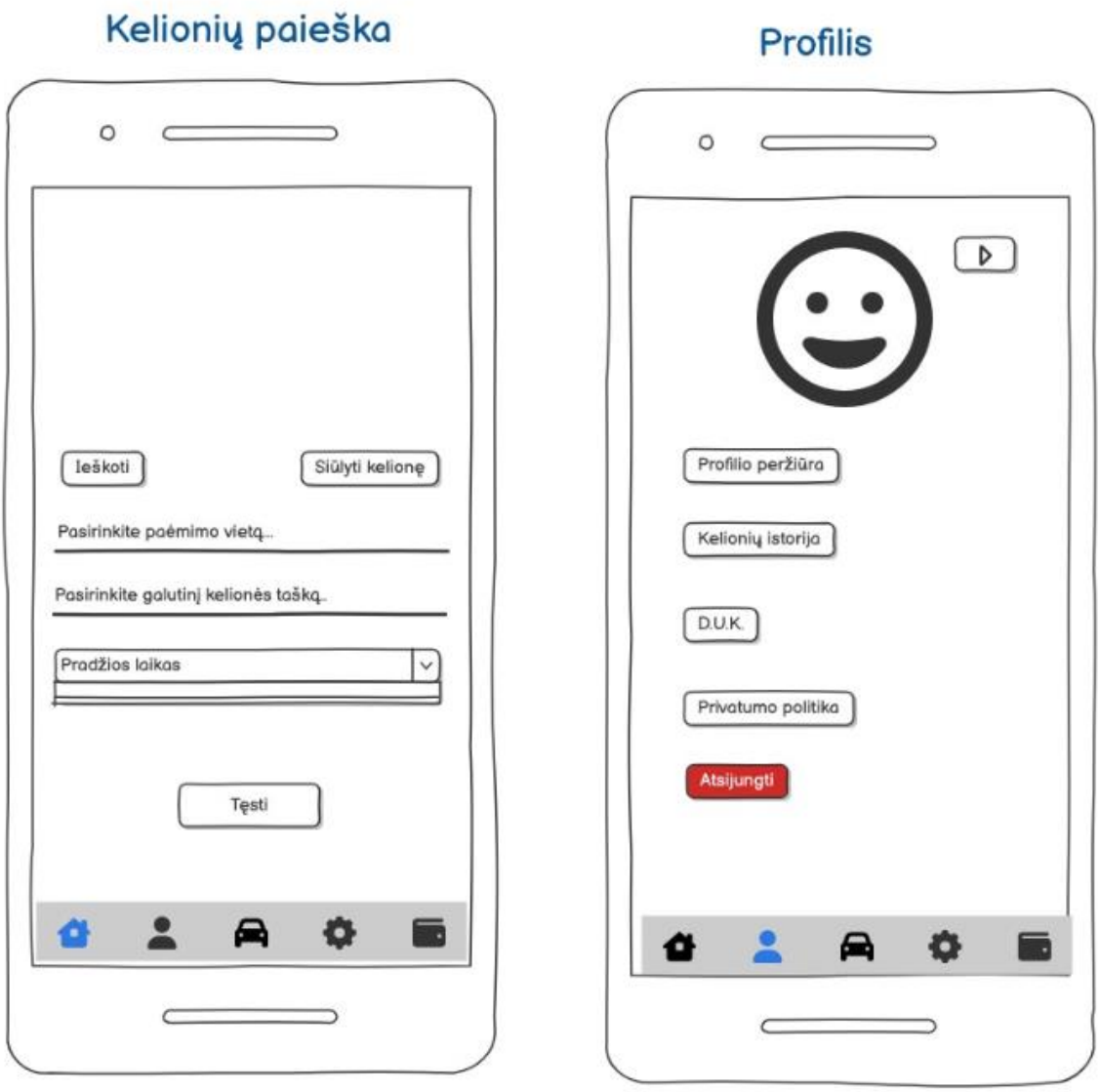
„JourneyFy“ programėlės kūrimui buvo naudojamas „Car Pooling App UI“ [5]. Tačiau pradžioje buvo paišomi eskizai naudojant Balsamiq (žr. 10 pav., 11 pav., 12 pav.).

Toliau pateikiami paveikslėliai yra bendriniai, bet galutinai nenusako galutinės programėlės vaizdo, šablonas buvo panaudotas, dėl spartesnio programėlės vystymo ir modernaus UI/UX sprendimo. Šie eskizai nubraižyti naudojant Balsamiq įrankį.



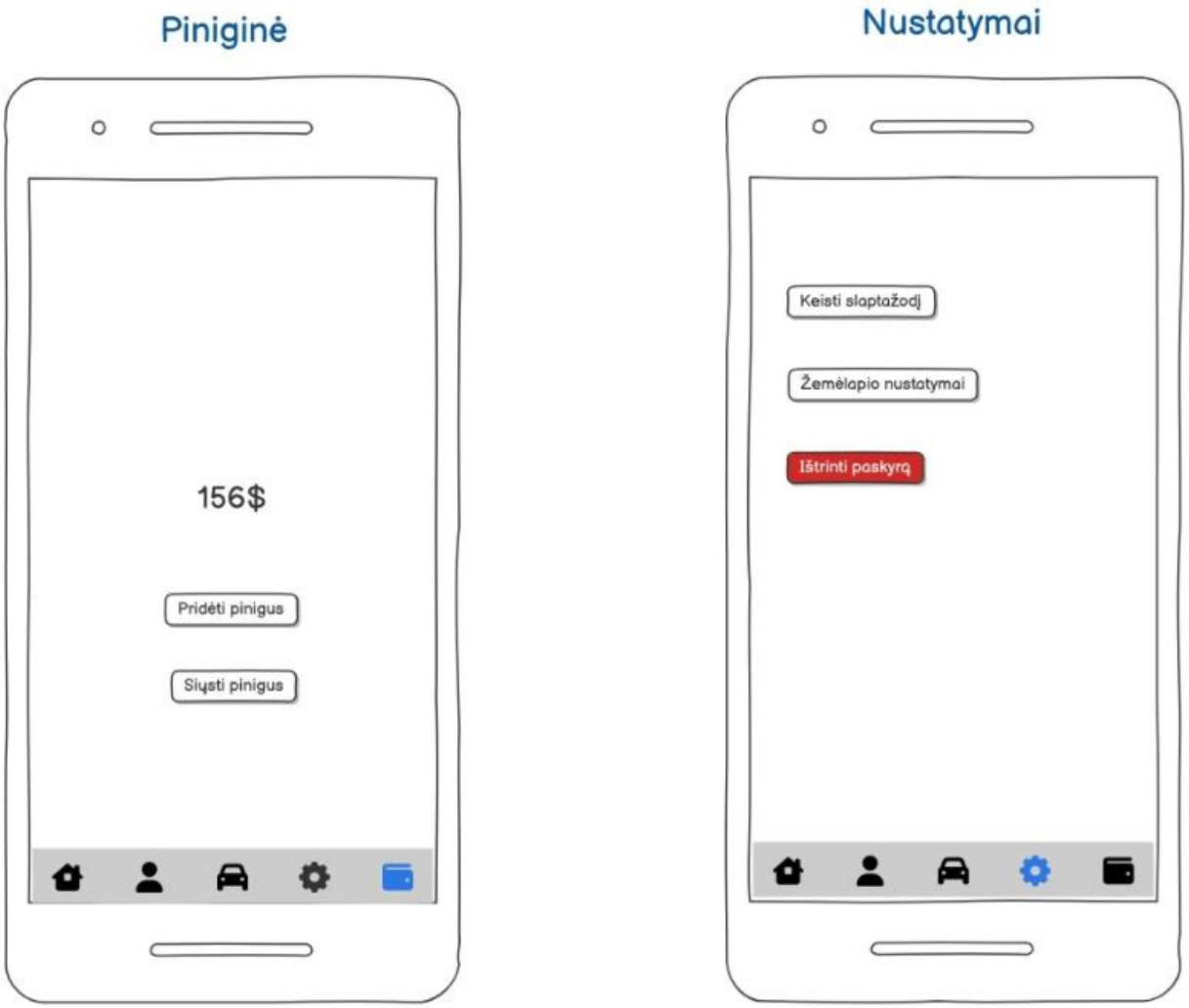
10 pav. Mašinos pridėjimo eskizas.

10 pav. pateikiamas pradinis mašinos pridėjimo lango eskizas, čia pradžioje buvo planuojami pradiniai informacijos įvedimo laukai mašinoms – pavadinimas, markė, tipas, modelis bei numeriai.



11 pav. Pradinio lango ir profilio eskizai.

11 pav. pateikiamas pradinio lango eskizas kuriame buvo planuojama implementuoti kelionių paiešką pasirenkant kelionės paėmimo ir pabaigos vietas bei pradžios laiką, tuo pačiu pateikiamas ir profilio lango eskizas ir visos galimos šio posistemio funkcijos, kaip nuotraukos pakeitimas, profilio informacijos peržiūra, kelionių istorija, atsijungimas.



12 pav. Piniginės ir nustatymų langų eskizai.

12 pav. pateikiamas piniginės posistemio ir nustatymų posistemio pradiniai eskizai, vystymo laikotarpiu šie posistemiai buvo apjungti su kitais, bendrai buvo sudėlioti šių posistemių funkcionalumai – slaptažodžio keitimas, žemėlapio nustatymai, paskyros ištrinimas.

2.1.4. Realizacijai keliami reikalavimai

Planavimo etapo metu „JourneyFy“ programėlei buvo apsirašyti ir iškelti šie pagrindiniai nefunkciniai reikalavimai:

1. Palaikoma programėlės naudotojo sąsaja turi būti lietuvių kalba;
2. Programa turi veikti Android telefonuose nuo API lygio 21 (Android 5.0) ir aukščiau;
3. Projekto dokumentacijoje turi būti aprašyti diegimo ir administravimo procesai;
4. Pagrindiniai ekranai (pvz., žemėlapis, rastų kelionių sąrašas) turi užsikrauti per mažiau nei 5 sekundes;
5. Kritiniai programėlės komponentai, ekranai, duomenų bazės operacijos turi būti padengti vienetų, integraciniais, naudotojų sąsajos testais.

2.1.5. Techninė specifikacija

Programėlei pasileisti reikia:

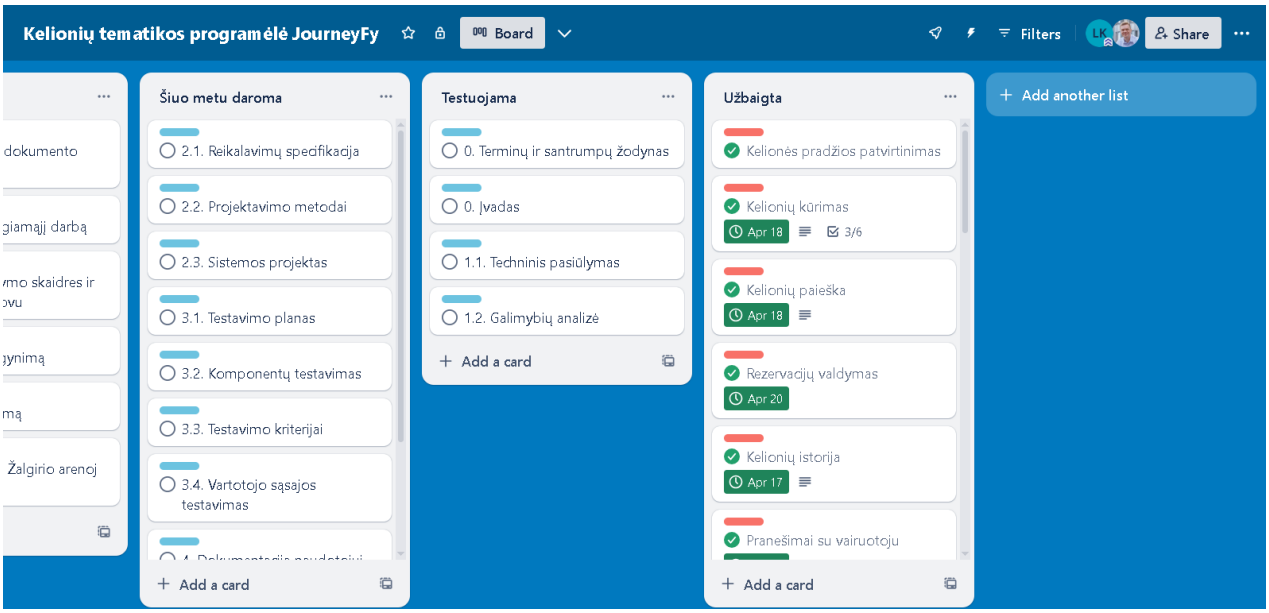
- IDE: bet kuri moderni JavaScript/TypeScript aplinka (pvz. Visual Studio Code, WebStorm);
- Android telefono su įrašyta **Expo Go (SDK 52)** aplinka (reikalavimas bent ne mažesnis negu 22 versijos Android API);
- Sukonfigūruoti aplinkos kintamieji (Firebase ir Google Maps servisų sąsajos kintamieji, t.y. API raktai).

2.2. Projektavimo metodai

2.2.1. Projektavimo valdymas ir eiga

Projektas vystomas pagal iteracinį modelį, leidžiantį pradėti vystymą neturint galutinio reikalavimų sąrašo ir nuolatos jį peržiūrėti bei papildyti kiekvienos iteracijos metu. Dėl šio modelio sistemos kūrimas vyksta etapais, todėl ankstyvoje fazėje lengviau aptikti defektus ir neoptimalias architektūros spragas. Be to, iteracinis modelis skatina skirti daugiau resursų programos projektavimui, vystymui ir testavimui nei išsamiai dokumentacijai.

Šiam metodui buvo pasitelktas „Trello“ įrankis. Čia buvo galima vizualizuoti visus darbus ir jau vykdomus. Šio įrankio pavyzdys pateikiamas 13 pav..



13 pav. Projekto valdymo įrankis Trello.

Visos užduotys Trello buvo išskirstytos:

- Bakalaurinio darbo dokumentas – užduotys susijusios su dokumento rašymu;
- Prisijungimo posistemis – prisijungimo posistemio darbai;
- Rezervacijos posistemis – rezervacijos posistemio darbai;
- Kelionių posistemis – kelionių posistemio darbai;
- Profilio posistemis – profilio posistemio darbai;
- Piniginės posistemis – piniginės posistemio darbai;
- „JourneyFy“ kūrimo darbai – užduotys, susijusios su programėlės vystymu;
- Šiuo metu daromos užduotys – „JourneyFy“ programėlės šiuo metu daromos užduotys;

- Testuojamos užduotys – užbaigtos užduotys, kurios yra testuojamos;
- Užbaigtos užduotys – „JourneyFy“ programėlės užbaigtos užduotys.

Darbai buvo skirstomi į „Nice-to-have“ (padaryti jeigu bus laiko) ir „Užbaigti kaip įmanoma greičiau“.

Buvo vykdomi dviejų savaičių sprintai, dirbama „Agile“ metodu. Iš viso buvo 8 sprintai, per kuriuos visi pagrindiniai „Užbaigti kaip įmanoma greičiau“ funkcionalumai buvo realizuoti. Bendras vystymo laikas: 2 mėnesiai. Prieš prasidedant sprintam, visi uždaviniai buvo iš anksto sudėlioti į bendras savo paskirties skiltis.

Visų sprintų aprašymai:

- Pirmojo sprinto ir antrojo sprinto metu – buvo paruoštas projekto šablonas, realizuota prisijungimo posistemė;
- Antrasis sprintas – buvo perkelti šablono UI elementai, kurie bus naudojami ir atnaujinta navigacija;
- Trečiasis sprintas – realizuotas kelionių posistemis;
- Ketvirtasis sprintas – realizuotas profilio posistemis;
- Penktasis sprintas – realizuotas piniginės posistemis;
- Šeštasis sprintas – realizuotas rezervacijos posistemis;
- Septintasis sprintas – testavimo darbai;
- Aštuntasis sprintas – galutiniai sistemos tvarkymo darbai.

2.2.2. Projektavimo technologija

Projektuojant sistemą buvo naudojamas universiteto suteiktas *MagicDraw* modeliavimui skirtas įrankis. Jame parengtos šios UML diagramos: naudotojo atvejų, veiklos, paketų, diegimo, klasių ir sekų. Kiekviena diagrama papildyta neformaliu aprašu teksto pavidalu.

Pagrindinės naudotų įrankių versijos:
- MagicDraw v11.0.23;
- UML v2.5.1.

2.2.3. Programavimo kalbos, derinimo, automatizavimo priemonės, operacinė sistemos

„JourneyFy“ buvo vystomas „Windows 10 Pro“ aplinkoje, naudojant Visual Studio Code ir Android Studio (Pixel 8, API 35) emuliatorių. Tikram įrenginių testavimui naudotas telefonas „Samsung A51“ su Expo Go.

Klientinė dalis sukurta su React Native per Expo Go, rašant TypeScript/JavaScript kodą.

