VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Baigiamasis bakalaurinis darbas

Neregių navigavimas naudojant Android

(Blind people navigation with „Android“)

Atliko: Kompiuterių mokslo 4 kurso 2 grupės studentas
Mykolas Banevičius
Darbo vadovas: prof. dr. Rimantas Vaicekauskas
Recenzentas: partnerystės docentas Rokas Masiulis

Vilnius
2019

Turinys

Įvadas 4
Darbo tikslas 4
1. Kuriamų programėlių neįgaliesiems standartai 6
2. „Android“ operacinės sistemos standartinis funkcionalumas 9
2.1. „Android“ standartinis priemonės regos negalią turinčiam asmeniui 9
2.1.1. „Text – to – speech“ integruota galimybė 9
2.1.2. „Voice assistant“. Standartinis funkcionalumas 11
2.1.3. „Accessibility shortcut“ standartinė priemonė 12
2.1.4. „Font size“ integruota funkcija 12
2.1.5. „Magnifier window“ standartinės galimybės 13
2.1.6. „Magnification gestures“ integruotas funkcionalumas 13
2.1.7. „Colour adjustment“ standartinis funkcionalumas 14
3. „Android“ operacinės sistemos nestandartinis funkcionalumas 15
3.1. „Ariadne GPS“ programėlė 15
3.1.1. „Ariadne GPS“ teigiamos savybės 16
3.1.2. „Ariadne GPS“ neigiamos savybės 16
3.2. „GetThere“ programėlė 17
3.2.1. „GetThere“ programėlės teigiamos savybės 17
3.2.2. „GetThere“ programėlės neigiamos savybės 17
4. Programėlės „Blindify“, regos negalią turinčiam asmeniui, kūrimas 18
4.1. Programėlės tikslas 18
4.2. „Android“ standartinių ir išplėstinių galimybių panaudojimas 18
4.2.1. Programėlės prisijungimas 19
4.2.2. „Firebase“ išorinės duomenų bazės duomenų atnaujinimas 19
4.2.3. Veiksmai su „Google maps“ 20
5. Sukurtos programėlės „Blindify“ nefunkciniai reikalavimai 26
6. Kuriamos programėlės („Blindify“) funkciniai reikalavimai 27
6.1. Prisijungimas 27

6.2. Registracija. 27
6.3. Naudotojų sąrašas 27
6.4. Navigavimas iki pasirinkto programėlės naudotojo 28
6.5. Kasdienių objektų atpažinimas 28
7. Sukurtos programėlės „Blindify“ komponentų diagramos 29
7.1. Prisijungimo lango diagrama 29
7.2. Registracijos lango diagrama 29
7.3. Naudotojo prisijungimo prie sąrašo diagrama 30
7.4. Naudotojo atsijungimo nuo sąrašo diagrama 30
7.5. Prisijungusio naudotojo radimas sąraše 31
7.6. Objekto atpažinimo diagrama 31
7.7. Naudotojo kelionė link kelionės tikslo 32
7.8. Naudotojo lokacijos radimo diagrama 33
8. Programėlės testavimas 33
8.1. Atliktos apklausos rezultatai 33
8.2. Išmaniosios programėlės testavusių asmenų atsiliepimai 37
9. Išnagrinėtų bei sukurtos programėlės funkcionalumo palyginimas 39
10. Rekomendacijos 40
11. Rezultatai bei išvados 41
12. Summary 43
13. Literatūros sąrašas 44

