



VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
INFORMATIKOS INSTITUTAS  
KOMPIUTERINIO IR DUOMENŲ MODELIAVIMO KATEDRA

Informacinių technologijų, 4 semestro kursinis darbas

**Mobilioji programėlė kelionės maršruto sudarymui**  
**Mobile Application for Travel Route Planning**

Atliko:  
Giedrius Dauknys

Vadovas:  
Dr. Rokas Astrauskas

Vilnius  
2024

# Turinys

<b>Santrauka</b>	<b>4</b>
<b>Summary</b>	<b>5</b>
<b>Įvadas</b>	<b>6</b>
<b>1. Šaltinių analizė</b>	<b>7</b>
1.1. Mobiliosios programėlės dizainas, kuris daro gerą poveikį vartotojų patirčiai . . .	7
1.2. Genetinio algoritmo taikymas maršruto optimizavime naudojant "Google Maps" API	7
1.3. Daugiaobjektyvės optimizacijos metodikų taikymas viešojo transporto maršrutų planavimui . . . . .	7
<b>2. Rinkos analizė</b>	<b>8</b>
2.1. "Google Maps" . . . . .	8
2.2. "Neakivaizdinis Vilnius" . . . . .	8
<b>3. Programėlės projektavimas ir argumentacija</b>	<b>9</b>
3.1. Tikslai ir paskirtis . . . . .	9
3.2. Dizainas . . . . .	9
3.3. Technologiniai sprendimai . . . . .	9
3.4. Viešojo transporto maršrutų planavimas . . . . .	10
3.5. Informacija apie objektus bei manipuliavimas jais žemėlapyje . . . . .	10
3.6. Naudotojo patirties gerinimas . . . . .	11
<b>4. Realizavimui naudojami įrankiai</b>	<b>11</b>
4.1. "Android Studio" . . . . .	11
4.2. "Google Maps API" . . . . .	11
4.3. "Google Cloud Console" . . . . .	12
<b>5. Mobiliosios programėlės realizavimas</b>	<b>12</b>
5.1. Sąrašo sudarymas . . . . .	12
5.1.1. Objektų detalesnės informacijos rodymas . . . . .	13
5.2. Maršruto funkcijos realizavimas . . . . .	14
5.3. Maršruto su autobusais funkcijos realizavimas . . . . .	14
5.3.1. Viešojo transporto duomenų integracija . . . . .	14
5.3.2. Maršruto su autobusais vizualizavimas . . . . .	15
5.3.3. Vartotojo sąveika ir informacijos atvaizdavimas . . . . .	15
5.4. Dizaino įgyvendinimas . . . . .	15
5.4.1. Pagrindinis ekranas . . . . .	15
5.4.2. Objektų pasirinkimo ekranas . . . . .	15
5.4.3. Maršruto ekranas . . . . .	16
5.4.4. Naudojimo patogumas . . . . .	16
5.4.5. Responsyvus dizainas . . . . .	16
<b>Išvados ir rekomendacijos</b>	<b>17</b>



## Santrauka

Mobilios programėlės, kurios sudaro maršrutą yra vis plačiau naudojamos. Tokios programėlės kaip "Neakivaizdinis Vilnius" [12] ar "Google Maps" [9] dažniausiai būna nemokamos ir supaprastina keliones, siūlydamos iš anksto nustatytus maršrutus ir lankytinas vietas. Nereikia pačiam ieškoti, galvoti ir planuoti - viską padaro programėlė. Šio kursinio darbo tikslas yra sukurti programėlę, kuri papildytų jau egzistuojančias tokio tipo programas įvairiais funkcionalumais. Programėlė leidžia vartotojui pasirinkti objektus, kuriuos jis nori aplankyti, šie objektai su aprašymais atsiranda žemėlapyje ir sukuriamas optimaliausias maršrutas. Programėlė taip pat, leidžia pasirinkti, kad maršrutas būtų skaičiuojamas ir parodomas žemėlapyje su atitinkamais autobusų maršrutais. Be to, programėlė efektyviai paskaičiuoja maršruto duomenis: ilgį ir laiką. Šiuos duomenis parodo vartotojui prieš pradėdant eiti maršrutą, kad vartotojas galėtų nuspresti ar sudarytas maršrutas jam nėra nei per ilgas, nei per trumpas.

Programėlė ir jos funkcionalumai kuriami išanalizavus jau esančias panašias programėles rinkoje, atsižvelgiant į jų trūkumus bei privalumus, tokias kaip "Neakivaizdinis Vilnius" ir "Google Maps".

## Summary

The use of mobile apps that make up the route are more and more widely used. Apps like "Neakivaizdinis Vilnius" [12] or "Google Maps" [9] are usually free and simplify travel by offering predefined routes and places of interest. You don't need to search, think and plan yourself - the app does everything. The aim of this coursework is to create an app that would enhance existing programs of this type with various functionalities. The app allows the user to select the objects they want to visit, then these objects appear on the map and the most optimal route is created. The app also allows for users to choose to have the route calculated and displayed on the map with the corresponding bus routes. Additionally, the app efficiently calculates route data, including length and time. This data is shown to the user before starting the route, so that the user can decide whether the route is neither too long nor too short for him.

The app and its functionalities are created after analyzing existing similar apps on the market, taking into account their advantages and disadvantages, such as "Neakivaizdinis Vilnius" and "Google Maps".