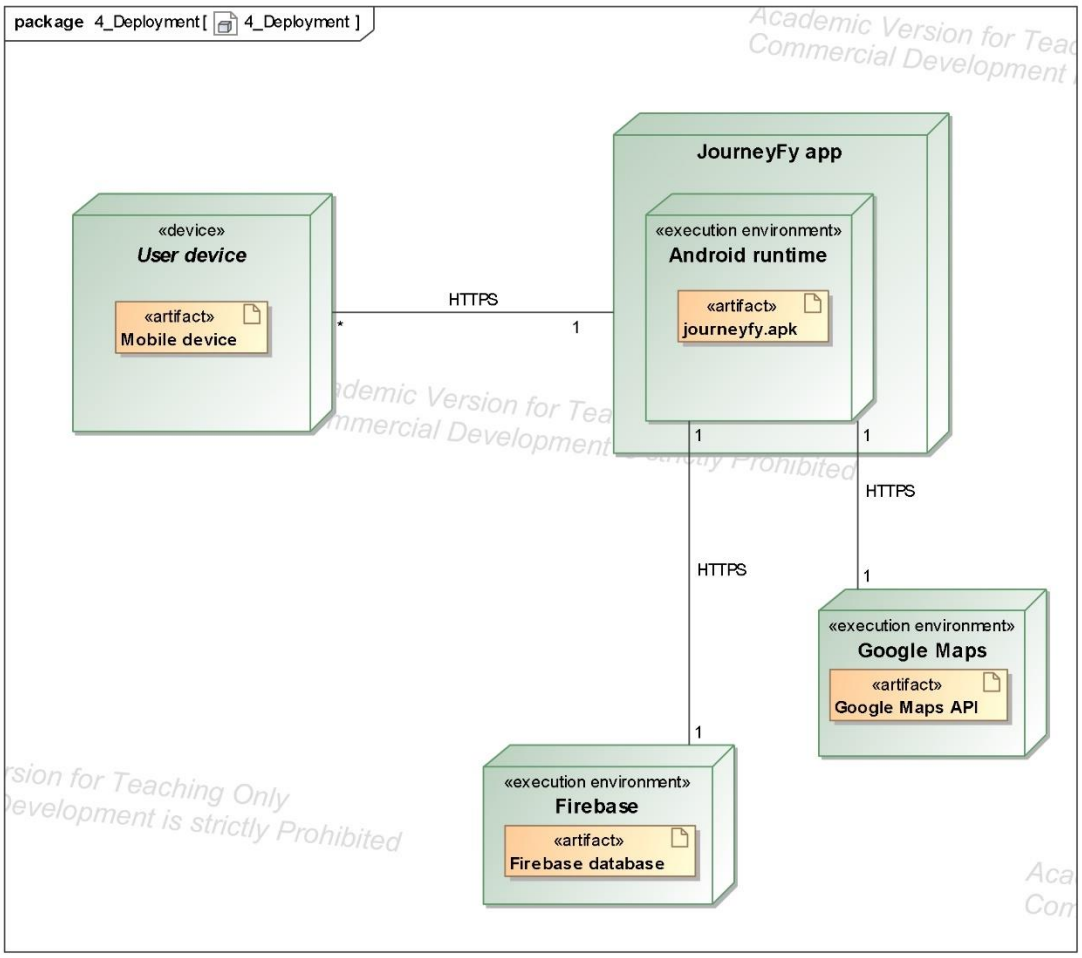
Serverinė dalis bei visi servisai (autentifikacija, realaus laiko duomenų bazė, saugykla, debesies funkcijos) įgyvendinti per Firebase.

Kodo bazė saugoma Github repozitorijoje.

2.3. Sistemos projektas

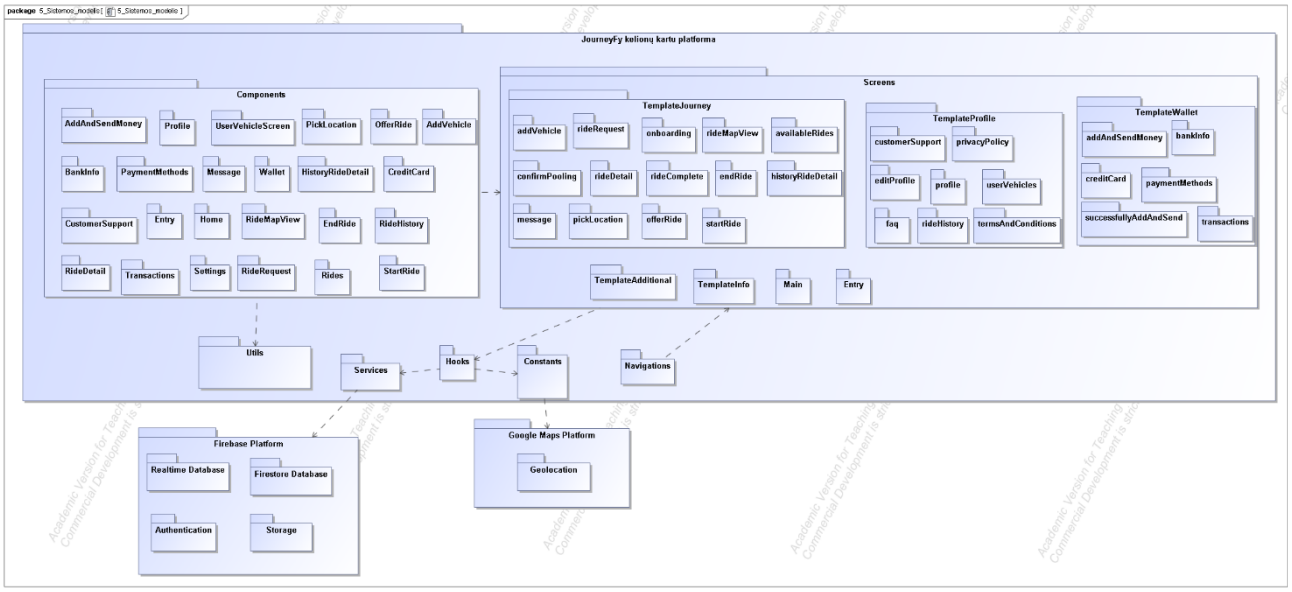
2.3.1. Statinis sistemos vaizdas

14 pav. vaizduojama sistemos diegimo diagrama, matomi pagrindiniai programėlėje naudojami servisai ir paleidimui naudojama aplinka.



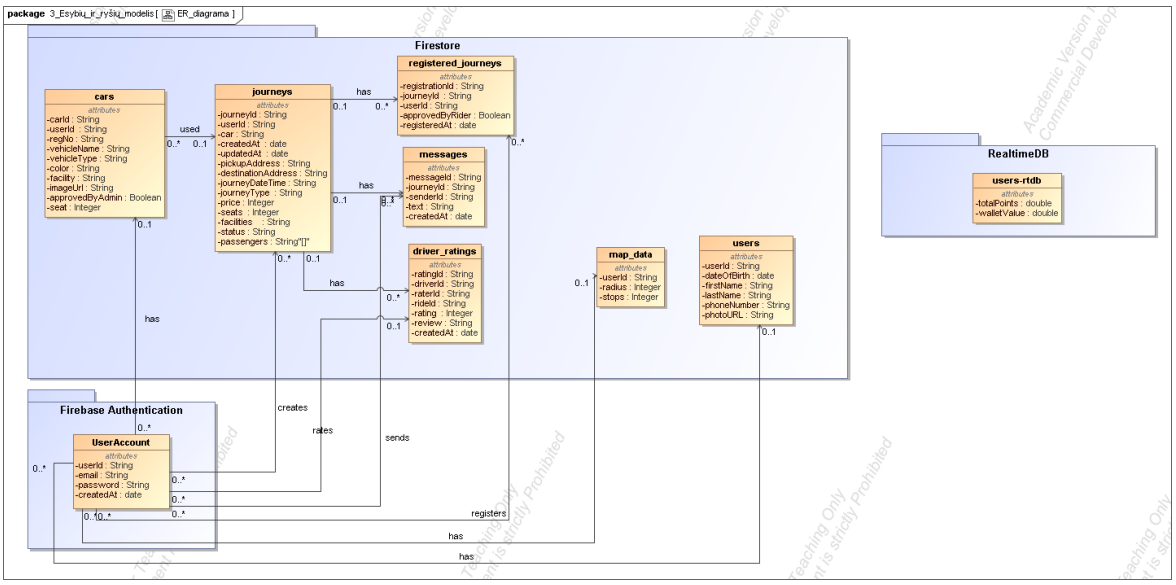
14 pav. „JourneyFy“ diegimo diagrama.

15 pav. vaizduojama paketų diagrama, matomas pagrindinis programėlės kliento ir serverio dalies pasiskirstymas. Pagrindiniai programėlės langai ir komponentai saugomi ekranų (angl. screens) ir komponentų (angl components) dalyse. Firebase duomenų bazės servisai naudojami servisų (angl. services) ir hooks dalyse.



15 pav. „JourneyFy“ paketų diagrama.

16 pav. vaizduojama programėlės ER diagrama. Buvo naudotas nerealiacinės duomenų bazės variantas su Firebase. Pagrindiniai naudotojų autentifikacijos duomenys saugomi Firebase Authentication (UserAccount) klasėje. Programėlės pagrindiniai naudojami duomenys, kaip kelionių (angl. journeys), mašinų (angl. cars), žemėlapio duomenų (angl. map data) ir naudotojų (angl. users) bei kitos jų vaikinės klasės saugomos Firebase Firestore duomenų bazėje. Realių duomenų duombazėje (angl. real time database) saugome naudotojų piniginės likutį ir taškus.

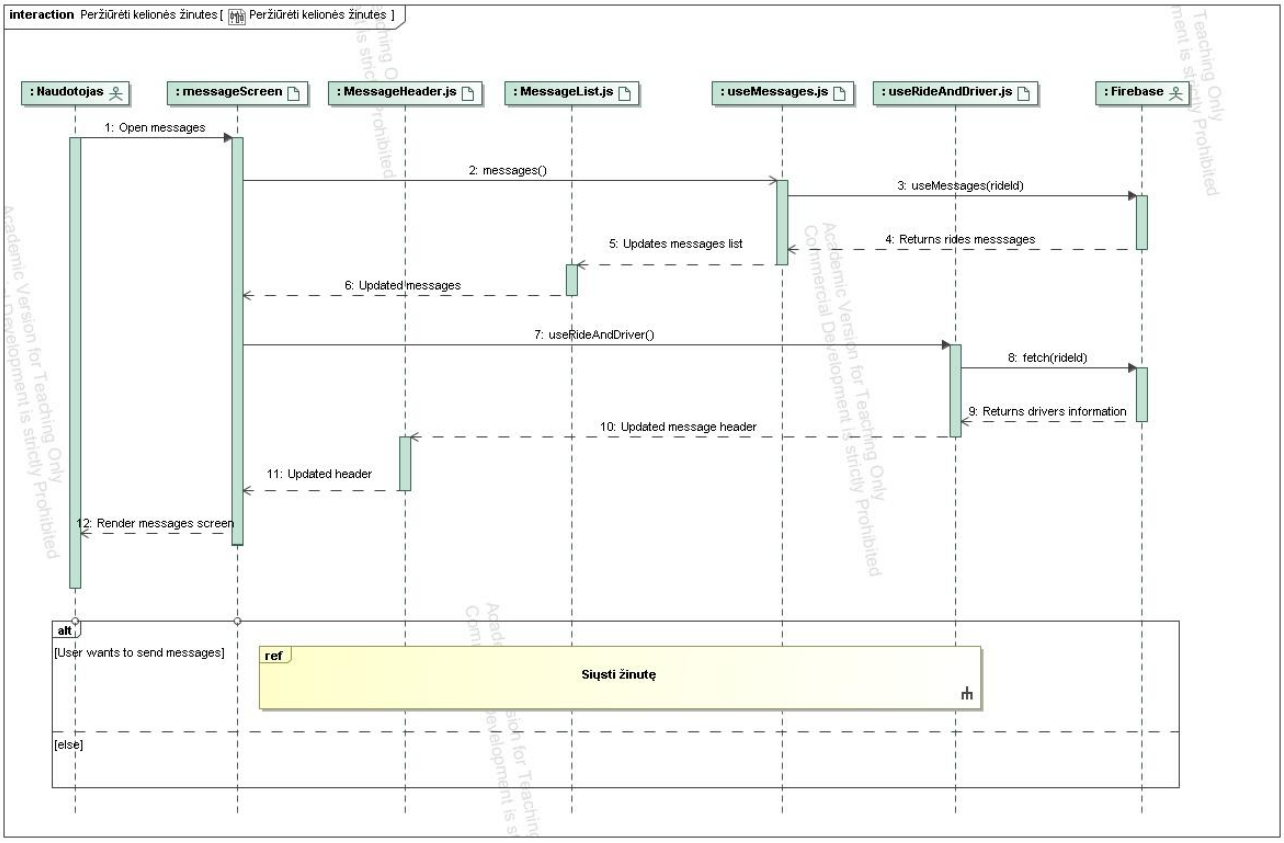


16 pav. „JourneyFy“ esybių ryšių diagrama.

2.3.2. Dinaminis sistemos vaizdas

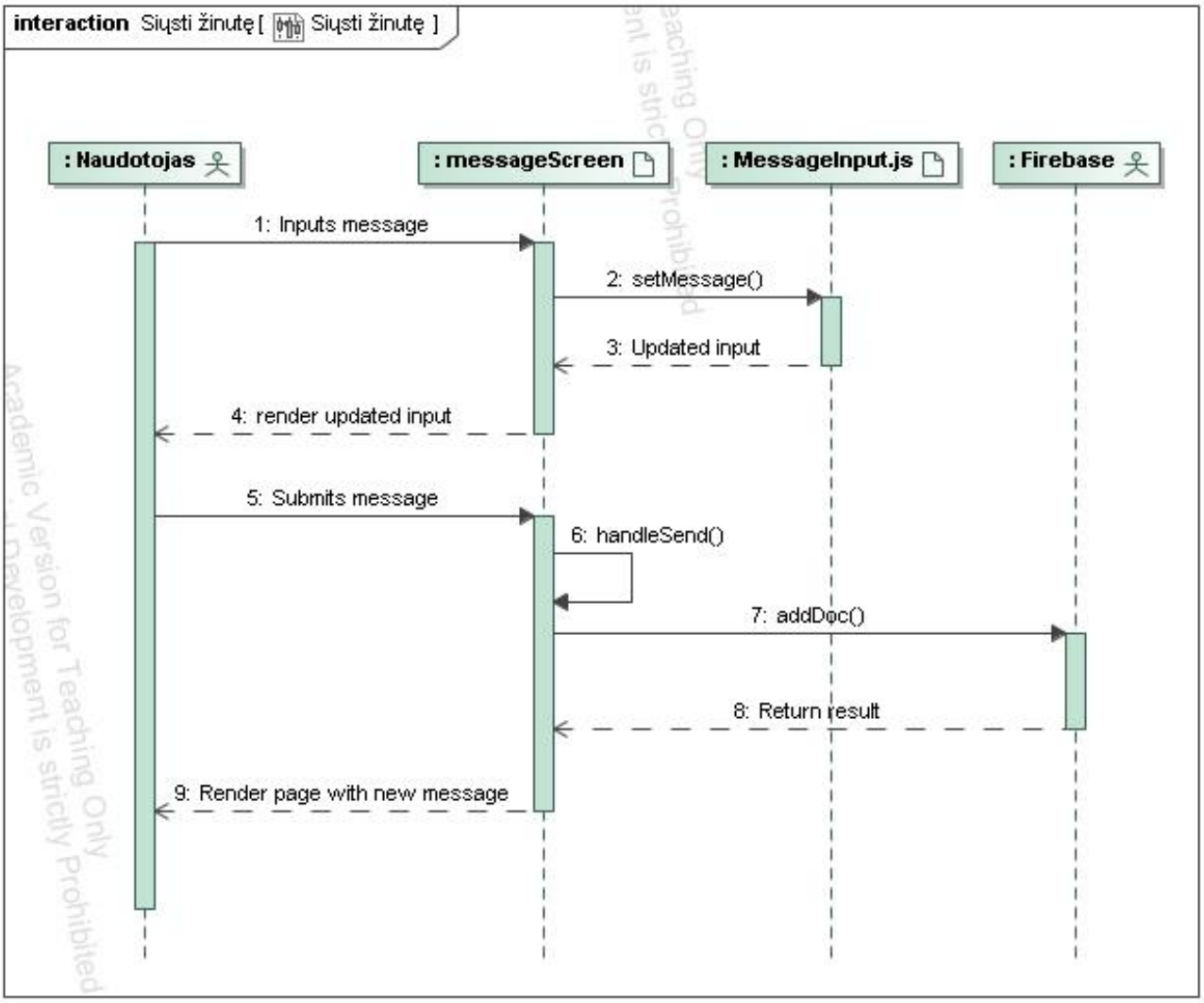
Dinaminė sistemos elgsena atvaizduojama sekų diagramomis: atrinkti pagrindiniai panaudojimo atvejai, geriausiai atspindintys „JourneyFy“ funkcionalumą, ir kiekvienam sukurtos diagramos, demonstruojančios komponentų tarpusavio sąveiką. Pagrindiniai PA atvejai, kurie buvo pasirinkti matomi lentelėse (**9 lentelė**, **17 lentelė**, **20 lentelė**, **21 lentelė**).

PA „Peržiūrėti kelionės žinutes“ sekų diagrama (žr. 17 pav.). Atvaizduojama ryšys tarp klientinės dalies – ekranų ir komponentų ir serverio dalies (angl. hooks). Visų pirmą yra kreipiamasi į ekranus, kurie naudojasi UI komponentais, kurie jau kreipiasi į mūsų Firebase naudojamus servisus.



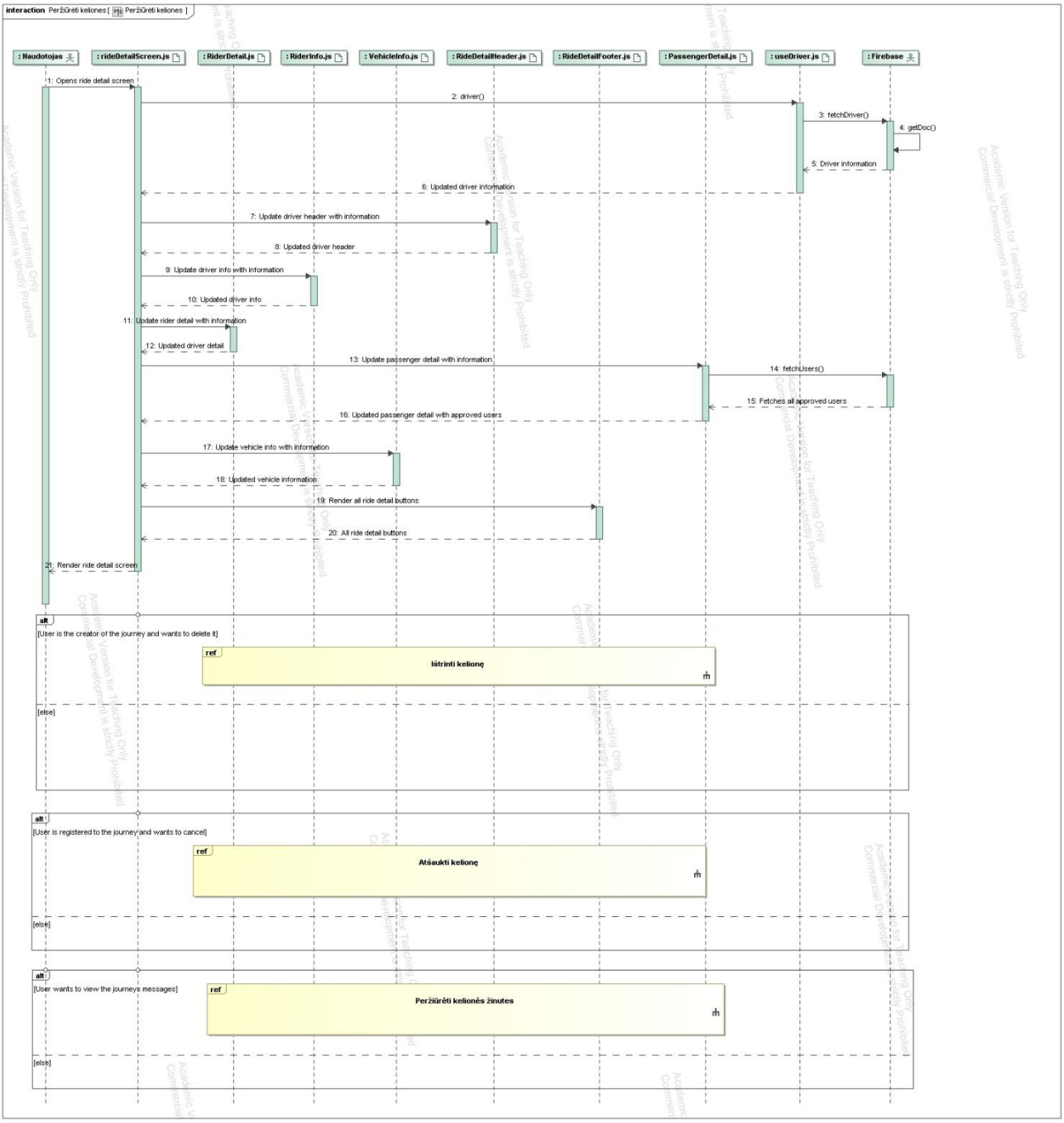
17 pav. „Peržiūrėti kelionės žinutes“ sekų diagrama.