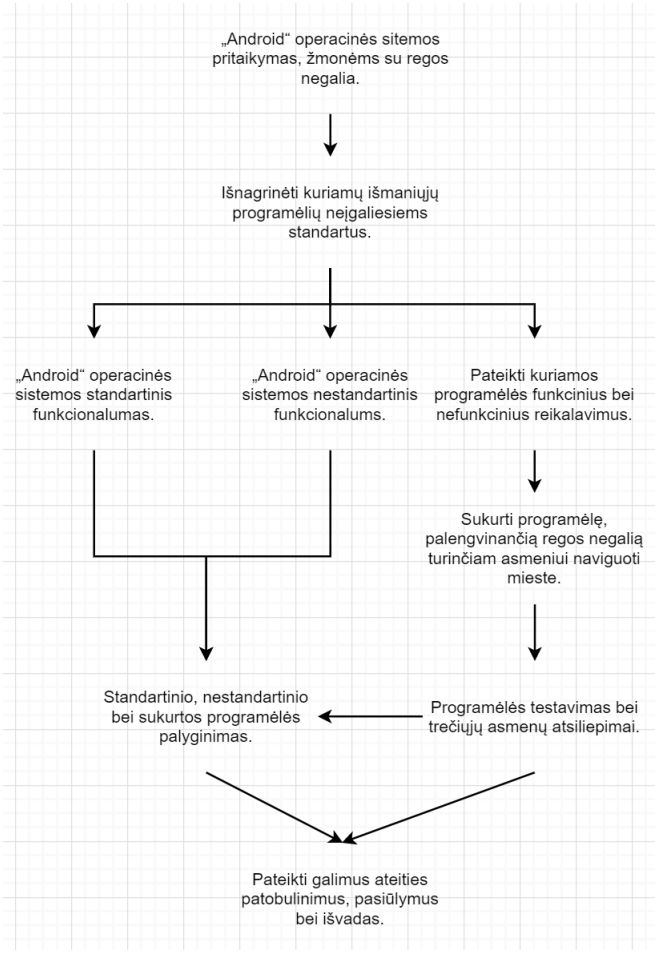
Įvadas

Pasaulyje yra sukurta milžiniškas kiekis išmaniųjų programėlių, pritaikytų mobiliesiems įrenginiams. Pagal 2019 metų „Statista“ duomenis, viena didžausių „Android“ operacinei sistemai pritaikytų programėlių parduotuvių „Google play“ savyje turi apie 2 100 000 išmaniųjų programėlių [I1]. Taip pat, yra prognozuojama, jog 2022–aisiais metais programėlių metinis atsisiuntimas pasieks net 258.2 milijardų [I2]. Šis rodiklis, bėgant metams, kils kartu su gyventojų rodiklio kilimu. Kad ir kaip bebūtų gaila, augant gyventojų populiacijai, yra neišvengiamas ir neįgalių asmenų skaičiaus didėjimas. Pagal „Disabled World“ 2018 metų duomenis net 650 milijonų žmonių visame pasaulyje yra neįgalūs bei apie 1,3 milijardo žmonių turi regos sutrikimų [I4]. Regos, klausos ir judėjimo negalią turintiems žmonėms reikia pagalbos ne tik kasdieniniuose buitiniuose reikaluose, tačiau ir kelionėse bei orientuojantis vietovėje.

Ne paslaptis, jog vienas didžiausių sunkumų regos negalią turinčiam asmeniui yra orientavimasis vietovėje (trumpiau – navigacija) ypač miestuuose. Aklo asmens laukia ne tik orientavimosi sunkumai, tačiau ir pavojai, nes greitų ir urbanizuotų miestų gyvenimo tempas yra dažniausiai per didelis sutrikimų turinčiam asmeniui. Taip pat, pastebėta, yra aktuali dviejų neregių asmenų susitikimo problema. Šio baigiamojo bakalaurinio darbo metu buvo atsižvelgta į regos negalią turinčio asmens sunkumus ir gilinamasi į jų navigacijos problemą. Iškeltas tikslas– sukurti „Android“ operacinei sistemai pritaikytą išmaniąją programėlę, kuri padėtų aklam naudotojui pasiekti savo draugus ar kitus programėlės naudotojus.