## **Įvadas**

Išmaniosios technologijos tampa vis labiau aktualios mūsų kasdieniniam gyvenime, suteikdamos galimybes greičiau ir efektyviau spręsti įvairias problemas. Daugelis žmonių savo gyvenimo dabar net neišsivaizduotų be išmaniojo įrenginio, tikriausiai ko ne kiekvienas žmogus dabar turi telefoną, kuris žmogui padeda įvairiausiais tikslais - nuo darbo iki pramogų.

Kelionių planavimas ir optimalus maršrutų sudarymas yra svarbi ir aktuali tema šiandieniniame pasaulyje. Programėlės, tokios kaip "Google Maps" [9], suteikia galimybę greitai ir patogiai rasti kelią nuo vieno taško iki kito, taip pat siūlo iš anksto nustatytus maršrutus ir lankytinas vietas. Tačiau, nepaisant jų funkcionalumo, šios programėlės turi tam tikrų ribotumų, ypač kai kalbama apie keliautojų individualius poreikius ir norą atrasti mažiau žinomas, bet įdomias vietas.

Šio kursinio darbo tikslas yra sukurti programėlę, kuri papildytų jau egzistuojančias tokio tipo programėles įvairiais funkcionalumais, suteikdama vartotojams daugiau pasirinkimo ir lankstumo planuojant keliones. Naujoji programėlė leis vartotojams pasirinkti konkrečius objektus, kuriuos jie nori aplankyti, ir šiuos objektus su jų aprašymais rodyti žemėlapyje, sukurdamą optimaliausią maršrutą. Taip pat, programėlė galės parodyti maršrutą su atitinkamais autobusų maršrutais, efektyviai apskaičiuos maršruto ilgį ir laiką bei pateiks šią informaciją vartotojui prieš pradėdant kelionę, kad jis galėtų įvertinti maršruto tinkamumą pagal savo poreikius.

Šio projekto siekiami rezultatai yra sukurti patogią ir efektyvią priemonę, kuri padėtų keliautojams atrasti naujas lankytinas vietas, pagal asmeninius poreikius. Programėlės kūrimo procesas remiasi išsamia rinkoje esančių analogiškų programėlių analize, atsižvelgiant į jų privalumus ir trūkumus. Tarp analizuojamų programėlių yra tokios kaip "Neakivaizdinis Vilnius" ir "Google Maps". Ši analizė leidžia ne tik išvengti esamų programėlių trūkumų, bet ir pasiūlyti naujas, inovatyvias funkcijas, kurios pagerins vartotojų patirtį planuojant keliones.

### **Darbo uždaviniai**

- Išanalizuoti šaltinius apie mobiliųjų programėlių dizainų gerąsias praktikas.
- Atlikti rinkos analizę, įvertinant esamų programėlių privalumus ir trūkumus, palyginti su savo programėle bei identifikuojant vartotojų poreikius.
- Pasirinkti technologijas, tinkamas naujos programėlės kūrimui, įskaitant tinkamas programavimo kalbas, duomenų saugojimo būdus ir API integraciją.
- Sukurti mobiliąją aplikaciją, įgyvendinančią nustatytus tikslus ir funkcionalumą.
- Aprašyti ir pavaizduoti darbo procesą, taip pat aplikacijos veikimą.

### **Darbo struktūra**

Darbo struktūra susideda iš dviejų dalių. Pirmoje dalyje yra supažindinama su projekto ideja ir tikslu, atliekamos šaltinių ir rinkos analizės. Antroje dalyje aprašomas projekto planavimas, įgyvendinimo procesas, paaiškinama jo architektūra ir pagrindinės funkcijos.

# 1. Šaltinių analizė

Šiame skyriuje nagrinėjami įvairūs moksliniai straipsniai, kurie analizuoja įvairius aspektus ir metodikas, susijusias su kelionių maršrutų planavimo mobiliosiomis aplikacijomis.

## 1.1. Mobiliosios programėlės dizainas, kuris daro gerą poveikį vartotojų patirčiai

Šiame straipsnyje [13] siūloma naudoti minimalistinį ir vartotojui draugišką dizainą, kuris pritrauktų naujų ir išlaikytų jau esančius vartotojus. Autoriai pabrėžia svarbą kuriant intuityvias navigacijos sistemas, aiškias ir suprantamas vizualines užuominas bei interaktyvias funkcijas, kurios padeda vartotojui lengvai planuoti savo keliones. Dažniausios klaidos programėlėje yra dėl prasto vizualaus dizaino, sunkios ir neaiškios navigacijos ir neaiškių instrukcijų.

## 1.2. Genetinio algoritmo taikymas maršruto optimizavime naudojant "Google Maps" API

Šis straipsnis [10] siūlo taikyti genetinius algoritmus maršrutų optimizavimui, siekiant rasti efektyviausius kelius, atsižvelgiant į įvairius faktorius, tokius kaip eismo sąlygos, atstumas ir kelionės laikas. Autoriai siūlo naudoti "Google Maps" API duomenis, kad algoritmas galėtų realiu laiku pritaikyti maršrutus pagal pasikeitusias sąlygas. Tai leidžia pasiekti didesnę efektyvumą ir sumažinti kelionės laiką, taip pagerinant naudotojų patirtį. Autoriai pabrėžia integruotos informacinės sistemos svarbą, kuri leidžia realiu laiku analizuoti ir optimizuoti maršrutus, atsižvelgiant į eismo sąlygas.

## 1.3. Daugiaobjektyvės optimizacijos metodikų taikymas viešojo transporto maršrutų planavimui

Šis straipsnis [14] siūlo naudoti daugiaobjektyvės optimizacijos metodikas viešojo transporto maršrutų planavimui, siekiant pagerinti transporto efektyvumą ir patogumą miesto kontekste. Siūlomos metodikos apima evoliucinius algoritmus ir daugiakriterinius sprendimų priėmimo modelius, kurie leidžia subalansuoti įvairius tikslus, tokius kaip kelionės trukmė ir autobusų atvykimo, išvykimo laikai.