PA „Siųsti žinutę“ sekų diagrama (žr. 18 pav.). Pavaizduojama, kaip įvyksta žinutės siuntimo procesas - kokie elementai, kaip ekranai ir komponentai komunikuoja tarpusavyje. Šio pavyzdžio atveju, matomas „messageScreen“ ekranas, kuris kreipiasi į „MessageInput“ UI komponentą iš kurio perduodama įvedimo informacija Firebase servisui.



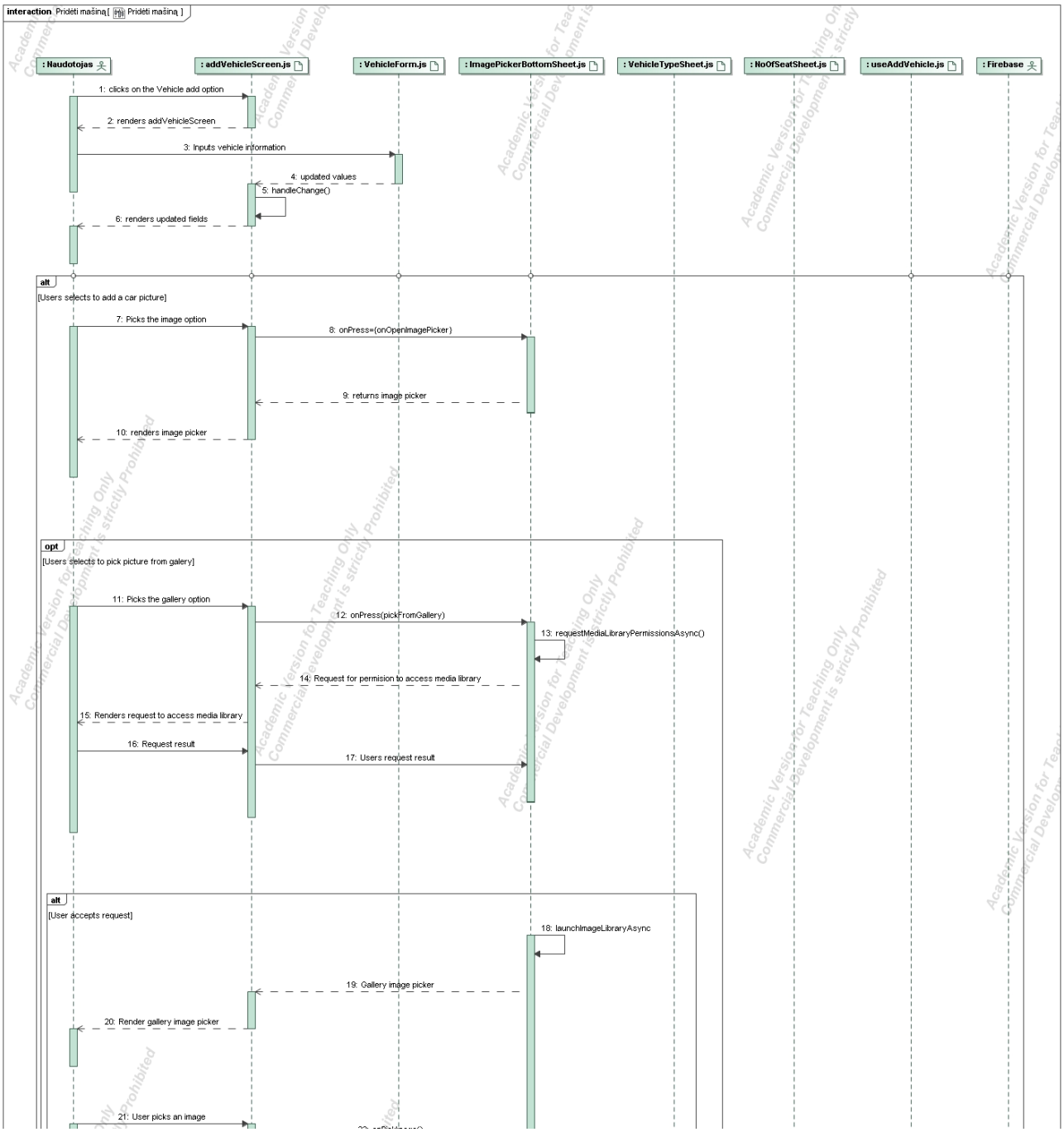
18 pav. „Siųsti žinutę“ sekų diagrama.

PA „Peržiūrėti keliones“ sekų diagrama (žr. 19 pav.). Pavaizduojamas kelionės peržiūros procesas ir kiti galimi veiksmai – kaip pradžioje užkraunamas ekranas su duomenimis naudojantis Firebase servisu, kaip visi komponentai šiuos duomenis išsaugoja ir pavaizduoja.

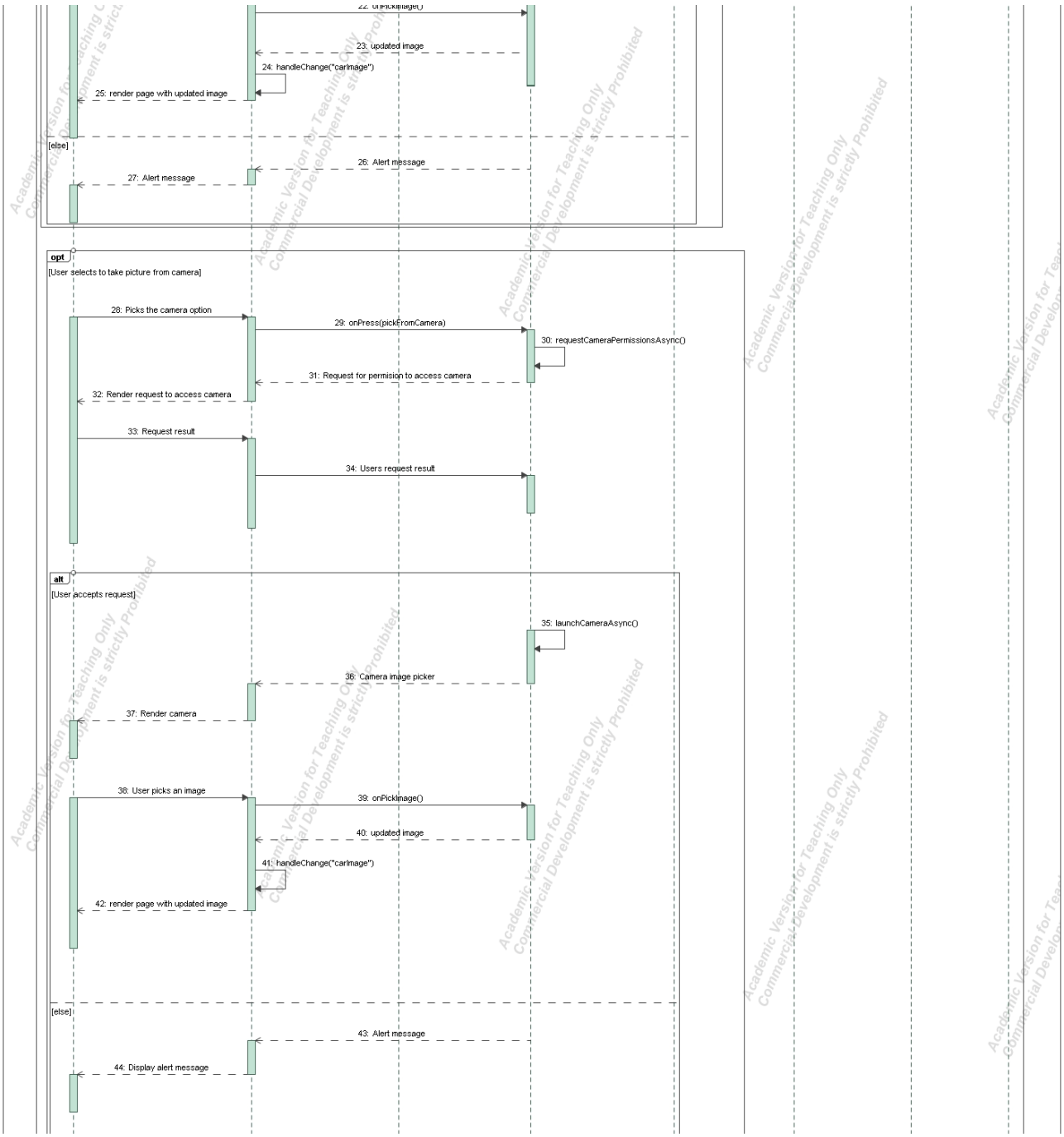


19 pav. „Peržiūrėti keliones“ sekų diagrama.

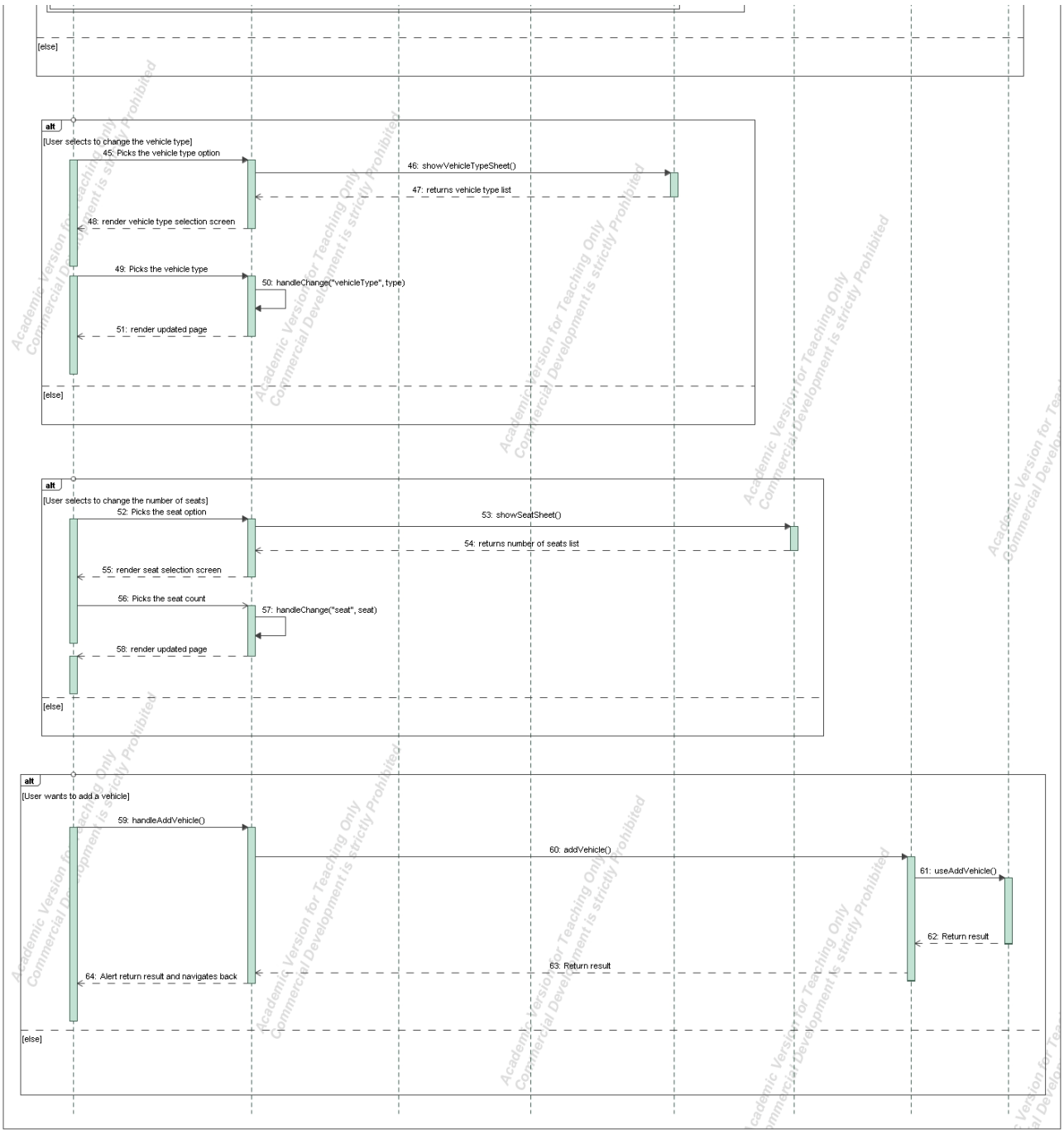
PA „Pridėti mašiną“ sekų diagrama (žr. 20 pav., 21 pav., 22 pav.). Atvaizduojamas kliento dalies ir serverio dalies komunikacijos procesas norint įkeltį nuotrauką, keičiant kitus mašinos informacijos parametrus, pridedant mašiną prie paskyros.



20 pav. „Pridėti mašiną“ sekų diagrama - 1 dalis.

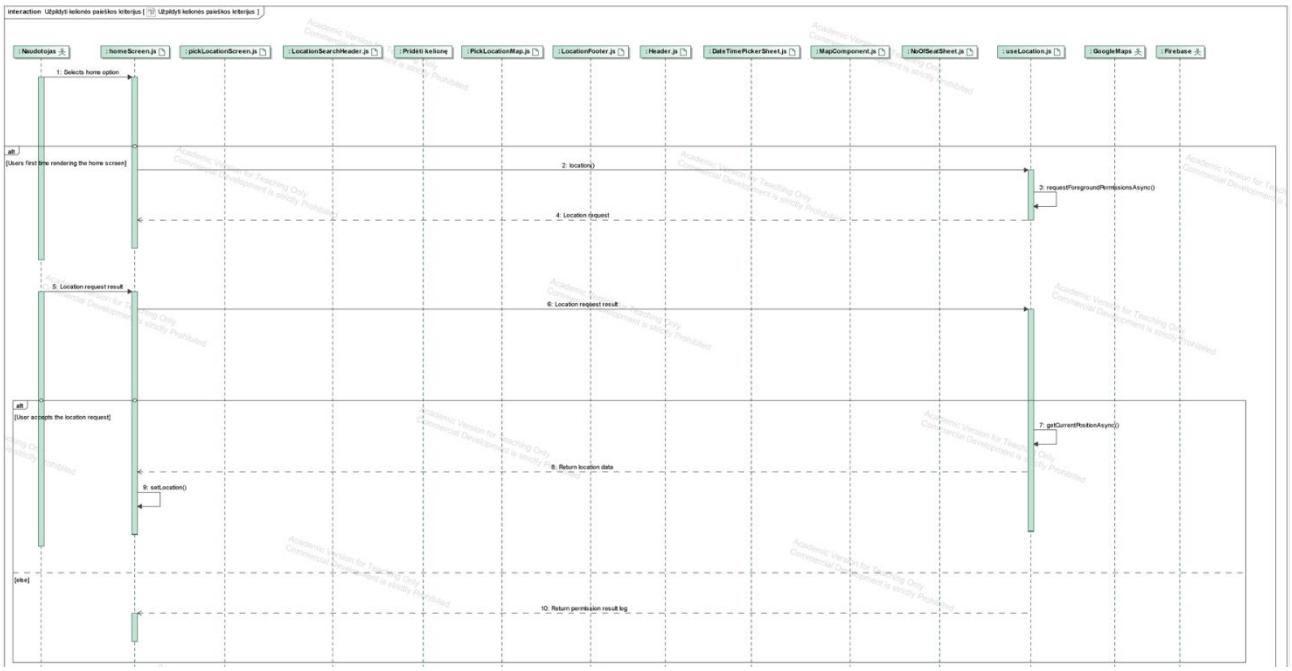


21 pav. „Pridėti mašiną“ sekų diagrama - 2 dalis.

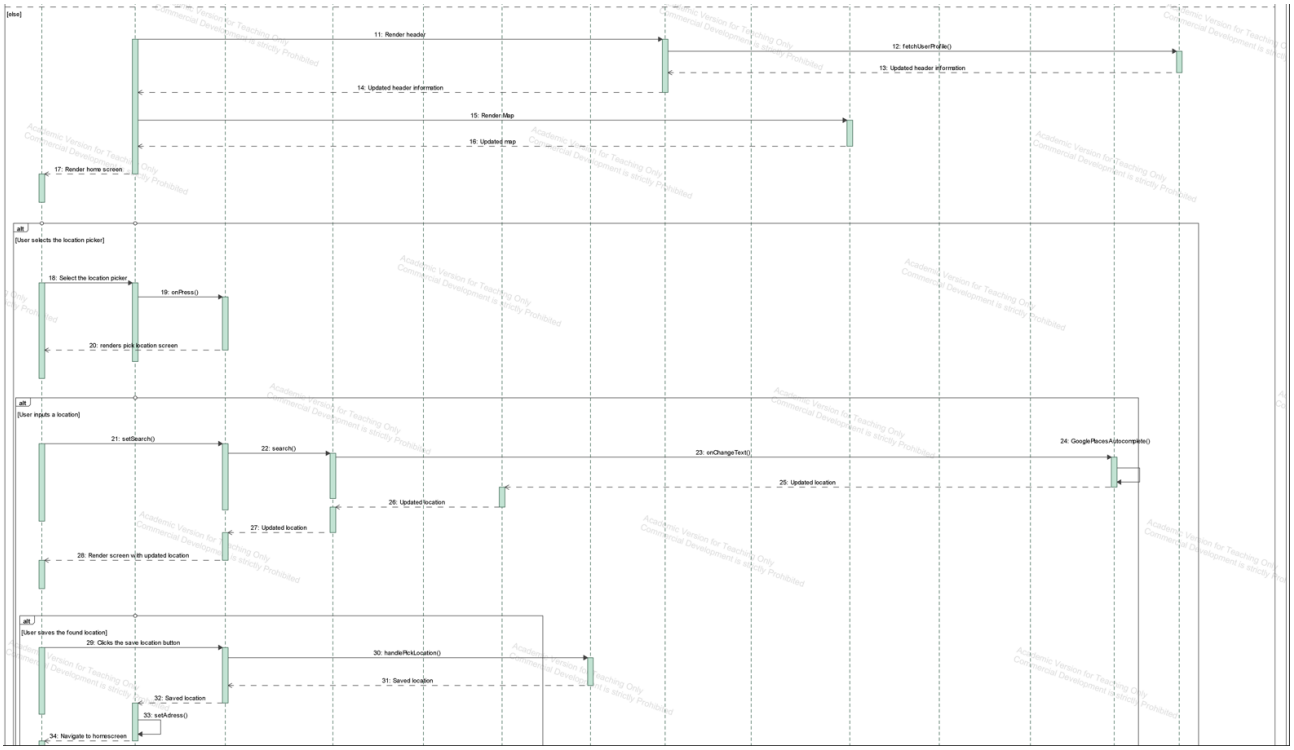


22 pav. „Pridėti mašiną“ sekų diagrama - 3 dalis.