## 2. Rinkos analizė

Kuriant naują programėlę, būtina atlikti išsamią rinkos analizę, siekiant užtikrinti programėlės unikalumą. Analizuojant panašias programėles, galima identifikuoti jų trūkumus, kurias galima ištaisyti savo projekte, taip pat įvertinti ir pritaikyti gerąsias funkcijas, kad būtų pagerinta vartotojų patirtis.

### 2.1. "Google Maps"

#### Aprašymas

"Google Maps" [9] yra viena populiariausių navigacijos ir žemėlapių programėlių pasaulyje, siūlanti vartotojams daugybę funkcijų, padedančių rasti maršrutus, naršyti po miestus ir atrasti vietas. Programėlės veikimo principas yra toks - atsidaręs programėlę, naudotojas gali atlikti paiešką pagal adresą, pavadinimą ar kategoriją. Vartotojas gali pasirinkti maršrutą automobiliu, pėsčiomis, dviračiu ar viešuoju transportu. Programėlėje pateikiama realaus laiko eismo informacija, padedanti išvengti kamščių ir vėlavimų. Persijungęs į paieškos režimą, gali rasti artimiausius restoranus, degalines ir kitus objektus. Vartotojai taip pat gali skaityti ir rašyti atsiliepimus apie vietas.

#### Trūkumai

- Programėlė naudoja daug mobiliųjų duomenų, ypač kai naudojama su "Street View" funkcija.
- Naujokams gali būti sunku naudotis dėl daugybės funkcijų ir nustatymų.
- Programėlė leidžia pasirinkti tik vieną objektą, jei nori matyti maršrutą su autobusų maršrutais.

### 2.2. "Neakivaizdinis Vilnius"

"Neakivaizdinis Vilnius" [12] yra Vilniaus miesto savivaldybės sukurta programėlė, skirta padėti tiek vietiniams gyventojams, tiek turistams atrasti mažiau žinomas, tačiau įdomias vietas mieste. Programėlės veikimo principas yra paprastas - atsidaręs programėlę, naudotojas gali pasirinkti teminius maršrutus, tokius kaip istoriniai, kultūriniai ar gamtiniai. Vartotojai gali peržiūrėti informaciją apie maršrutuose esančias vietas, jų istoriją ir svarbą, naudodamiesi interaktyviais žemėlapiais, kuriuose rodoma jų vietovė ir įvairios ikonos, reprezentuojančios objektus. Paspaudus ant ikonos, pateikiamas objekto pavadinimas, o paspaudus ant jo - išsami informacija, įskaitant atstumą iki objekto, nuotrauką ir aprašymą.

#### Trūkumai

- Neturi tiek pažangių navigacijos funkcijų kaip "Google Maps", pvz., realaus laiko eismo informacijos.
- Maršrutai jau yra sudaryti ir vartotojas negali jų keisti pagal savo norus.



### 3. Programėlės projektavimas ir argumentacija

Šiame skyriuje aprašomas programėlės projektavimas ir argumentacija tam tikriems sprendimams.

#### 3.1. Tikslai ir paskirtis

Programėlė buvo sukurta siekiant palengvinti vaikščiojimo po Vilnių patirtį tiek vietiniams gyventojams, tiek turistams. Vartotojai gali pasirinkti objektus, kuriuos nori aplankyti, iš pateikto sąrašo, o programėlė sugeneruoja maršrutą, kuris rodomas žemėlapyje. Taip pat programėlė turi funkciją, kuri leidžia apskaičiuoti maršrutą naudojant viešojo transporto maršrutus.

#### 3.2. Dizainas

Programėlei svarbu turėti paprastą bei gražų dizainą, kad vartotojams jis atrodytų patrauklus ir suprantamas. Svarbiausia, kad vartotojas be sunkumų rastų norimas funkcijas. Straipsnyje [13] (1.1) pabrėžiama, jog minimalistinis ir vartotojui draugiškas dizainas yra esminis veiksnys pritraukiant naujus ir išlaikant esamus vartotojus. Remiantis šia rekomendacija, intuityvios navigacijos sistemos, aiškos vizualinės užuominos ir interaktyvios funkcijos yra būtinos. Pavyzdžiui, pagrindiniame ekrane turėtų būti pateiktos pagrindinės funkcijos su dideliais ir lengvai pasiekiamais mygtukais, visi įrankiai turėtų būti aiškiai matomi ir prieinami. Tokia struktūra sumažina naudotojų klaidų tikimybę ir užtikrina, kad vartotojai galėtų lengvai ir greitai planuoti savo keliones, pasinaudodami visomis programėlės teikiamomis galimybėmis.

#### 3.3. Technologiniai sprendimai

Programėlėje naudojami "Directions API" [4] servisas ir "Nearest neighbor algoritmas" [2] dėl jų efektyvumo ir patikimumo. "Google Maps API" [5] turi pažangių optimizavimo metodų, kurie naudojami siekiant rasti geriausius maršrutus, atsižvelgiant į tokius faktorius kaip eismo sąlygos, atstumas ir kelionės laikas. Straipsnyje [10] siūloma taikyti genetinius algoritmus maršrutų optimizavimui, naudojant "Google Maps API" duomenis realiu laiku. Tačiau šiame projekte nuspręsta naudoti "Nearest neighbor algoritmą" kartu su "Directions API", be genetinių algoritmų. Šis sprendimas buvo priimtas dėl kelių priežasčių:

- **Efektyvumas ir patikimumas:** "Nearest neighbor algoritmas" kartu su "Directions API" leidžia optimaliai sudaryti maršrutus atsižvelgiant į realaus laiko eismo sąlygas, atstumą ir kelionės laiką. Tai užtikrina, kad vartotojai gaus patikimus ir efektyvius maršrutus be papildomų sudėtingų optimizavimo algoritmų.
- **Laiko ir išteklių taupymas:** Naudojant "nearest neighbor algoritmą" ir "Directions API" pašalinamas poreikis kurti ir taikyti sudėtingus optimizavimo algoritmus, kadangi šie sprendimai jau suteikia išbandytus ir veiksmingus maršrutų planavimo metodus. Tai leidžia daugiau dėmesio skirti kitoms programėlės funkcijoms ir naudotojo patirties gerinimui.
- **Integracija su viešojo transporto maršrutais:** "Directions API" taip pat leidžia integruoti viešojo transporto maršrutus, suteikiant vartotojams galimybę rinktis patogiausius kelionės būdus.

### 3.4. Viešojo transporto maršrutų planavimas

Viešojo transporto maršrutų planavimas yra svarbi programėlės funkcija, leidžianti vartotojams pasirinkti optimalų keliavimo būdą. Straipsnyje [14] (1.3) aptariama daugiaobjektinė optimizacija, naudojama viešojo transporto maršrutams planuoti. Šiame straipsnyje nagrinėjamos įvairios optimizavimo strategijos, kurios gali pagerinti viešojo transporto efektyvumą, atsižvelgiant į tokius veiksnius kaip kelionės laikas, patogumas, kaštai ir aplinkosaugos aspektai. Nors šiame projekte tiesiogiai nenaudojama daugiaobjektinė optimizacija, remiantis straipsnio rekomendacijomis, viešojo transporto maršrutų integracija atliekama naudojant "Directions API", kuris leidžia realiu laiku gauti optimaliausius maršrutus, atsižvelgiant į vartotojų poreikius ir pageidavimus.

### 3.5. Informacija apie objektus bei manipuliavimas jais žemėlapyje

Norint užtikrinti naudotojui patogų ir informatyvų objektų peržiūrėjimą, programėlėje įdiegtos šios funkcijos:

- **Objektų sąrašas:** Kadangi vartotojai pirmą kartą aplankys pasirinktus objektus, jiems reikia pateikti bent minimalią vizualinę informaciją ir aprašymą. Objektai sąrašė vaizduojami su paveikslėliais, kad vartotojas iškart matytų, kas jam atrodo gražu ir įdomu, taip pat vartotojas gali pasiskaityti plačiau apie kiekvieną objektą. Tai padeda greičiau apsispręsti ir pasirinkti lankytinas vietas.
- **Objektų filtras:** Vartotojas turi galimybę matyti visus objektus, kurie sistemos buvo surasti. Objektai yra sufiltruoti į kategorijas, tokias kaip: bažnyčios, paminklai, muziejai ir panašiai. Tai leidžia vartotojui lengviau rasti ir pasirinkti ji dominančias vietas pagal jų tipą, sutaupant laiko ir pastangų.
- **Objektų vieta žemėlapyje:** Pasirinkus objektus iš sąrašo, jie iškart rodomi žemėlapyje, nurodant jų tikslias vietas. Tai padeda vartotojui geriau orientuotis ir suprasti, kur yra pasirinkti objektai, palengvinant planavimą ir navigaciją.
- **Detalesnė informacija apie objektą:** Vartotojui norint sužinoti daugiau apie objektą, jis gali paspausti ant jo markerio žemėlapyje ir bus pateikta detalesnė informacija apie objektą. Tai suteikia galimybę vartotojui susidaryti išsamesnį vaizdą apie lankytiną vietą ir priimti pagrįstą sprendimą, ar verta ją aplankyti.
- **Pradėti maršrutą:** Paspaudus mygtuką "Pradėti maršrutą", markeriai pasikeičia į sunumeruotas ikonėles, kad vartotojui būtų lengviau suprasti, kuriuo eiliškumu reikia aplankyti objektus. Tai padeda aiškiau suvokti maršruto eigą ir išvengti painiavos, ypač lankant daug objektų.
- **Naujų objektų pridėjimas ir senų objektų išėmimas iš maršruto:** Vartotojui sužinojus visą sistemos turimą informaciją apie objektą, jis gali nuspręsti, ar nori šį objektą įtraukti į maršrutą ar išimti iš jo. Tam tereikia pažymėti arba nužymėti atitinkamą check box'ą. Tai suteikia lankstumo planuojant kelionę ir leidžia vartotojui lengvai koreguoti maršrutą pagal savo pageidavimus.
- **Maršruto perskaičiavimas:** Vartotojui atliekiant pakeitimus maršrute, įtraukiant ar išimant objektus, maršrutas automatiškai perskaičiuojamas atsižvelgiant į naują pasirinkimą. Tai

užtikrina, kad vartotojas visada turės optimalų maršrutą, atitinkantį jo poreikius ir norus, ir nereikės rankiniu būdu keisti maršruto planavimo.

### 3.6. Naudotojo patirties gerinimas

Siekiant užtikrinti gerą naudotojo patirtį, buvo atlikti keli testavimo etapai, įtraukiantys realius vartotojus. Gauti atsiliepimai padėjo tobulinti programėlės dizainą ir funkcionalumą, kad ji būtų kuo patogesnė ir efektyvesnė.