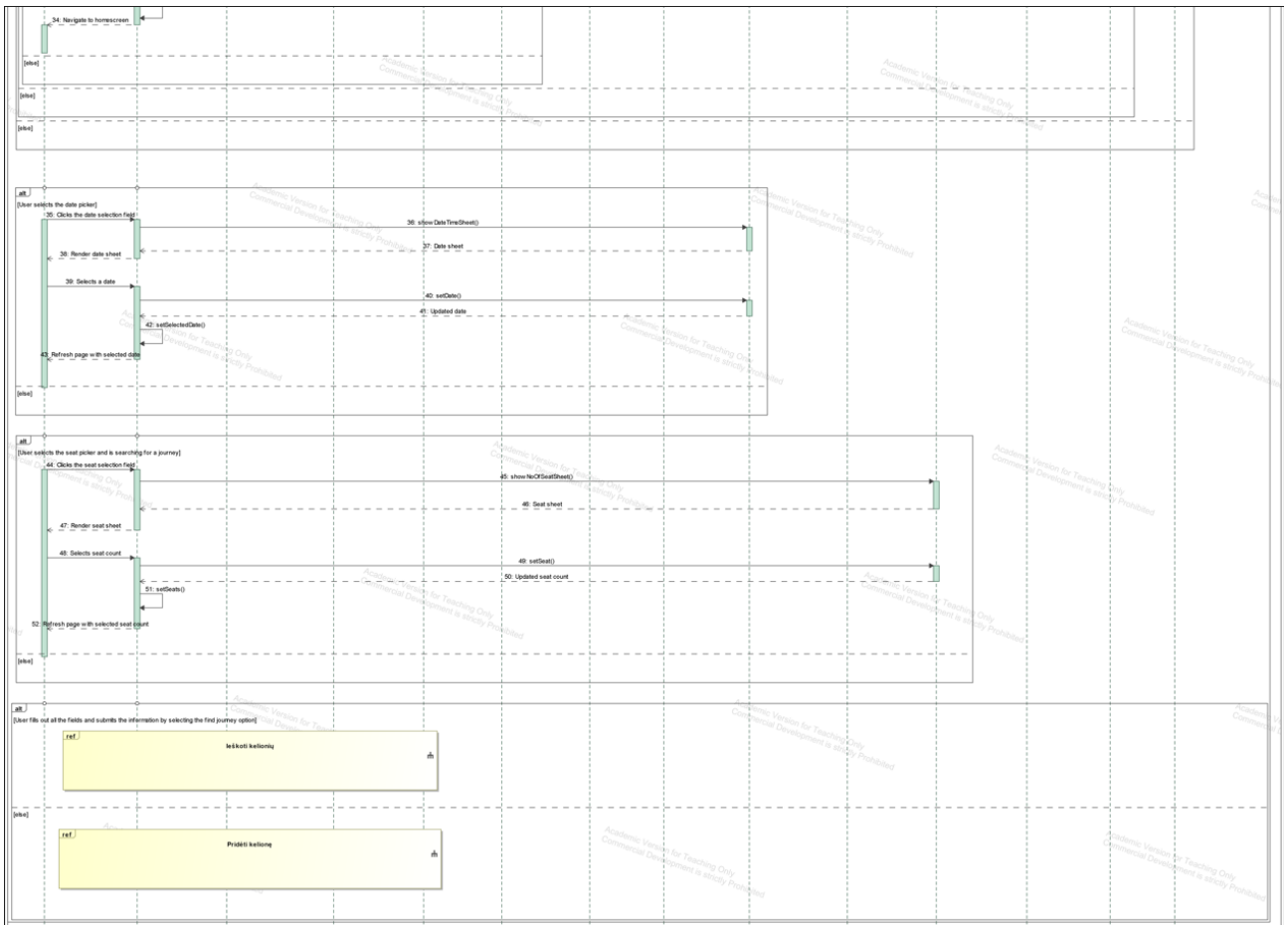
PA „Užpildyti kelionės paieškos kriterijus“ sekų diagrama (žr. 23 pav., 24 pav., 25 pav.). Atvaizduojamas kelionės paieškos procesas, kaip skirtingi paieškos kriterijų komponentai komunikuoja su kitais „JourneyFy“ paketais.



23 pav. „Užpildyti kelionės paieškos kriterijus“ sekų diagrama - 1 dalis.



24 pav. „Užpildyti kelionės paieškos kriterijus“ sekų diagrama - 2 dalis.



25 pav. „Užpildyti kelionės paieškos kriterijus“ sekų diagrama - 3 dalis.

3. Testavimas

3.1. Testavimo planas

Programėlės testavimui buvo numatyti tokie žingsniai:

- Atliekama kodo peržiūra – pereinamos pagrindinės kodo dalys naudojant statinės kodo analizės įrankį Lint;
- Vienetų testai naudojant Jest testavimo biblioteką Javascript/Typescript kodui – testai pagrindiniams programėlėms ekranams (angl. screens);
- Integraciniai testai – atliekami integraciniai testai naudojant React Native Testing Library ir Jest pagrindiniams programėlės Firebase servisams naudojamiems programėlėje (angl. hooks);
- Naudotojo sąsajos testai naudojant Jest ir React Native Testing Library visiems programėlės UI komponentams (angl. components).

Šias užduotis reikia atlikti prieš pradedant testavimo veiklą:

- Yra išsami programinės įrangos specifikacija, išreikšta atvejų ir naudojimo scenarijų modeliais;
- Veikianti ir įdiegta programinė įranga (t.y. programėlė);
- Nustatytos procedūros su problemų sprendimais testavimo metu;
- Apibrėžtas panaudojimų atvejų modelis priimtimumo testavimui;
- Sukurta testavimo aplinka;
- Pilnai paskirstyti testavimo ištekliai;
- Apibrėžti priėmimo testavimo standartai.

Bus naudojami šie testavimo metodai:

1. Vienetų testavimai;
2. Integraciniai testavimai;
3. Statinė kodo analizė;
4. UI/UX testavimas.

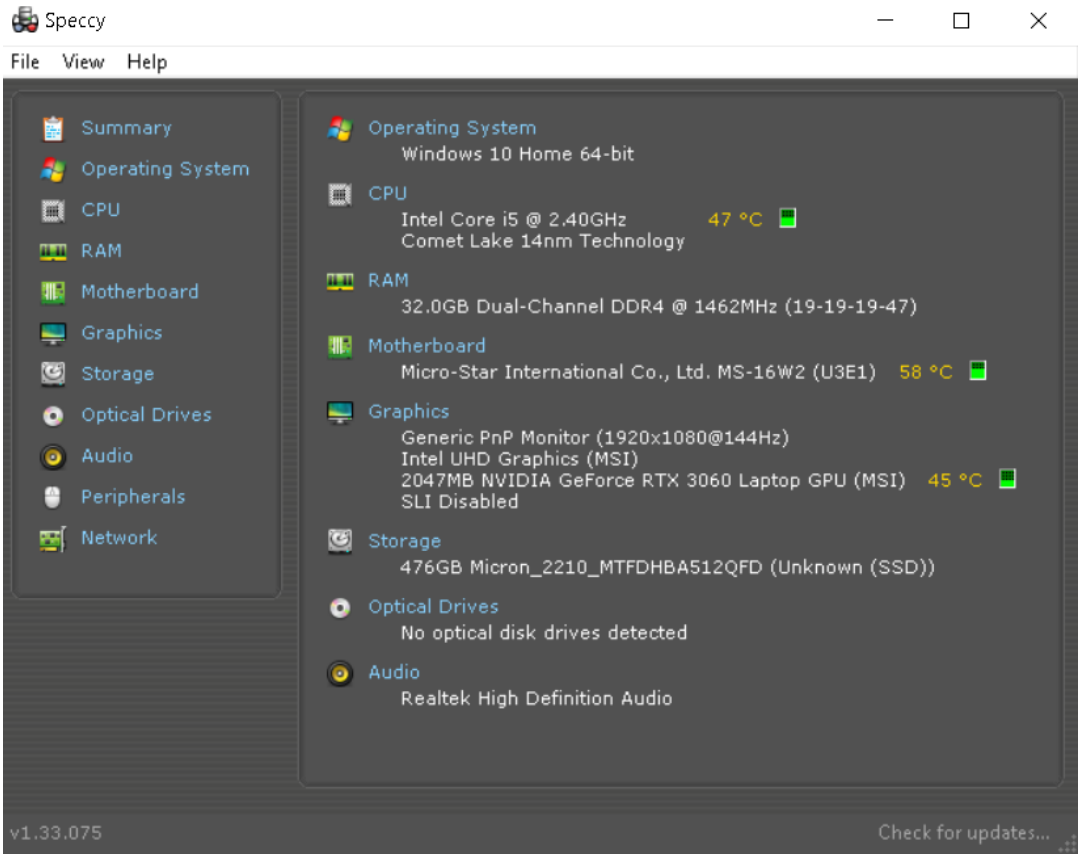
Testavimui bus reikalingas kompiuteris. Testavimui galima naudoti telefoną arba emuliatorių. Įrankiai, kurie bus naudojami:

- Expo Go (platforma leidžianti pasileisti React-Native projektą per savo telefoną);
- Android Studio (Androido emuliatorius simuliuojantis Android telefoną);
- Visual Studio Code (Programėlės paleidimui naudotas IDE).

Testavimo tvarkaraštis

Testavimo užduotis	Pradžia	Terminas
Statinė kodo analizė	2024-04-23	2024-04-26
Vienetų testai	2024-04-25	2024-04-28
Integracijos testai	2024-04-28	2024-04-30
Naudotojo sąsajos testai	2024-04-30	2024-05

Testavimas buvo atliekamas naudojant Windows 10 operacinę sistemą. Platesnė sistemos specifikacija gali būti matoma nuotraukoje, naudojant „Speccy“ įrankį (žr. 26 pav.).



26 pav. Testavimo kompiuterio specifikacija.

3.2. Testavimo kriterijai

Statinė kodo analizė:

- Kodo refaktorizacijos darbai naudojant Lint įrankį;
- Kodas yra lengvai palaikomas;
- Kodas yra lengvai skaitomas ir suprantamas.

Vienetų testai:

- Kodo padengimo lygis turi apimti bent 50% kritinių programėlės dalių, tai pasirinkta dėl programėlės dydžio;
- Padengiami vienetų testais pagrindiniai programėlės langai (angl. screens);
- Rašomi automatizuoti vienetų testai naudojant Jest ir React Native Testing Library.

Integraciniai testai:

- Kodo padengimo lygis turi apimti bent 50% kritinių programėlės dalių, tai pasirinkta dėl programėlės dydžio;
- Padengiami integraciniais testais pagrindiniai programėlės Firebase servisai (angl. hooks);
- Rašomi automatizuoti integraciniai testai naudojant Jest ir React Native Testing Library.

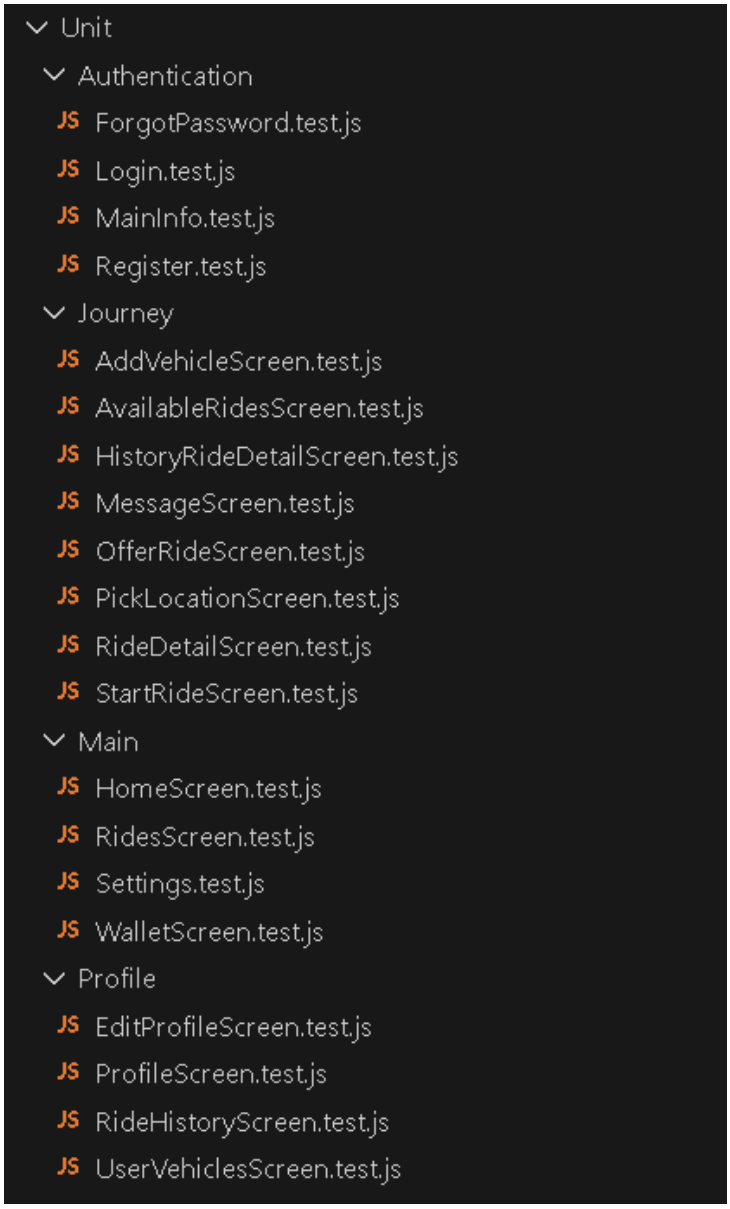
Naudotojo sąsajos testai:

- Kodo padengimo lygis turi apimti bent 50% kritinių programėlės dalių, tai pasirinkta dėl programėlės dydžio;
- Padengiami UI/UX testais pagrindiniai programėlės UI komponentai (angl. components);
- Rašomi automatizuoti naudotojo sąsajos testai naudojant Jest ir React Native Testing Library.

3.3. Vienetų testavimas

Vienetų testams apsirašyti buvo pasirinkta naudoti Jest ir React Native Testing Library, dėl paprastos integracijos su React Native karkasu. Šių testų pagrindinis tikslas – ištestuoti „JourneyFy“ programėlės pagrindinių ekranų funkcionalumą. Buvo testuojamas pagrindinis šių elementų funkcionalumas ir logika. Šie testai yra automatizuoti.

Buvo simuliuojami visi naudojami ekranų komponentai ir tikrinami gauti rezultatai. Bendrai buvo parašyti 52 vienetų testai, kurie visi įvykdyti sėkmingai (žr. 27 pav.).



27 pav. „JourneyFy“ vienetų testai.

Pagrindinis šių testų kriterijus – 50% kritinių programėlės funkcijų padengimas testais. Vienetų testai turėjo padengti pagrindinius programėlės langus. Bendra vienetų testų padengimo statistika matoma nuotraukoje (žr. 28 pav.).

screens/Entry		98.83%	85/86	81.25%	26/32	100%	23/23	98.83%	85/86
screens/Entry/auth		0%	0/84	0%	0/18	0%	0/50	0%	0/83
screens/Main		66.07%	74/112	50%	19/38	60%	18/30	66.66%	70/105
screens/TemplateAdditional/reviews		0%	0/7	0%	0/2	0%	0/4	0%	0/7
screens/TemplateInfo/notifications		0%	0/30	0%	0/7	0%	0/11	0%	0/28
screens/TemplateJourney/addVehicle		92%	23/25	50%	1/2	91.66%	11/12	92%	23/25
screens/TemplateJourney/availableRides		92.59%	25/27	70%	14/20	100%	9/9	92.3%	24/26
screens/TemplateJourney/confirmPooling		0%	0/17	0%	0/2	0%	0/8	0%	0/17
screens/TemplateJourney/endRide		0%	0/19	0%	0/4	0%	0/4	0%	0/18
screens/TemplateJourney/historyRideDetail		80%	8/10	100%	4/4	33.33%	1/3	80%	8/10
screens/TemplateJourney/message		83.33%	10/12	50%	2/4	100%	2/2	90.9%	10/11
screens/TemplateJourney/offerRide		75%	33/44	62.5%	20/32	60%	6/10	74.41%	32/43
screens/TemplateJourney/onboarding		0%	0/40	0%	0/14	0%	0/19	0%	0/40
screens/TemplateJourney/pickLocation		40.5%	32/79	30.43%	7/23	34.78%	8/23	40.5%	32/79
screens/TemplateJourney/rideComplete		0%	0/17	0%	0/2	0%	0/8	0%	0/17
screens/TemplateJourney/rideDetail		36.36%	20/55	8.69%	2/23	36.36%	4/11	37.73%	20/53

28 pav. „JourneyFy“ vienetų testų padengimas.

Galutiniai visų testų (vienetų, integracinių, UI/UX) padengimai ir rezultatai matomi paveikslėliuose. (žr. 29 pav., 30 pav.). Iš viso buvo sukurti 178 skirtingi testai.

```
Test Suites: 88 passed, 88 total
Tests:       178 passed, 178 total
Snapshots:   0 total
Time:        8.654 s
```

29 pav. „JourneyFy“ testų rezultatai - 1 dalis.