## 4. Realizavimui naudojami įrankiai

Šiame skyriuje aptariami įrankiai ir technologijos, naudotos programėlės kūrimui, bei jų pasirinkimo priežastys.

### 4.1. "Android Studio"

Programėlės kūrimui buvo pasirinkta "Android Studio" [3] – oficiali integruota kūrimo aplinka skirta Android programėlėms kurti. Android Studio yra pagrįsta IntelliJ IDEA [11] ir suteikia platų įrankių rinkinį, leidžiantį programuotojams kurti, testuoti ir diegti programas.

#### Privalumai naudojant "Android Studio":

- **Visapusiškas įrankių rinkinys:** "Android Studio" teikia išsamų įrankių rinkinį, apimantį visus programėlės kūrimo etapus nuo projektavimo iki testavimo ir diegimo.
- **Integruota su "Git" [7] versijų valdymo sistema:** Tai leidžia lengvai sekti kodo pakeitimus, bendradarbiauti su kitais programuotojais ir valdyti programinės įrangos versijas.
- **Kodas rašomas Java kalba:** Java yra viena iš populiariausių ir plačiausiai naudojamų programavimo kalbų pasaulyje, ypač tinkama Android programų kūrimui dėl jos saugumo, stabilumo ir gausios bendruomenės.

### 4.2. "Google Maps API"

Nors yra galimybė kurti savo duomenų bazes, kurios saugotų maršrutų informaciją ir kitas su programėlės funkcionalumu susijusias duomenis, šis procesas reikalauja daug laiko ir pastangų. Vietoj to, naudojantis API (angl. *Application Programming Interface*) galima greitai ir efektyviai gauti visą reikiamą informaciją, nes API teikia išbandytus ir veiksmingus sprendimus. Tai leidžia sutaupyti laiko ir išteklių, kurie gali būti skirti kitoms programėlės funkcijoms tobulinti. Programėlėje naudojamas "Google Maps API" [5] ir jo komponentai: "Maps SDK for Android" [6] ir "Directions API" [4]. Šie API suteikia galimybę integruoti Google žemėlapius ir maršrutų planavimo funkcionalumą į programėlę.

**"Maps SDK for Android":** Tai API, kuris leidžian Android programėlėms naudoti Google žemėlapius, įskaitant visas interaktyvias funkcijas, kaip žemėlapių rodymas, markeriai, formų piešimas ant žemėlapių ir kt.

**"Directions API":** Šis API suteikia galimybę gauti maršrutų informaciją, įskaitant važiavimo, vaikščiojimo, dviračių ir viešojo transporto maršrutus. "Directions API" leidžia skaičiuoti optimaliausius maršrutus, atsižvelgiant į realaus laiko eismo sąlygas. Pasirinkas būtent šis API dėl to, nes jam galima paduoti įvairius vietos parametrus - pavadinimą, adresą ar koordinates.

#### **Privalumai naudojant "Google Maps API":**

- **Patikimumas ir tikslumas:** "Google Maps" [9] yra viena iš labiausiai naudojamų ir patikimų žemėlapių paslaugų, siūlanti tikslią ir nuolat atnaujinamą informaciją.
- **Platus funkcionalumas:** API suteikia daug funkcijų, įskaitant įvairių maršrutų tipų skaičiavimą, vietų paiešką ir atstumų skaičiavimą, todėl galima sukurti labai funkcionalią ir vartotojui patogią programėlę.

### **4.3. "Google Cloud Console"**

Programėlė ir jos duomenys yra saugomi "Google Cloud Console" [1] – platforma, kuri teikia įvairias paslaugas, nuo duomenų saugojimo iki serverių valdymo.

#### **Privalumai naudojant "Google Cloud Console":**

- **Mastelio keitimas ir našumas:** "Google Cloud" platforma leidžia lengvai keisti saugomų duomenų ir apdorojimo galimybes priklausomai nuo poreikių, užtikrinant aukštą našumą ir patikimumą.
- **Saugumas:** "Google Cloud" teikia aukšto lygio saugumo priemonės, įskaitant duomenų šifravimą, prieigos valdymą ir saugumo auditus.
- **Integracija su kitomis "Google" [8] paslaugomis:** "Google Cloud Console" lengvai integruojama su kitomis Google paslaugomis, kaip "Google Maps" API, todėl duomenų mainai tarp paslaugų yra greiti ir efektyvūs.

Šie įrankiai ir technologijos buvo pasirinkti dėl jų patikimumo, funkcionalumo, saugumo ir lengvo integravimo. Jie užtikrina, kad programėlė veiks efektyviai, patikimai ir atitiks vartotojų poreikius, suteikdama jiems sklandžią ir patogią naudotojo patirtį.

## **5. Mobiliosios programėlės realizavimas**

Šiame skyriuje aptariamas mobiliosios programėlės kūrimo procesas.

### **5.1. Sąrašo sudarymas**

Pirmasis veiksmas, kurį vartotojas turi atlikti, yra pasirinkti patinkančius objektus, kuriuos jis norėtų aplankyti. Kad vartotojui būtų lengviau orientuotis dideliame sąraše, visi objektai yra logiškai suskirstyti į grupes ir rodomi abėcėlės tvarka. Iš viso yra 10 grupių:

- **Akmenys** (angl. stones) – tai kategorija, į kurią įeina įvairūs istoriniai ir gamtiniai akmenys bei jų rinkiniai.