All files

44.27% Statements 1040/2349 **31.63%** Branches 291/920 **38.76%** Functions 271/699 **44.57%** Lines 1011/2268

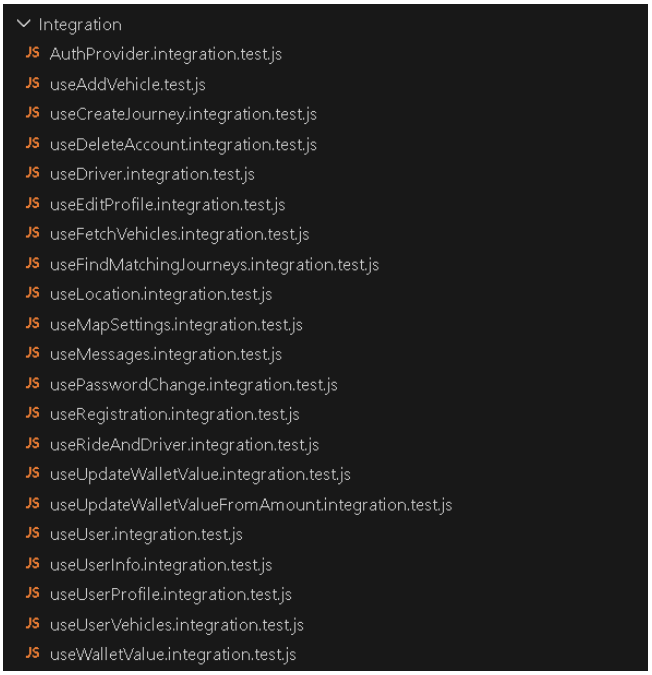
30 pav. „JourneyFy“ testų rezultatai - 2 dalis.

3.4. Integracinis testavimas

Integraciniams testams apsirašyti buvo pasirinkta naudoti Jest ir React Native Testing Library, dėl paprastos integracijos su React Native karkasu. Šių testų pagrindinis tikslas – ištestuoti „JourneyFy“ programėlės pagrindinių Firebase servisų funkcionalumą (angl. hooks). Šie testai yra automatizuoti.

Iš esmės buvo simuliuojami visi naudojami Firebase servisai (angl. hooks), kurie atlikdavo duomenų bazės operacijas.

Iš viso buvo parašyti 45 integraciniai testai, kurie visi įvykdyti sėkmingai (žr. 31 pav.).



31 pav. „JourneyFy“ integraciniai testai.

Pagrindinis šių testų kriterijus – 50% kritinių programėlės funkcijų padengimas testais. Integraciniai testai turėjo padengti pagrindinius programėlės Firebase servisus. Bendra integracinių testų padengimo statistika matoma nuotraukoje (žr. 32 pav.).

File		Statements		Branches		Functions		Lines	
hooks		66.24%	369/557	41.21%	68/165	70.1%	68/97	66.6%	359/539

32 pav. „JourneyFy“ integracinių testų padengimas.

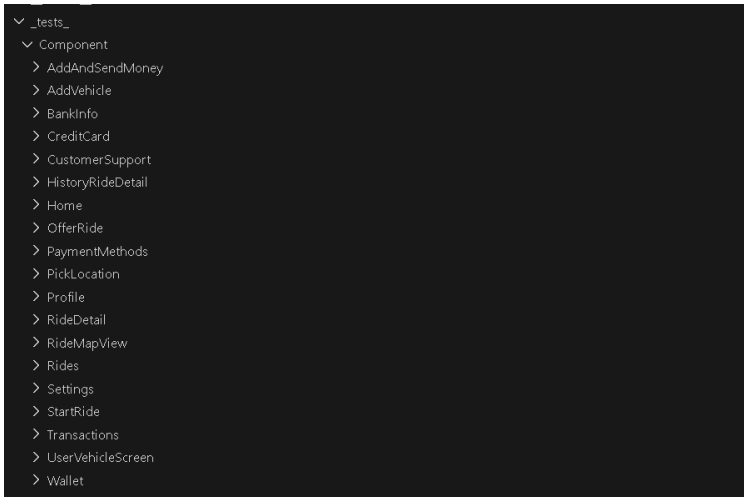
Galutiniai visų testų (vienetų, integracinių, UI/UX) padengimai ir rezultatai matomi paveikslėliuose. (žr. 29 pav., 30 pav.). Iš viso buvo sukurti 178 skirtingi testai.

3.5. Naudotojo sąsajos testavimas

Naudotojo sąsajos testams apsirašyti buvo pasirinkta naudoti Jest ir React Native Testing Library, dėl paprastos integracijos su React Native karkasu. Šių testų pagrindinis tikslas – ištestuoti „JourneyFy“ programėlės pagrindinius UI komponentus (angl. components). Šie testai yra automatizuoti.

Iš esmės buvo simuliuojamos ir testuojamos visų komponentų formos, mygtukai ir tikrinami gauti rezultatai.

Iš viso buvo parašyti 81 naudotojo sąsajos testų, kurie visi įvykdyti sėkmingai (žr. 33 pav.).



33 pav. „JourneyFy“ naudotojo sąsajos testai.

Pagrindinis šių testų kriterijus – 50% kritinių programėlės funkcijų padengimas testais. Naudotojo sąsajos testai turėjo padengti pagrindinius programėlės UI/UX komponentus. Bendra naudotojo sąsajos testų padengimo statistika matoma nuotraukoje (žr. 34 pav.).

components		6.06%	2/33	0%	0/14	5.88%	1/17	6.06%	2/33
components/AddAndSendMoney		75%	6/8	100%	2/2	66.66%	2/3	75%	6/8
components/AddVehicle		83.33%	35/42	70.37%	19/27	90.9%	10/11	83.33%	35/42
components/BankInfo		64.28%	9/14	100%	0/0	60%	3/5	64.28%	9/14
components/CreditCard		78.57%	11/14	100%	0/0	75%	3/4	78.57%	11/14
components/CustomerSupport		100%	11/11	50%	1/2	100%	4/4	100%	11/11
components/EndRide		0%	0/39	0%	0/17	0%	0/11	0%	0/35
components/Entry		0%	0/7	0%	0/9	0%	0/4	0%	0/6
components/HistoryRideDetail		75.86%	22/29	78.57%	11/14	75%	6/8	75.86%	22/29
components/Home		32.65%	16/49	33.89%	20/59	36%	9/25	31.25%	15/48
components/Message		0%	0/47	0%	0/52	0%	0/12	0%	0/42
components/OfferRide		38.23%	13/34	25%	7/28	46.15%	6/13	38.23%	13/34
components/PaymentMethods		100%	9/9	83.33%	5/6	100%	5/5	100%	9/9
components/PickLocation		13.04%	3/23	25%	2/8	11.11%	1/9	13.04%	3/23
components/Profile		23.68%	18/76	6.66%	3/45	42.85%	6/14	20.83%	15/72
components/RideDetail		41.55%	32/77	22.22%	10/45	42.3%	11/26	42.66%	32/75
components/RideHistory		0%	0/15	100%	0/0	0%	0/8	0%	0/15
components/RideMapView		28.57%	4/14	100%	0/0	50%	2/4	28.57%	4/14
components/RideRequest		0%	0/37	0%	0/17	0%	0/15	0%	0/35
components/Rides		33.33%	6/18	66.66%	4/6	25%	2/8	33.33%	6/18
components/Settings		27.27%	6/22	0%	0/10	28.57%	2/7	27.27%	6/22
components/StartRide		40%	10/25	42.85%	3/7	28.57%	2/7	34.78%	8/23
components/Transactions		100%	9/9	100%	4/4	100%	4/4	100%	9/9
components/UserVehicleScreen		54.16%	13/24	66.66%	4/6	39.46%	5/13	54.16%	13/24
components/Wallet		16.66%	3/18	0%	0/4	12.5%	1/8	16.66%	3/18

34 pav. „JourneyFy“ naudotojo sąsajos testų padengimas.

Galutiniai visų testų (vienetų, integracinių, UI/UX) padengimai ir rezultatai matomi paveikslėliuose. (žr. 29 pav., 30 pav.). Iš viso buvo sukurti 178 skirtingi testai.

4. Dokumentacija naudotojui

4.1. Apibendrintas sistemos galimybių aprašymas

„JourneyFy“ programėlės pagrindinis aktorius – naudotojas. Naudotojas paveldi visas svečio funkcijas. Tuo pačiu naudotojas gali prisijungęs peržiūrėti savo registruotas ir sukurtas keliones. Naudotojas gali peržiūrėti savo sukurtų kelionių kitų keleivių prašymus bei pradėti ir užbaigti keliones. Tuo pačiu naudotojas gali peržiūrėti savo piniginę – papildyti šią piniginę arba siųsti pinigus (dabar dar nėra pilnai integruotas Paypal sprendimas, bet ateityje jeigu bus poreikis tai bus padaryta), tai iki dabar simuliuojamas pinigų pridėjimas ir siuntimas. Naudotojas taipogi gali peržiūrėti savo profilį (papildyti savo profilio informaciją, keisti nuotrauką), tuo pačiu pridėtį automobilį (arba jį pašalinti). Svarbu paminėti, kad jeigu norima, kad šis automobilis būtų registruotas jis pirma būtinai turi būti patvirtintas administratoriaus. Naudotojas gali ir peržiūrėti savo buvusias keliones ir jas gali įvertinti jeigu nėra vairuotojas. Naudotojas tuo pačiu gali peržiūrėti nustatymus – keisti savo paskyros slaptažodį, keisti savo žemėlapio nustatymus bei ištrinti paskyra. Iš profilio dalies naudotojas gali atsijungti nuo programėlės. Pagrindiniame ekrane naudotojas gali ieškoti arba siūlyti keliones. Pagrindiniai ieškomų kelionių kriterijai – paėmimo ir paskirties taškai, vietų skaičius ir pradžios laikas. Prie ieškomų kelionių galima prisiregistuoti, susirašinėti vienam bendram kanale žinutėmis su kelionės vairuotoju, atšaukti savo registraciją prie kelionės. Siūlant keliones naudotojas gali tiesiog sukurti kelionę prie kurios gali prisiregistruoti kiti naudotojai.

Turime ir kitą programėlės aktorių – administratorių. Šis aktorius turi prieiga prie Firebase konsolės, kur gali valdyti naudotojų autentifikacijos duomenis, kelionių, profilio, mašinų informaciją. Tačiau dėl paprastumo jis nėra priskiriamas kaip programėlės aktorius, nes viskas vyksta per trečiosios šalies servisus.

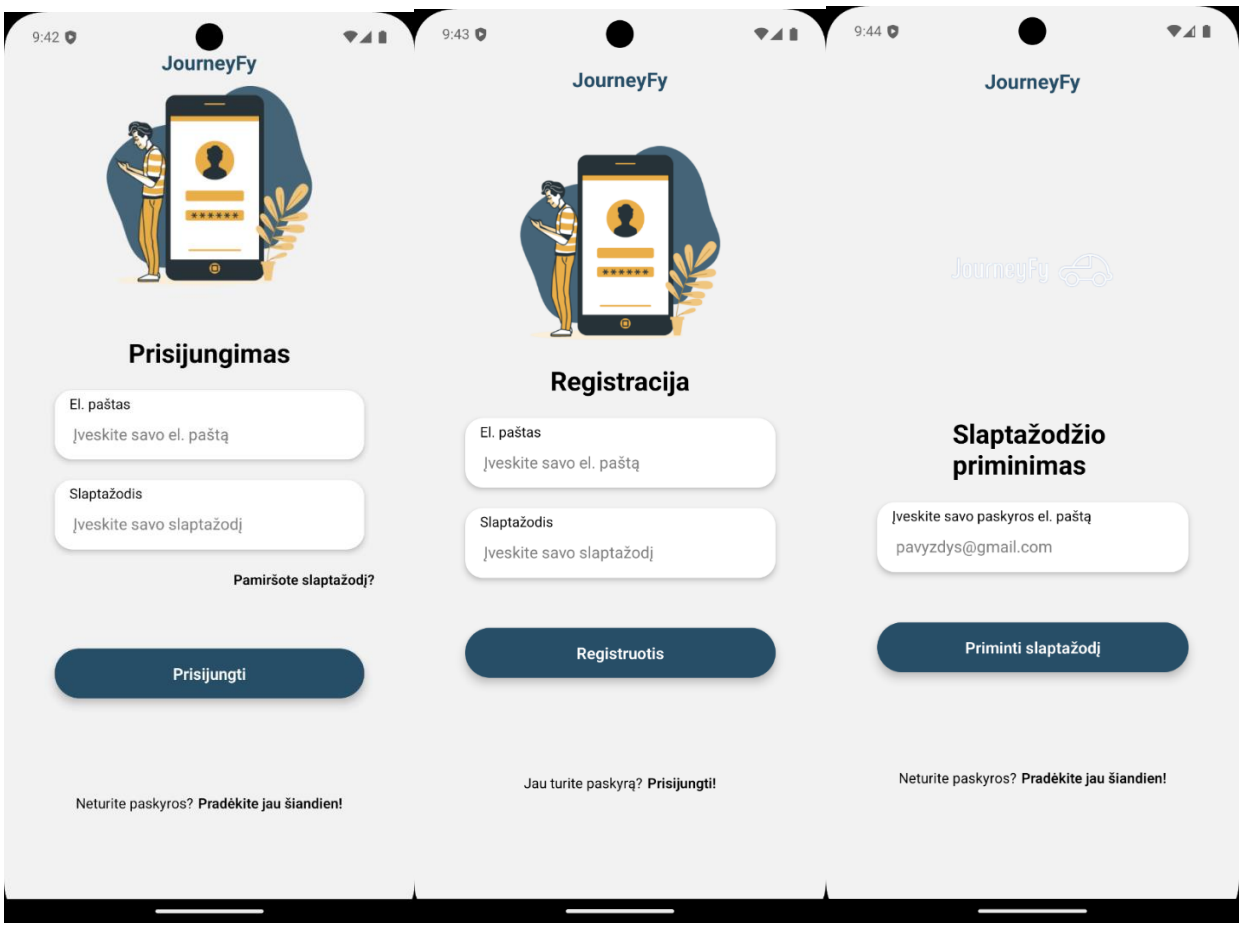
4.2. Naudotojo vadovas

Čia pateikiamas kelionių kartu programėlės „JourneyFy“ (žr. 35 pav.) naudotojo vadovas, su pagrindinėm programėlės funkcijom ir nuotraukų pavyzdžiais ir patarimais, kaip korektiškai naudotis sistema.

„JourneyFy“ kelionių tematikos programėlė

35 pav. „JourneyFy“ logotipas.

Naudotojas arba svečias gali registruotis prie programėlės, po registracijos jis būtinai privalo patvirtinti savo el. paštą su kuriuo buvo registruotasi. Po šio proceso yra galima prisijungti prie programėlės. Tuo pačiu, jeigu naudotojas užmirštą savo slaptažodį, visada gali gauti užklausą, užpildydamas slaptažodžio užmiršimo lango formą (žr. 36 pav.).

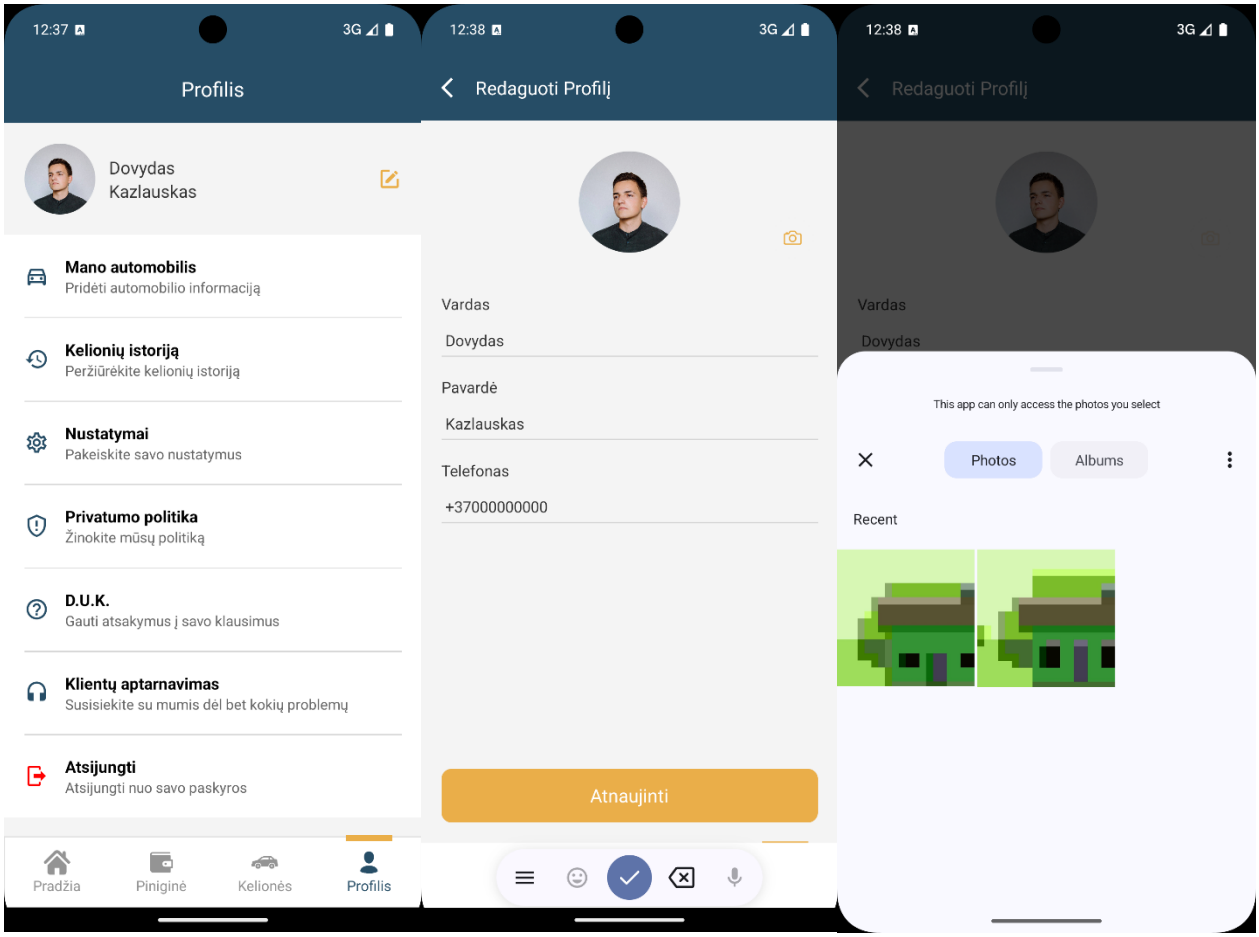


36 pav. Prisijungimo posistemio langai.