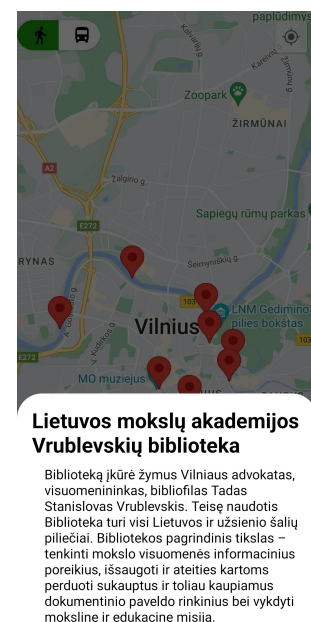
- **Bažnyčios** (angl. churches) – tai kategorija, apimanti įvairias bažnyčias ir religinius pastatus.
- **Bibliotekos** (angl. libraries) – tai kategorija, į kurią įeina viešosios ir specializuotos bibliotekos.
- **Fontanai** (angl. fountains) – tai kategorija, apimanti įvairius dekoratyvinius ir funkcionalius fontanus.
- **Muziejai** (angl. museums) – tai kategorija, į kurią įeina visų įmanomų rūšių muziejai.
- **Paminklai asmenybėms** (angl. monuments to personalities) – tai kategorija, skirta paminklams, pastatytiems pagerbti garsias asmenybes.
- **Panoramų vietos** (angl. viewpoints) – tai kategorija, apimanti vietas, iš kurių atsiveria įspūdingos panoramos.
- **Skulptūros** (angl. sculptures) – tai kategorija, į kurią įeina įvairios skulptūros ir meniniai objektai.
- **Teatrai** (angl. theaters) – tai kategorija, apimanti įvairius teatrus ir scenos menų vietas.
- **Tiltai** (angl. bridges) – tai kategorija, skirta įvairiems tiltams, turintiems istorinę ar architektūrinę reikšmę.

Objektų pasirinkimas yra itin paprastas net ir pirmą kartą programėlę naudojantiems vartotojams. Reikia tik pažymėti atitinkamą langelį šalia norimo objekto, ir jis bus įtrauktas į maršrutą. Vartotojui pasirinkus objektus ir grįžus į pagrindinį programėlės langą, žemėlapyje bus rodomi visi pasirinkti objektai.

#### 5.1.1. Objektų detalesnės informacijos rodymas

Paspaudus ant bet kurio markerio žemėlapyje, iš apačios atsivers langas su objekto pavadinimu ir detaliu aprašymu. Šiame lange taip pat yra galimybė paspausti ant objekto pavadinimo, kad programėlė automatiškai nukreiptų į sąrašo vietą, kur tas objektas yra, taip padedant vartotojui lengviau keisti ir koreguoti maršrutą (žr. 1 pav.).

Be to, šia informaciją galite pasiekti ir prieš pasirinkdami objektą, sąrašė paspaude ant informacijos ikonėlės. Programėlėje dar yra funkcionalumas "Atžymėti visus checkbox'us", kuris leidžia vartotojui pradėti maršruto kūrimą iš naujo, jei jis nori išvalyti visus pasirinktus objektus ir pradėti nuo pradžių.



1 pav. Informacijos langas

## 5.2. Maršruto funkcijos realizavimas

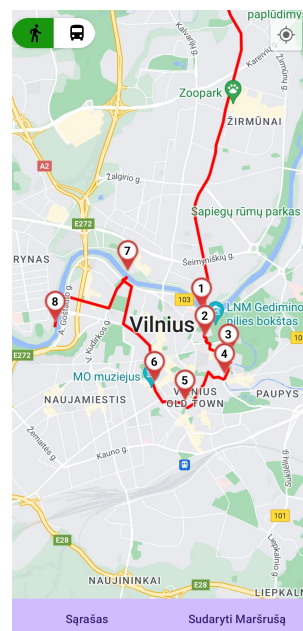
Pagrindinė programėlės funkcija yra sudaryti vartotojui maršrutą iš pasirinktų objektų (žr. 2 pav.). Ši funkcija suteikia galimybę efektyviai ir patogiai planuoti keliones, leidžiant vartotojui optimizuoti savo laiką ir pamatyti norimus objektus kuo patogesne tvarka.

Pasirinkus norimus objektus iš sąrašo, vartotojas gali pradėti maršrutą, spausdamas mygtuką "Pradėti maršrutą". Programėlė surenka visų pasirinktų objektų koordinates ir sudeda jas į masyvą. Taip pat, išsaugoma vartotojo vietos koordinatė, kuri bus pradinis maršruto taškas.

Surinkus visų objektų koordinates, masyvas yra pertvarkomas naudojant tam tikrą logiką: naudojantis "Directions API" [4], pirmiausia ieškoma artimiausio objekto nuo vartotojo vietos. Radus artimiausią objektą, sekantis ieškomas objektas yra tas, kuris yra artimiausias jau rastam objektui. Ši logika tęsiama tol, kol visi objektai yra įtraukiami į maršrutą. Toks metodas užtikrina optimalų maršruto sudarymą, leidžiant vartotojui per trumpiausią laiką aplankyti visus norimus objektus.

Radus optimalų maršrutą, rezultatas perduodamas kitai funkcijai, kuri apdoroja po du objektus, pradedant nuo vartotojo vietos ir baigiant paskutiniu objektu. Žemėlapiio žymekliai atnaujinami į sunumeruotus žymeklius, kad vartotojas galėtų lengviau orientuotis, į kurią objektą eiti pirmiausia. Numeracija padeda vizualiai suvokti maršruto seką ir lengviau sekti numatytą kelią.

Galiausiai, maršrutas perduodamas funkcijai *addPolylinesToMap*, kuri vizualizuoja maršrutą raudona linija. Ši linija aiškiai nurodo kelionės kelią žemėlapyje, padedant vartotojui lengviau sekti maršrutą ir nesuklysti kelionės metu.