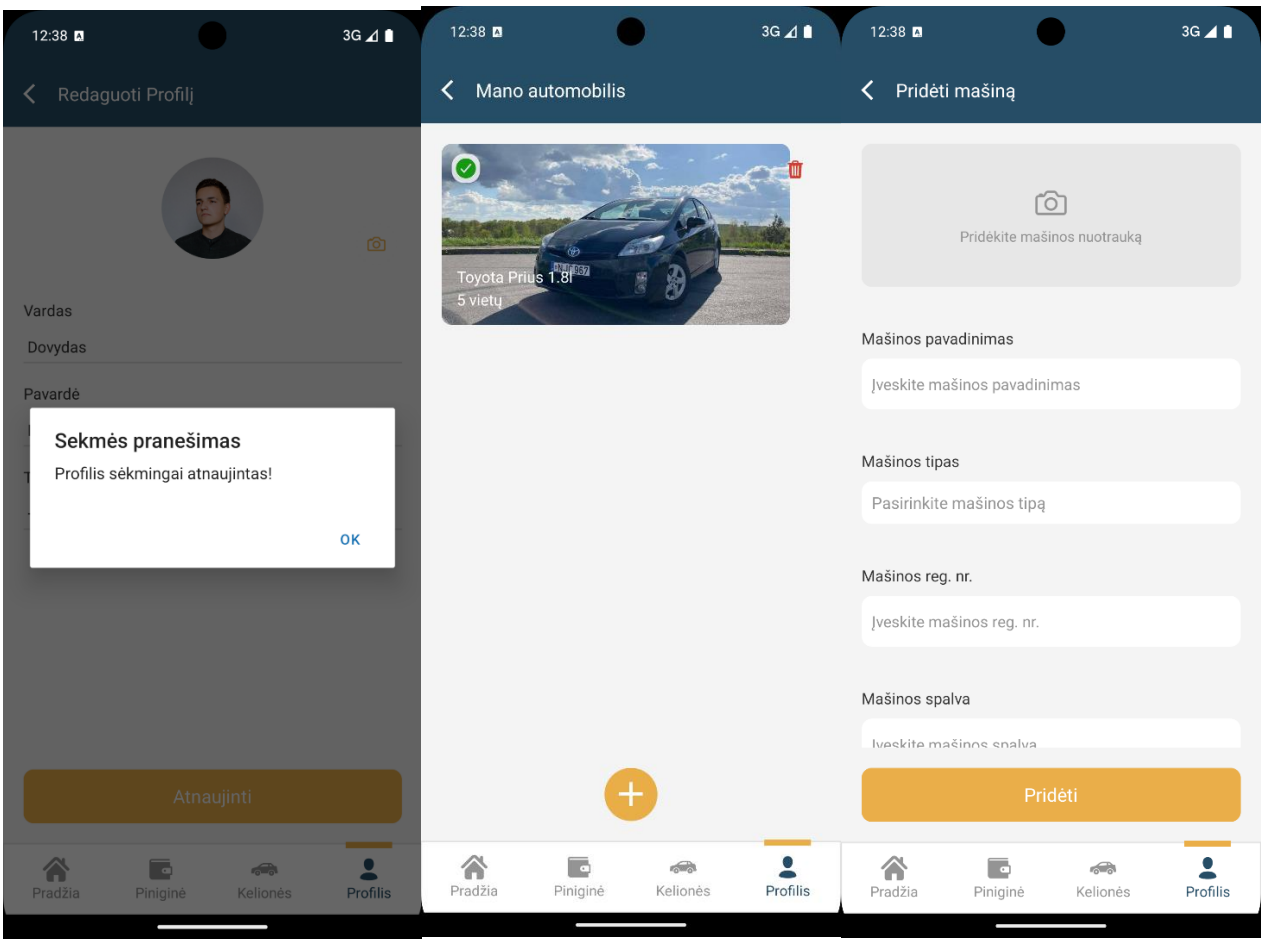
Pagrindiniai duomenys, kurie reikalingi ir išsaugomi registruojantis – el. paštas ir slaptažodis. Šie duomenys yra saugomi Firebase autentifikacijos duomenų bazėje. Naudotojas, kuris jau turi paskyrą, visada gali išsiųsti slaptažodžio keitimo užklausą į savo el. paštą, tačiau šios užklausas limituojamos.

Profilio posistemio pagrindiniai funkcionalumai:

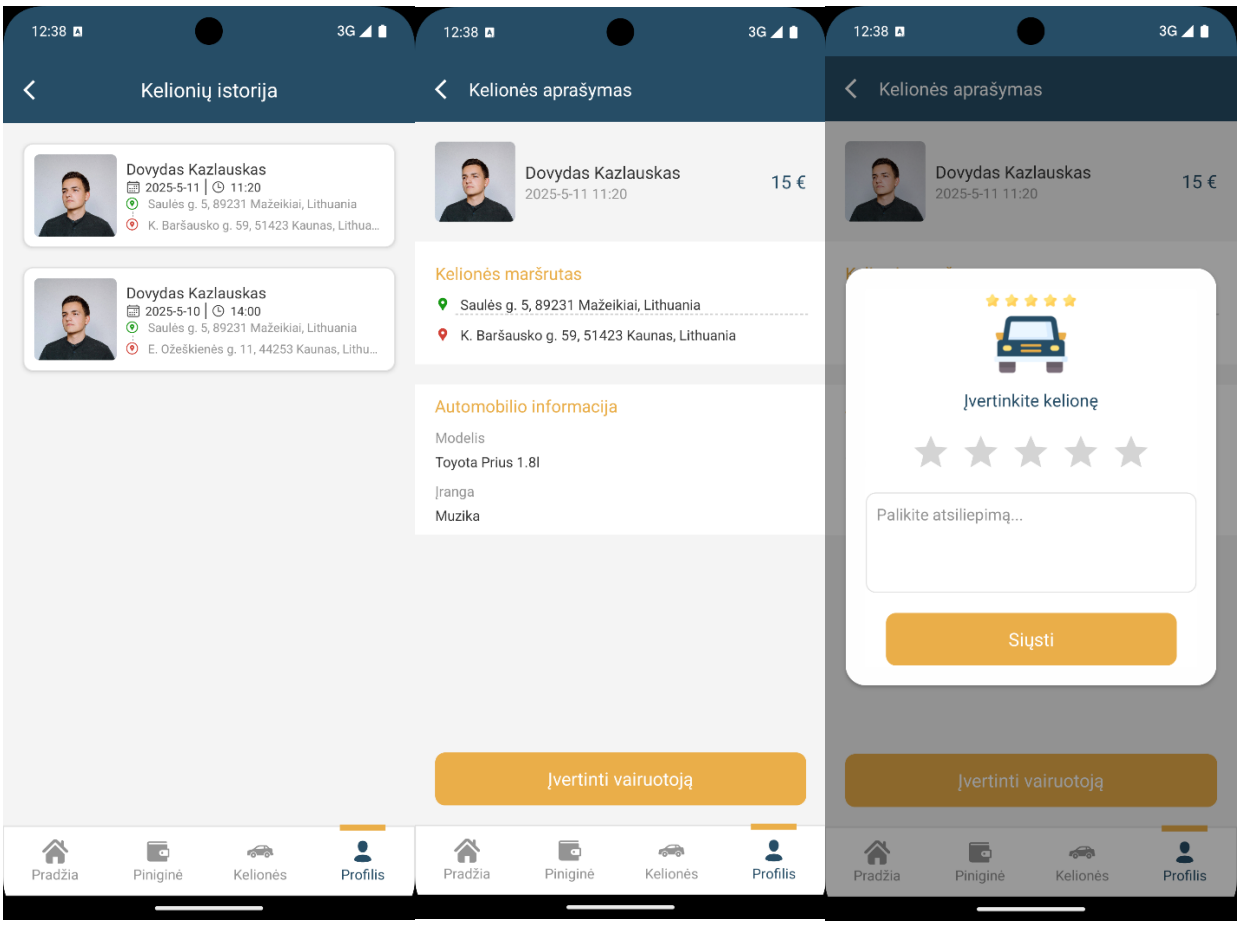
- Naudotojas gali peržiūrėti savo profilio informaciją ir keisti savo profilio duomenis, kaip vardas, pavardė, telefono numeris. Pirmą kartą registruojantis privalome užpildyti savo gimimo datą – tačiau jos vėliau modifikuoti nebegalime (žr. 37 pav.);
- Naudotojas gali peržiūrėti ir pridėti savo automobilį (šiuo metu limituojamas vienas automobilis vienam naudotojui). Šis automobilis privalo būti patvirtintas administratoriaus. Pagrindinė automobilio informacija, kurią dabar užpildome yra pavadinimas, numeriai, tipas, spalva bei papildomi patogumai. Visada galime pridėti nuotrauką, kuri suteikia daugiau konteksto. (žr. 38 pav.);
- Buvusių kelionių peržiūra – čia yra galima matyti savo registruotų arba sukurtų kelionių sąrašą, kurios jau yra pasibaigusios, čia galima jas tuo pačiu įvertinti. Vertinimas skaičiuojamas tik iš keleivių pusės, pats vairuotojas negali pridėti savo kelionės įvertinimo. Keleivis vertiną vairuotoją žvaigždutėmis (nuo 1 iki 5) pagal savo kelionės kokybę. Šie įvertinimai matomi vėliau kitiems naudotojams, prieš registruojantis arba peržiūrint vairuotojo kelionę. (žr. 39 pav.);
- Nustatymai – slaptažodžio pakeitimas, žemėlapio nustatymai, paskyros ištrinimas (žr. 40 pav.). Naudotojas gali pakeisti savo paskyros slaptažodį (užpildoma slaptažodžio keitimo forma). Žemėlapio nustatymai – ateityje planuojamas kelionių paieškos optimizavimo kriterijus, čia nustatome savo kelionės filtrus – sustojimų skaičių, galutinio taško spindulį (aplink kokiu spinduliu nuo galutinio taško galima užbaigti kelionę);
- Atsijungimas nuo paskyros (žr. 40 pav.).



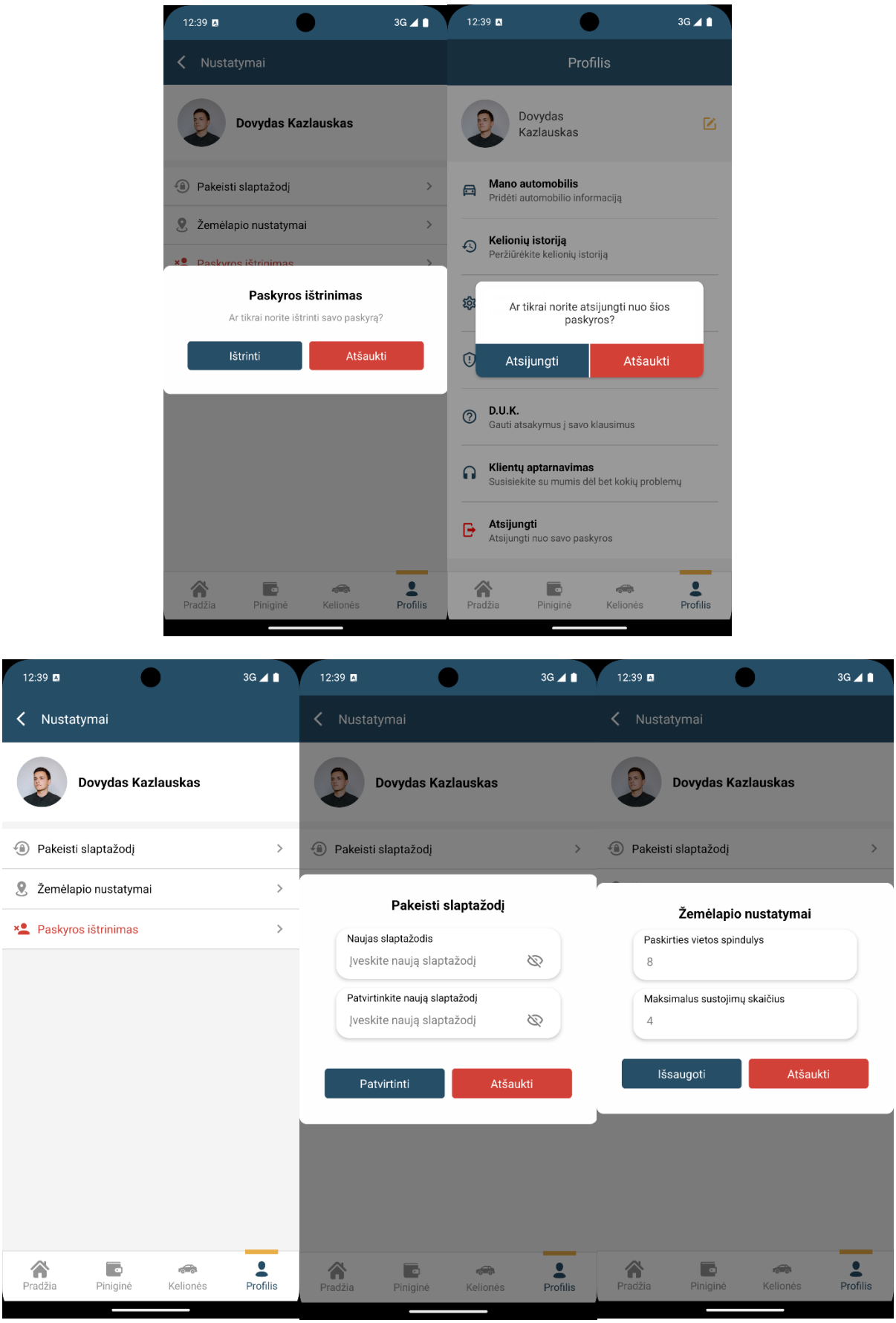
37 pav. Profilio redagavimo langai.



38 pav. Mašinos pridėjimo langai.



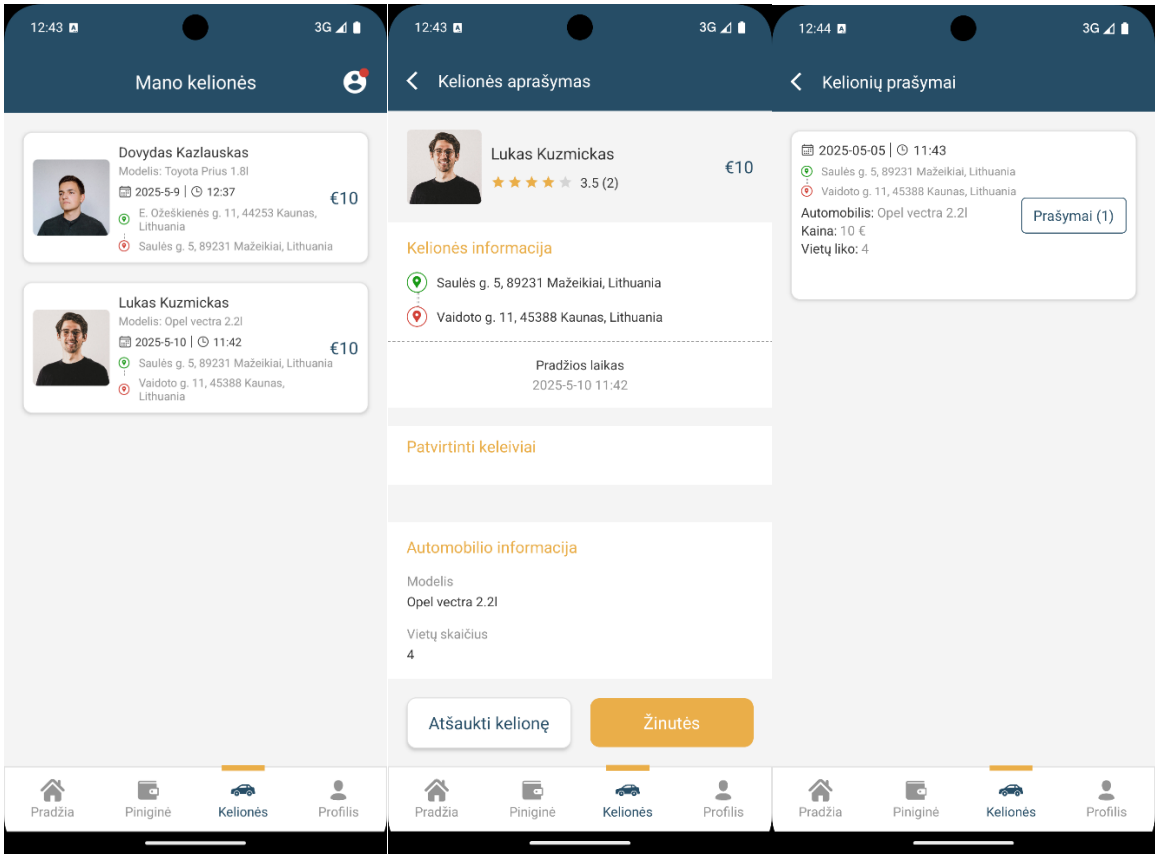
39 pav. Kelionių istorijos ir įvertinimo langai.



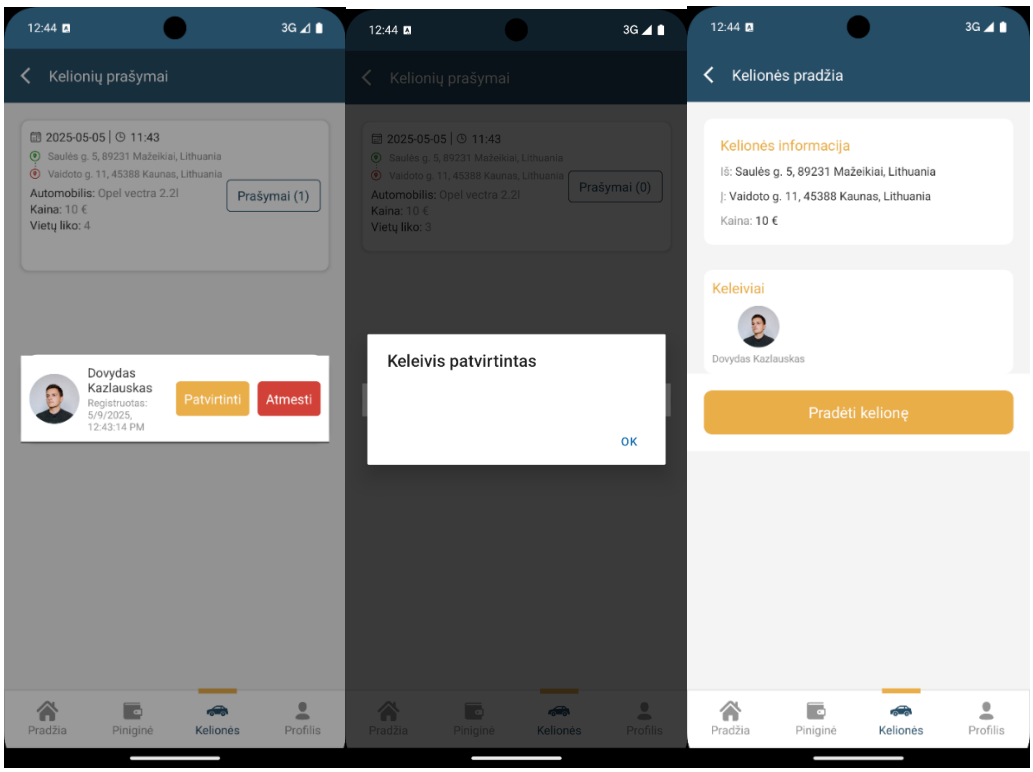
40 pav. Nustatymų langai.