2 pav. Sudarytas maršrutas

## 5.3. Maršruto su autobusais funkcijos realizavimas

Maršruto su autobusais funkcija yra esminė, kai vartotojai nori keliauti viešuoju transportu. Ši funkcija integruoja viešojo transporto tvarkaraščius bei stotelių vietas. Tai suteikia vartotojams galimybę greičiau aplankyti visus objektus jų sąrašė.

### 5.3.1. Viešojo transporto duomenų integracija

Pirmasis žingsnis realizuojant maršruto su autobusais funkciją yra viešojo transporto duomenų integracija į programėlę. Šiam tikslui naudojamas "Google Maps API" [5] komponentas "Directions API" [4], kuris teikia viešojo transporto maršrutų ir tvarkaraščių informaciją. Taip pat, leidžia gauti išsamią informaciją apie keliones viešuoju transportu, įskaitant autobusų numerius ir stotelių pavadinimus.

Programėlė gauna maršruto viešuoju transportu informaciją iš "Directions API" atsakymo ir ištraukia detales, tokias kaip autobuso numeris, išvykimo ir atvykimo stotelės, ir parodo šią informaciją vartotojui. Kiekvienas žingsnis apdorojamas, norint išgauti konkrečią informaciją apie viešąjį transportą.

### 5.3.2. Maršruto su autobusais vizualizavimas

Gautą maršrutą su autobusais reikia tinkamai vizualizuoti žemėlapyje, kad vartotojas galėtų lengvai sekti kelionę. Vizualizacija apima tiek pėsčiųjų, tiek viešojo transporto etapus.

Programa vizualizuoja maršrutą žemėlapyje, naudodamas "Directions API" gautą informaciją. Viešojo transporto etapai žymimi žalia linija, o pėsčiųjų dalis – raudona linija (žr. 3 pav.). Kiekvienas maršruto žingsnis apdorojamas ir atvaizduojamas, kad vartotojas galėtų lengvai sekti savo kelionę.



### 5.3.3. Vartotojo sąveika ir informacijos atvaizdavimas

Kad vartotojas galėtų lengviau sekti maršrutą ir su- prasti kelionės etapus, svarbu pateikti aiškią ir suprantamą informaciją apie kiekvieną maršruto segmentą. Vartotojas gali spustelėti bet kurį žymeklį žemėlapyje, kuris pateiks informaciją apie konkrečią stotelę ir autobusą, į kurį reikia įlipti.

Be to, sudarius maršrutą, atsiranda papildomas mygtukas. Šis mygtukas leidžia vartotojui atidaryti informacinį langą, kuriame pateikiama išsami informacija apie kelionę: koku autobusu reikia vykti, iš kurios stotelės išvykti ir į kurią stotelę atvykti. Tai suteikia vartotojui aiškią ir lengvai suprantamą informaciją, padedančią sklandžiai sekti maršrutą.

3 pav. Sudarytas maršrutas su autobu-  
sais

## 5.4. Dizaino įgyvendinimas

Dizainas yra viena svarbiausių programėlės dalių, nes jis užtikrina patogų ir intuityvų naudojimąsi. Kuriant dizainą buvo siekiama, kad vartotojo sąsaja būtų estetiška, lengvai suprantama ir funkcionali. Toliau aprašyti pagrindiniai dizaino elementai ir jų įgyvendinimas.

### 5.4.1. Pagrindinis ekranas

Pagrindinis ekranas yra pirmasis vartotojo matomas vaizdas, kuris suteikia bendrą programėlės veikimo supratimą. Šiame ekrane vartotojas gali matyti žemėlapi su žymekliais, atitinkančiais pasirinktus objektus. Viršutinėje ekrano dalyje yra mygtukas, kuris leidžia vartotojui pasirinkti kaip jis nori įveikti maršrutą, pėsčiomis ar su autobusais. Apačioje yra mygtukai "Sąrašas" ir "Pradėti maršrutą".

### 5.4.2. Objektų pasirinkimo ekranas

Objektų pasirinkimo ekrane vartotojas gali pasirinkti norimus aplankyti objektus iš skirtingų kategorijų. Kiekviena kategorija turi savo pavadinimą, padedantį vartotojui greitai ir lengvai rasti reikiamą objektą. Kategorijos yra pateiktos pagal abėcėlę, kad būtų lengviau naršyti. Pasirinkti objektai yra pažymėti varnelėmis, o vartotojas gali lengvai juos pridėti arba pašalinti iš maršruto.

#### **5.4.3. Maršruto ekranas**

Maršruto ekranas rodo sudarytą maršrutą žemėlapyje. Žymekliai yra sunumeruoti, nurodant aplankymo seką, o maršruto linija yra aiškiai pažymėta. Pėsčiųjų dalis yra pažymėta raudona linija, o viešojo transporto dalis – žalia linija. Vartotojas gali spustelėti bet kurį žymeklį, kad gautų daugiau informacijos apie konkretų maršruto tašką.

#### **5.4.4. Naudojimo patogumas**

Dizainas buvo sukurtas atsižvelgiant į vartotojo patogumą. Visi svarbūs mygtukai yra lengvai pasiekiami, o jų funkcijos yra aiškiai nurodytos. Naudojama navigacija yra intuityvi, leidžianti vartotojui greitai rasti reikiamą informaciją ir atlikti norimus veiksmus.

#### **5.4.5. Responsyvus dizainas**

Programėlės dizainas yra responsyvus, todėl jis gerai veikia įvairiuose įrenginiuose – tiek mobiliuosiuose telefonuose, tiek planšetėse. Visi elementai automatiškai prisitaiko prie ekrano dydžio, išlaikydami aiškumą ir funkcionalumą. Tai užtikrina, kad vartotojai turės puikią patirtį nepriklausomai nuo naudojamo įrenginio.



# Išvados ir rekomendacijos

## Išvados:

Kuriant mobiliąją programėlę, skirtą sudaryti maršrutus su įvairiais lankytiniais objektais ir viešojo transporto integracija, buvo pasiekti keli svarbūs tikslai:

- Sukurta funkcionali vartotojo sąsaja, leidžianti lengvai pasirinkti lankytinus objektus iš logiškai suskirstytų kategorijų.
- Integruotas "Google Maps API" [5], suteikiantis vartotojams galimybę matyti pasirinktus objektus žemėlapyje ir gauti išsamią informaciją apie kiekvieną objektą.
- Įgyvendinta maršrutų sudarymo funkcija, kuri naudoja "Directions API" [4], siekiant optimizuoti maršrutą ir pateikti jį vartotojui.
- Sukurta maršruto su autobusais funkcija, leidžianti vartotojui gauti informaciją apie viešojo transporto maršrutus ir lengvai sekti kelionę.
- Užtikrintas aiškus ir suprantamas maršruto vizualizavimas žemėlapyje, naudojant skirtingas spalvas pėsčiųjų ir viešojo transporto etapams.
- Sukurtas patrauklus ir funkcionalus dizainas, kuris yra patogus vartotojui ir prisitaiko prie įvairių įrenginių ekranų dydžių.

## Rekomendacijos:

Nors sukurta programėlė yra funkcionali ir naudinga, yra keletas sričių, kuriose galima toliau tobulinti ir plėsti jos galimybes:

- **Duomenų bazės integracija:** Nors šiuo metu vietų informacija yra saugoma programėlės kode, rekomenduojama integruoti duomenų bazę, kad būtų lengviau pridėti, redaguoti ir valdyti objektų duomenis.
- **Personalizuoti maršrutai:** Rekomenduojama pridėti personalizavimo funkciją, leidžiančią vartotojams išsaugoti dažnai naudojamus maršrutus ir juos lengvai pasiekti.
- **Realus laikas:** Integruoti realaus laiko duomenis apie transporto priemonių judėjimą ir spūstis, kad vartotojai galėtų geriau planuoti savo keliones.
- **Papildomos kalbos:** Kad programėlė būtų prieinama platesnei auditorijai, rekomenduojama pridėti daugiau kalbų pasirinkimų, siekiant tarptautinio naudojimo.
- **Vartotojo atsiliepimai:** Integruoti atsiliepimų sistemą, leidžiančią vartotojams teikti atsiliepimus ir pasiūlymus, kurie gali padėti toliau tobulinti programėlę.

Apibendrinant, sukurta programėlė jau yra funkcionali ir naudinga, tačiau toliau tobulinant ir pridėdant naujas funkcijas, ji gali tapti dar labiau pritaikyta vartotojų poreikiams ir dar efektyvesnė kasdieniame naudojime.

## Literatūros šaltiniai

- [1] Google Cloud. Google cloud console. <https://cloud.google.com/console>. Pažiūrėta: 2024-06-16.
- [2] Wikipedia Contributors. Nearest neighbor algorithm. [https://en.wikipedia.org/wiki/Nearest\\_neighbor\\_algorithm](https://en.wikipedia.org/wiki/Nearest_neighbor_algorithm). Accessed: 2024-06-16.
- [3] Android Developers. Android studio. <https://developer.android.com/studio>. Pažiūrėta: 2024-06-16.
- [4] Google Developers. Directions api. <https://developers.google.com/maps/documentation/directions/overview>. Pažiūrėta: 2024-06-16.
- [5] Google Developers. Google maps api. <https://developers.google.com/maps/documentation>. Pažiūrėta: 2024-06-16.
- [6] Google Developers. Google maps sdk for android. <https://developers.google.com/maps/documentation/android-sdk>. Pažiūrėta: 2024-06-16.
- [7] Git. Git: Distributed version control system. <https://git-scm.com/>. Pažiūrėta: 2024-06-16.
- [8] Google. Google. <https://www.google.com>. Pažiūrėta: 2024-06-16.
- [9] Google. Google Maps. <https://maps.google.com>. Pažiūrėta: 2024-06-16.
- [10] Ivan Hintoro, Sandy Kosasi, David, and Susanti Margaretha Kuway. Applying genetic algorithms for route optimization using google maps api. *International Journal of Advanced Research in Artificial Intelligence*, 2023. Pažiūrėta: 2024-06-16.
- [11] JetBrains. IntelliJ idea. <https://www.jetbrains.com/idea/>. Pažiūrėta: 2024-06-16.
- [12] Vilniaus miesto savivaldybė. Neakivaizdinis Vilnius. <https://neakivaizdinisvilnius.lt/apie/>. Pažiūrėta: 2024-06-16.
- [13] Ningning Wang, Xueqin Wei, Xixi Wang, Bo Zhang, Bo Zhang, and Feng Xu. Improving user experience with mobile app design: A case study. *Journal of Systems and Software*, 186:111209, 2022. Pažiūrėta: 2024-06-16.
- [14] Ming Xiao, Lihua Chen, Haoxiong Feng, Zhigao Peng, and Qiong Long. Daugiaobjektyvės optimizacijos metodikų taikymas viešojo transporto maršrutų planavimui. *Springer-Link*, 2024. Pažiūrėta: 2024-06-16.