Kelionės posistemio funkcionalumai:

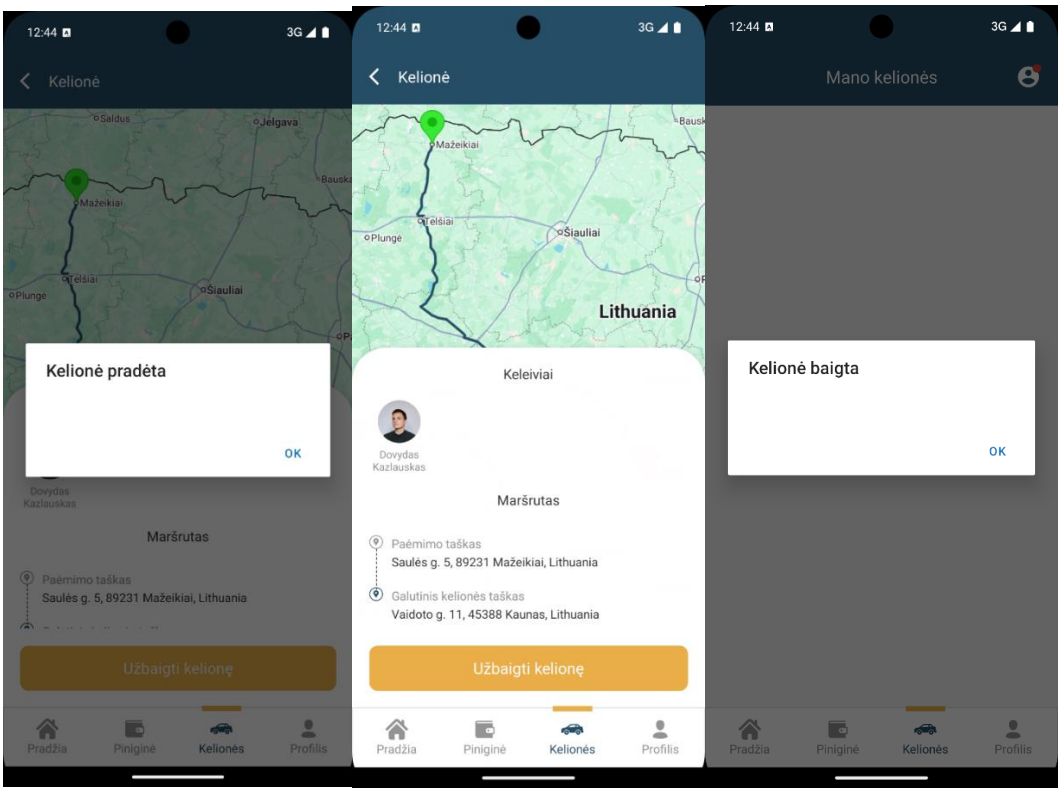
- Kelionės peržiūra – naudotojas gali matyti savo registruotas arba pridėtas keliones ir jų aprašymus (žr. 41 pav.);
- Kelionės atšaukimas arba ištrinimas – priklausant ar naudotojas yra kelionės vairuotojas ar keleivis jis gali atšaukti kelionę aplamai arba atšaukti savo kelionės rezervaciją (žr. 41 pav.);
- Kelionės žinutės ir jų siuntimas – galima komunikuoti bendrai viename pokalbių lange visiems kelionės dalyviams. Svarbu paminėti, kad gali siųsti žinutes net ir rezervacijos neatlike naudotojai – taip sudaromas šansas pasiklausinėti apie kelionę;
- Kelionių prašymai – matomi kitų keleivių kelionių prašymai. Rodomi rezervacijos prašymai vairuotojui, kuriuos jis gali atšaukti arba patvirtinti (žr. 42 pav.);
- Kelionės pradžia - čia kelionė pradedama ir yra sudaromas kelionės maršrutas, jeigu kelionė jau yra pradėta, visada grįžtama į maršruto langą. Kelionės maršrutui sudaryti naudojamas Google Maps (žr. 43 pav.);
- Kelionės pabaiga - vairuotojas visada gali užbaigti kelionę, ši kelionė neberodoma kitiems keleiviams kaip aktyvi (žr. 43 pav.).



41 pav. Kelionės peržiūros ir prašymų langai.



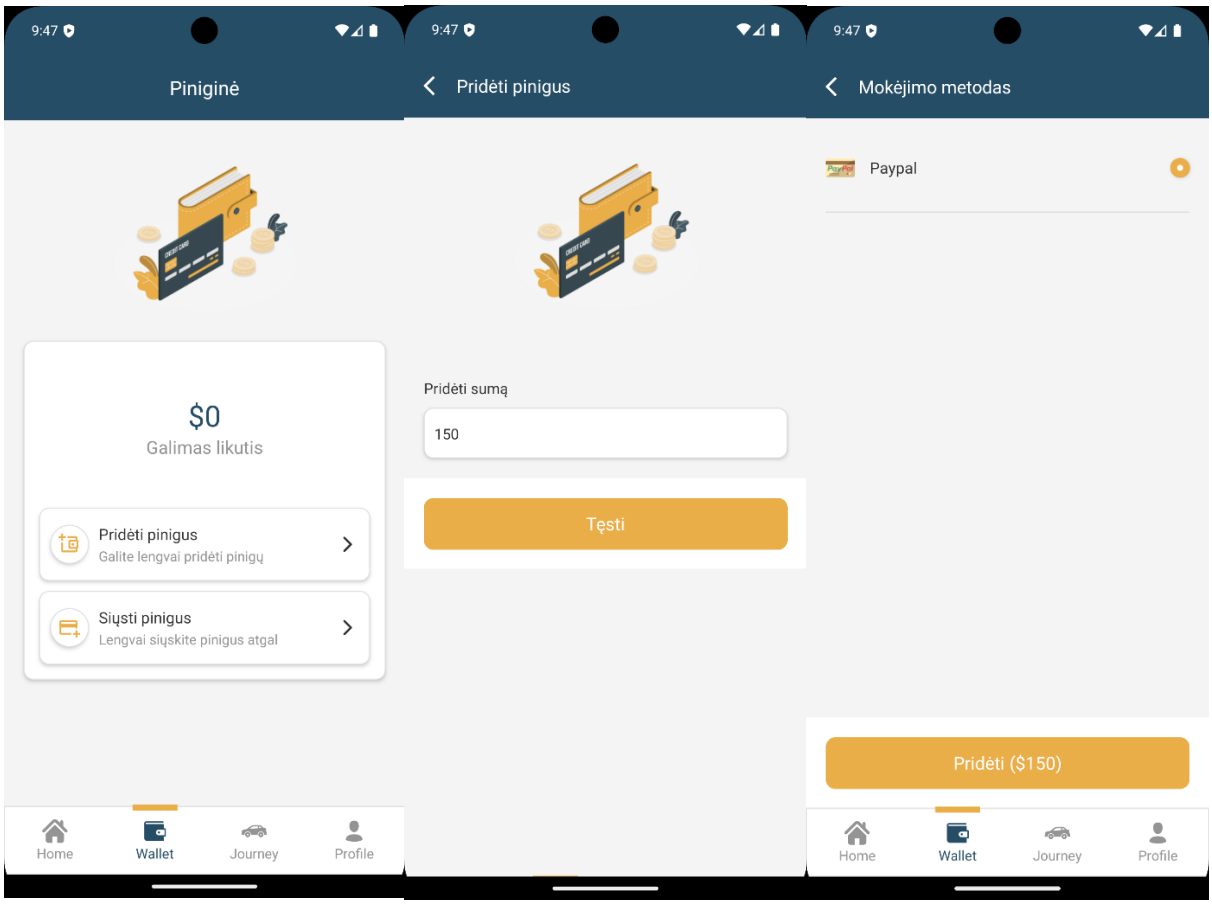
42 pav. Kelionės rezervacijos ir pradžios langai.



43 pav. Kelionės pradžios ir pabaigos langai.

Piniginės posistemio funkcionalumas (žr. 44 pav.):

- Piniginės peržiūra – naudotojas gali peržiūrėti savo piniginės likutį;
- Pinigų pridėjimas – pridėti pinigų prie savo paskyros;
- Pinigų siuntimas – siųsti pinigus iš savo paskyros.

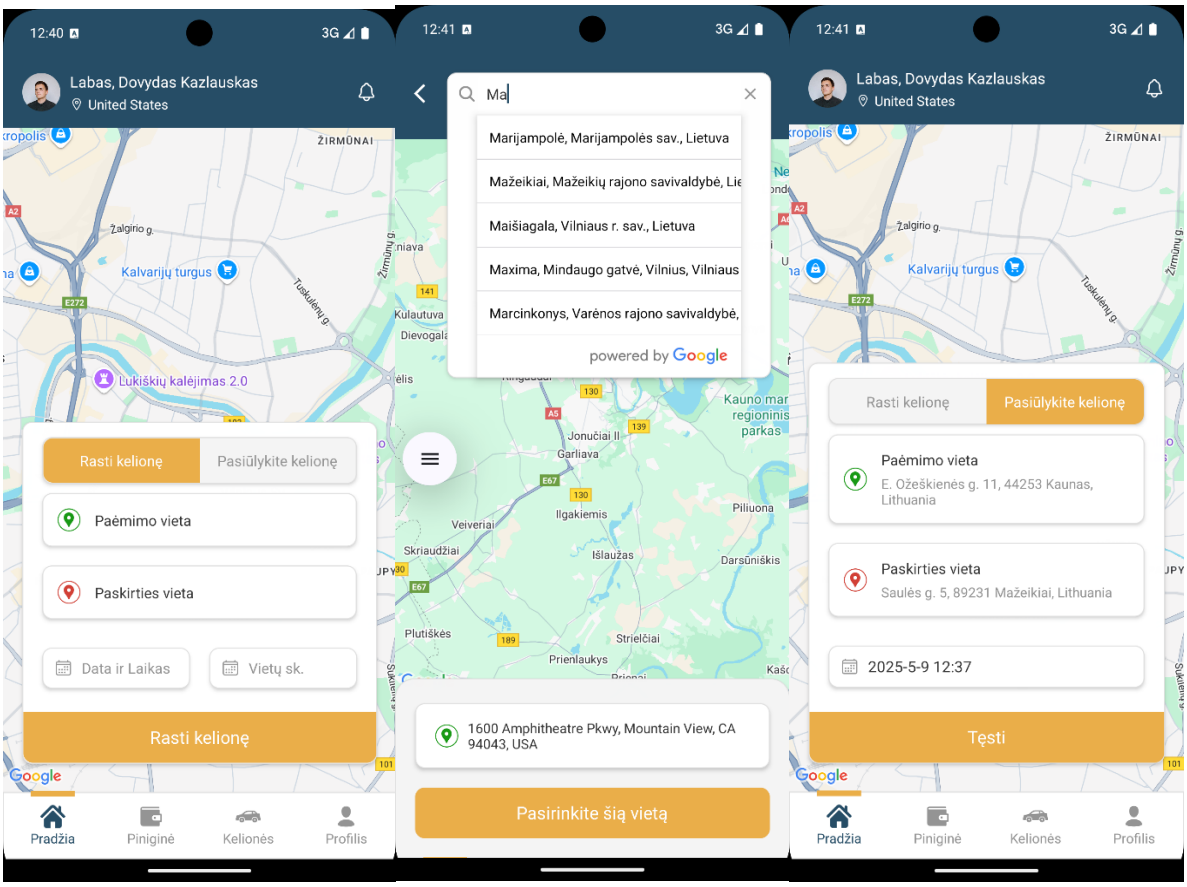


44 pav. Piniginės posistemio langai.

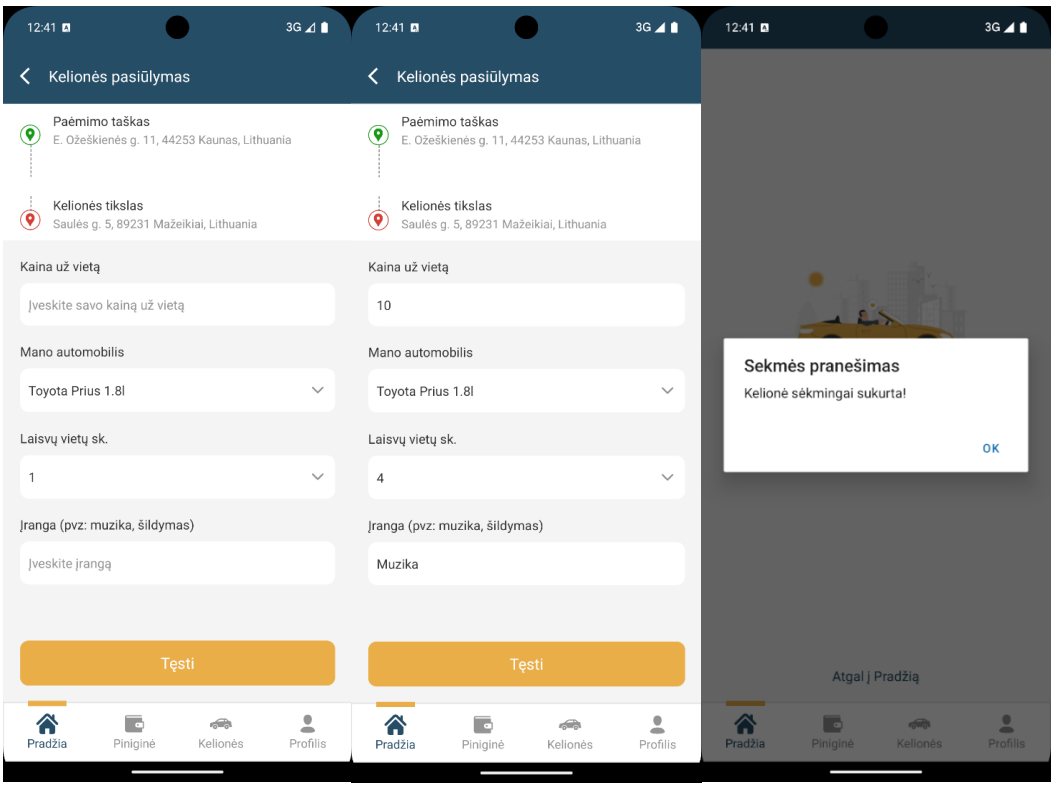
Rezervacijos posistemio funkcionalumas:

- Kelionių paieška ir kūrimas - naudotojas pasirenka pagrindinius kelionės kriterijus – paėmimo ir pabaigos taškus, vietų skaičių, pradžios laiką. Šiuos paėmimo ir pabaigos taškus pasirenka įvesdamas vietovės pavadinimą į paieškos lauką ir pasirinkdamas iš surastų variantų. Vietovei pasirinkti reikalingas patvirtinimas (žr. 45 pav., 46 pav.);
- Kelionių rezervacija (kelionių registracija ir atšaukimas), bei kelionės ištrinimas (žr. 47 pav., 48 pav.);
- Kelionių žinutės – bendras pokalbių kanalas tarp visų keleivių ir vairuotojo.

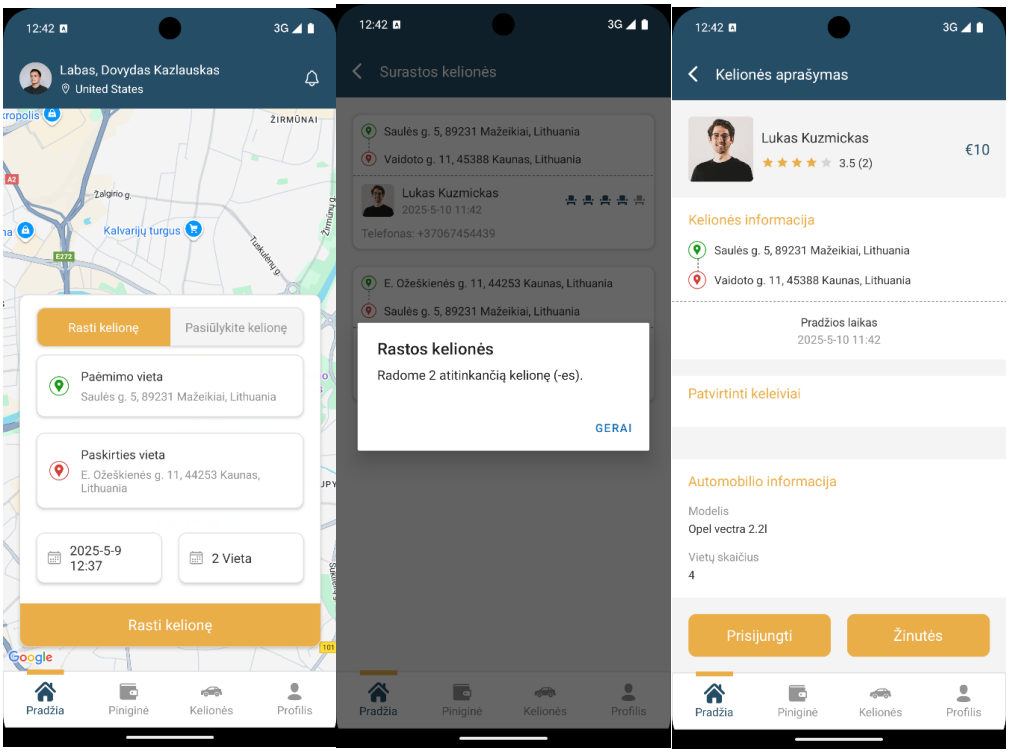
Labai svarbu paminėti, jeigu norima sukurti kelionę, būtinai reikia turėti administratoriaus patvirtintą mašiną savo paskyroje (kitaip nebus galima užbaigti kelionės pridėjimo).



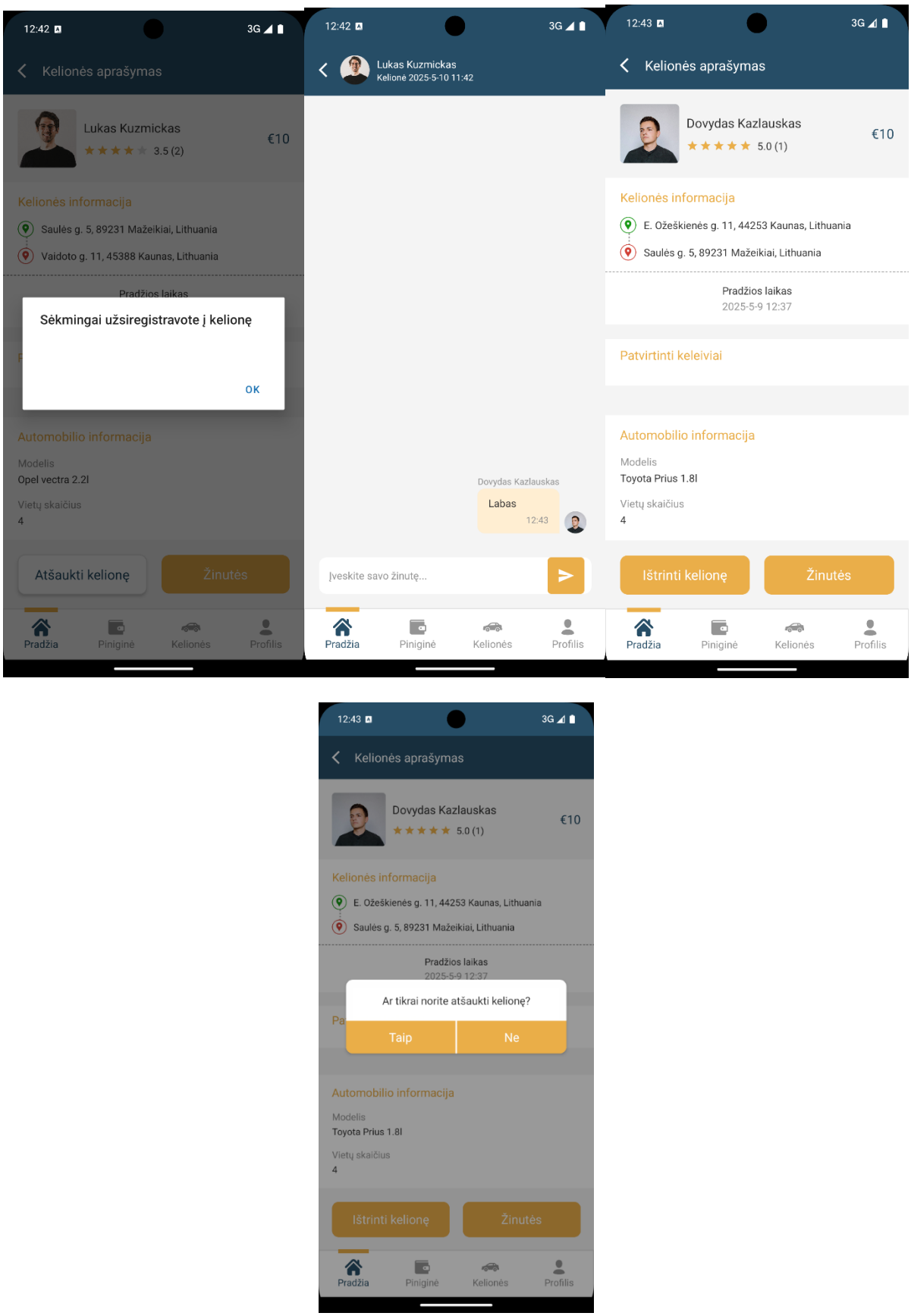
45 pav. Kelionės paieškos langai.



46 pav. Kelionės pridėjimo langai.



47 pav. Surastų kelionių langai.



48 pav. Pokalbių ir kelionės peržiūros langai.

4.3. Diegimo vadovas

Įdiegimas:

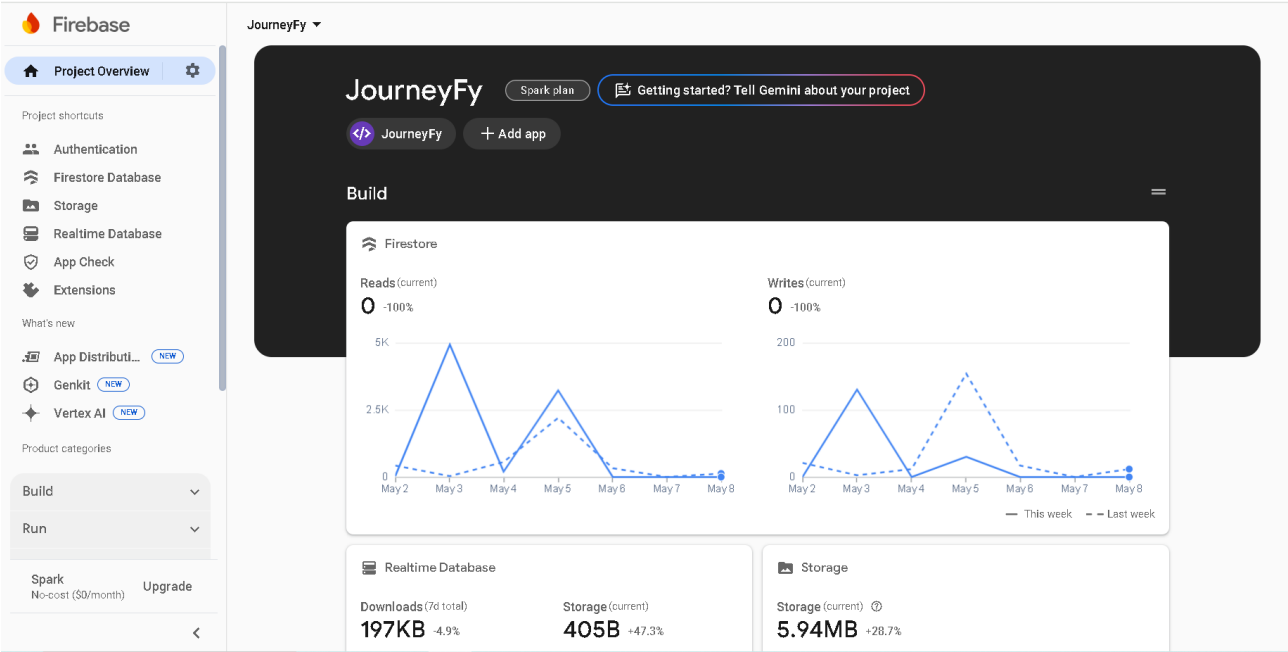
- Norėdami pradėti naudotis JourneyFy, atlikite šiuos veiksmus:
- Klonuokite Github saugyklą: git clone ...
- Eikite į projekto katalogą: cd JourneyFy
- Įdiekite priklausomybes: npm install ir surašykite .env kintamuosius.
- Paleiskite programą emuliatoriuje arba įrenginyje: npx react-native run-android arba npm start (arba naudojant Expo Go ir skanuojant QR kodą).

Technologijos:

- React Native: skirta mobiliajai programai kurti.
- „Firebase“: duomenų bazės paslaugoms, įskaitant autentifikavimą, duomenų bazę realiuoju laiku ir pranešimus.

4.4. Administravimo vadovas

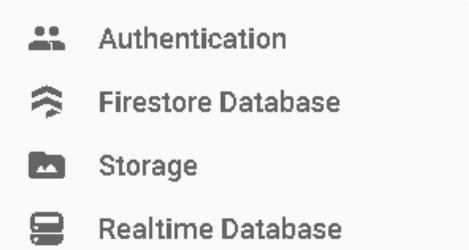
Administravimo įrankis: **Firebase console** (žr. 49 pav.).



49 pav. Firebase console puslapio vaizdas.

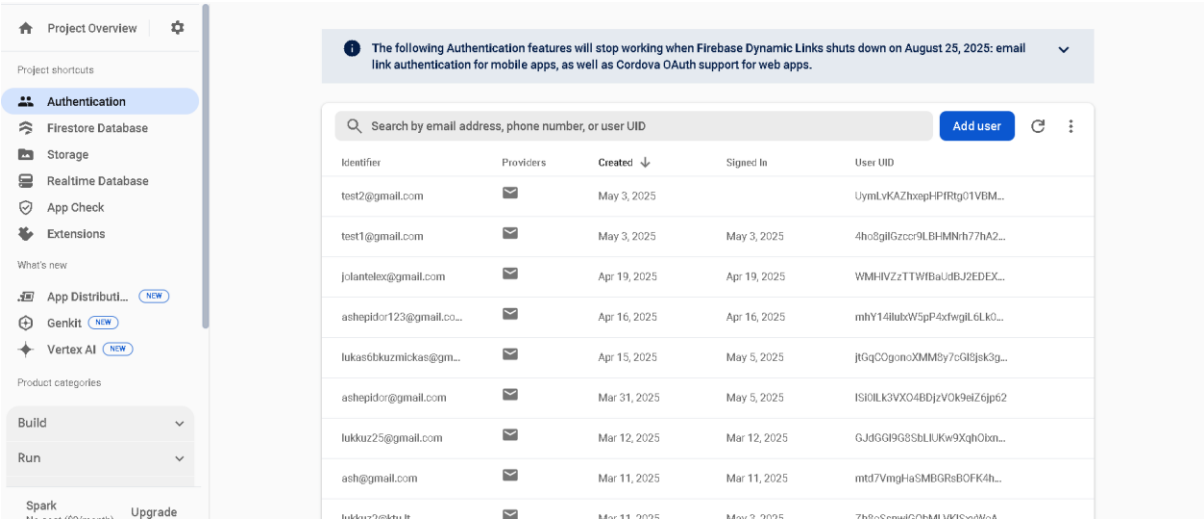
Šis įrankis padeda palaikyti „JourneyFy“ programėlės funkcionalumą ir valdyti kitus naudotojus bei jų duomenis.

Pagrindiniai naudojami Firebase servisai „JourneyFy“ programėlėje (žr. 50 pav.).



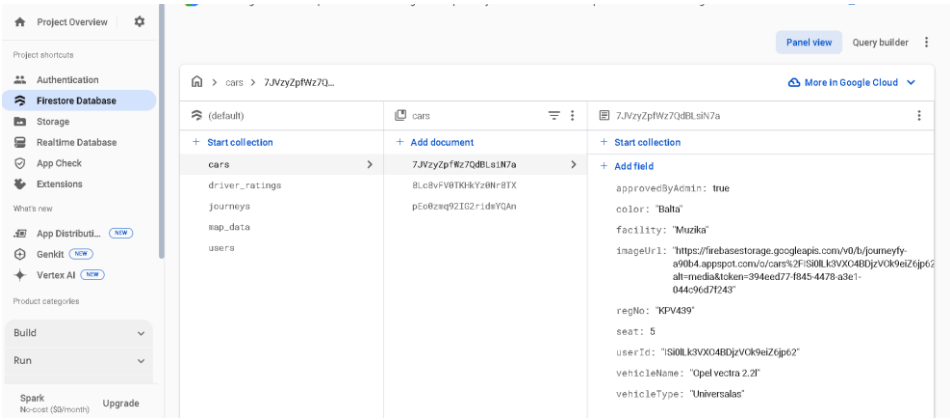
50 pav. Firebase siūlomi servisai.

Authentication – tai yra mūsų registruotų naudotojų duomenys, čia galime valdyti visus programėlės naudotojus (keisti slaptažodžius, juos ištrinti, juos sukurti) (žr. 51 pav.).



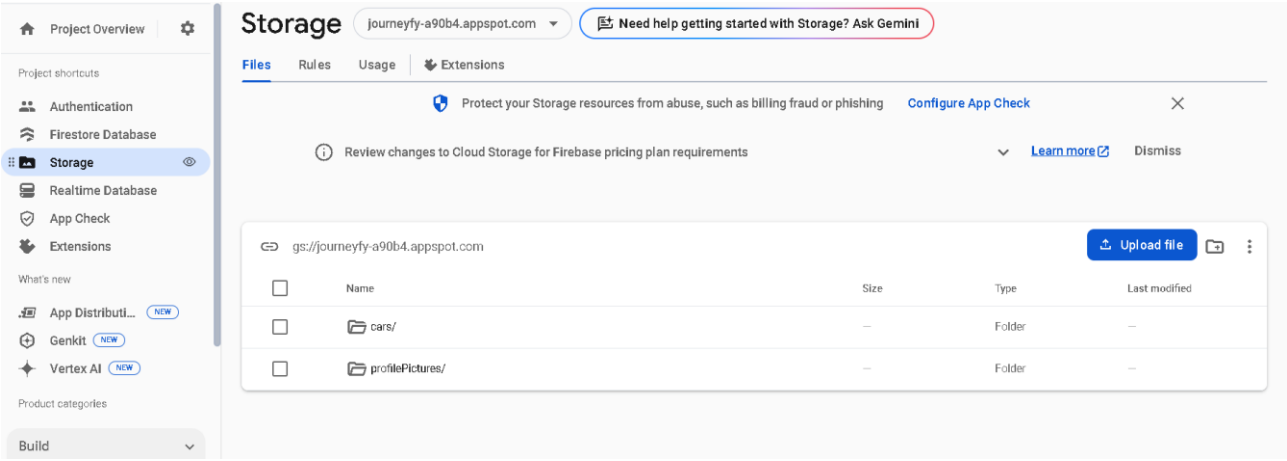
51 pav. Naudotojų autentifikacijos duomenų valdymas.

Firestore Database – mūsų pagrindinė „JourneyFy“ programėlės duomenų bazė. Čia galime valdyti visus mašinų (angl. cars), reitingų (angl. driver_ratings), kelionių (angl. journeys), žemėlapio nustatymų (angl. map_data) ir profilio informacijos (angl. users) laukus. Labai svarbu paminėti, kad būtent per čia administratorius privalo patvirtinti naudotojų mašinas, t.y. pakeisti vieno lauko reikšmę, kai naudotojas tikrai yra patvirtintas ir gali naudotis mašina savo paskyroje (žr. 52 pav.).



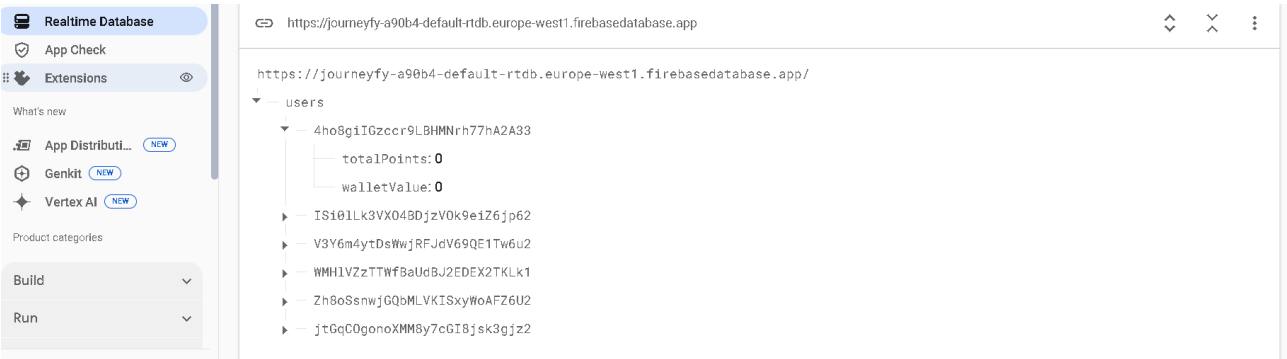
52 pav. Firestore duomenų valdymas.

Storage – čia galime valdyti visų naudotojų įkeltas profilio nuotraukas ir mašinos nuotraukas (žr. 53 pav.).



53 pav. Firebase storage duomenų valdymas.

Realtime Database – čia saugomi papildomi kiekvieno naudotojo duomenys (piniginės likutis, bendri taškai). Juos galima čia valdyti (žr. 54 pav.).



54 pav. Firebase Realtime database valdymas.

Rezultatai ir išvados

1. Egzistuojančių sprendimų analizė parodė, jog nagrinėtos kelionių programėlės neturi papildomo funkcionalumo, vieningos platformos ar pakankamos naudotojų laisvės, todėl sukurta „JourneyFy“ – programėlė, leidžianti rezervuoti ir valdyti ilgų kelionių maršrutus, derinti trumpuosius ir ilgus maršrutus, naudotis interaktyviais žemėlapiais bei individualiai ieškoti kelionių.
2. Atlikus techninių galimybių analizę buvo pasirinkta debesų pagrindu veikianti Firebase platforma, kurios realiojo laiko duomenų bazė, autentifikacija ir failų saugykla ženkliai sutrumpino vystymo laiką.
3. Sistemai testuoti buvo sukurti 52 vienetų testai, 45 integraciniai testai bei 81 naudotojo sąsajos testai, leidę pasiekti 50% kodo padengiamumą. Tai pagreitino sistemos testavimą bei užtikrino korektišką kritinių sistemos modulių veikimą viso kūrimo proceso metu.
4. Šiuo metu JourneyFy saugomas Github kodo bazėje ir yra ruošiamas publikuoti Google Play.

Literatūros sąrašas

1. Hanchuk O. V., Bondarenko O. V., Pakhomova O. V., Varfolomyeyeva I. M. Characteristics of BlaBlaCar as one of the world's ridesharing leaders [interaktyvus]. IOP Conf. Ser.: Earth Environ. Sci. 2023;1254(1):012133. doi:10.1088/1755-1315/1254/1/012133. Prieiga per: https://www.researchgate.net/publication/375305634_Characteristics_of_BlaBlaCar_as_one_of_the_world's_ridesharing_leaders [žiūrėta 2025-05-08].
2. Bolt Technology. Išbandykite „Bolt" paslaugas: universali judumo programėlė [interaktyvus]. 2025. Prieiga per: https://bolt.eu/lt-lt/ [žiūrėta 2025-05-08].
3. BlaBlaCar. Pakeleivių paieška kelionėms Europoje [interaktyvus]. 2025. Prieiga per: https://www.blablacar.com/ [žiūrėta 2025-05-08].
4. BananaCar. Pakeleivių paieška kelionėms po Lietuvą ir Europą [interaktyvus]. 2025. Prieiga per: https://bananacar.lt/ [žiūrėta 2025-05-08].
5. GoPoolar. Car Pooling App UI in React Native | BlaBlaCar Clone in React Native [internetinis vaizdo įrašas]. YouTube. 2023. Prieiga per: https://www.youtube.com/watch?v=HlPrLa4aOu0 [žiūrėta 2025-05-08].