Darbo tikslas

Šio darbo tikslas yra išnagrinėti „Android“ operacinės sistemos teikiamą standartinį bei kitų programinės įrangos kūrėjų papildytą funkcionalumą regos negalią turinčiam asmeniui. Kritiškai įvertinti jų teigiamas bei neigiamas savybes bei jas palyginti tarpusavyje.Taip pat, išnagrinėti standartus, kurie apibrėžia, kaip turi atrodyti lengvai naudojama programėlė neįgaliam asmeniui. Trečia, suprojektuoti ir, atsižvelgus į jau egzistuojančių programėlių teigiamas bei neigiamas savybes bei išnagrinėtus standartus, sukurti „Android“ oparacinei sistemai pritaikytą išmaniąją programėlę, regos negalią turinčiam asmeniui. Aprašyti idėją, pateikti funkcinius bei nefunkcinius reikalavimus, ją duoti išbandyti trečioms šalims (asmenims) bei pateikti jų vertinimus, nuomonę bei galimus patobulinimus. Ketvirta, palyginti, kuo sukurta programėlė yra pranašesnė už egzistuojančias programas bei standartinį „Android“ operacinės sistemos funkcionalumą. Pateikti rekomendacijas bei išvadas apie šios operacinės sistemos pritaikymo galimybes regos sutrikimų turintiems naudotojams.



1 pav. Bakalaurinio darbo tikslo diagrama.

1. Kuriamų programėlių neįgaliesiems standartai

Sparčiai plintant išmaniųjų telefonų bei planšetinių kompiuterių rinkai bei naudojimui, pastebėta, jog išaugo poreikis jais naudotis ne tik pilnai funkcionuojančiam žmogui, bet ir tam tikrų neįgalumų turintiems asmenims. Šio amžiaus pirmajame dešimtmetyje buvo susidomėta, kaip visiems prieinamus Internetinius puslapius reiktų pritaikyti ir tam tikrą negalią, pavyzdžiui, regos, turinčiam asmeniui. Problema buvo iškelta ir analizuojama moksliniame tyrime [MFT05], kuriame autoriai apibrėžė Internetinio puslapio prieinamumo svarbą regos negalią turinčiam asmeniui ir taip pat nurodė standartus, kokiu procesu turėtų būti kuriami internetiniai puslapiai, norint nesusidurti su dizaino problemomis tolesnėse programavimo stadijose bei sukurti juos lengvai naudojamus ne tik visaverčiam žmogui, bet ir regos negalią turinčiam asmeniui. Panašūs standartai kaip Internetiniuose puslapiuose išlieka ir išmaniųjų telefonų bei planšetinių kompiuterių programėlių kūrėjams.

Išmaniųjų programėlių, regos negalę, tačiau ne visišką aklumą, turinčiam asmeniui, standartus plačiai aprašomo [Ego17] moksliniame darbe. Autorė pateikia pačias pagrindines bei butiniausias kuriamos programėlės savybes, kurios turi palengvinti pusaklio (ne visiškai aklo) asmens naudojimąsi ja.

Pirmoji savybė – lengvai skaitomas tekstas. Kadangi programėlės naudotojų didžioji dauguma turi regos sutrikimų, tad lengvai susiliejančių objektų ekrane vengimas yra būtina, nes kitaip naudotojas turės problemų su vaizdinio turinio suvokimu. Autorė teigia, jog iki 80 procentų visos programėlės informacijos pateikimas tekstu pagerintų programėlės suvokimą. Teksto dydis bei kontrastas turi būti atrinktas taip, jog jis būtų kiek įmanoma skaitomesnis vartotojui. Taip pat svarbu pasirinkti tinkamą teksto šriftą. Būtina atsižvelgti, jog vartotojas lengvai atskirtų raides. Pateiktas pavyzdys, kaip „Roboto“ ir „SanFrancisco“ šriftų raides mato tam tikrų regos sutrikimų turintys asmenys.

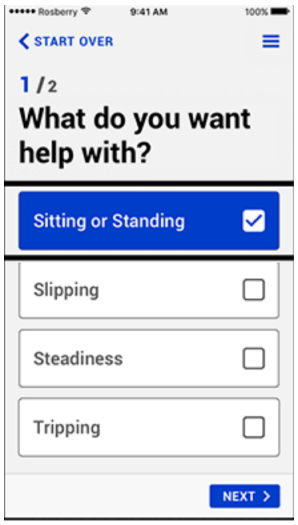


2 pav. Šrifto raidžių matymo palyginimas.

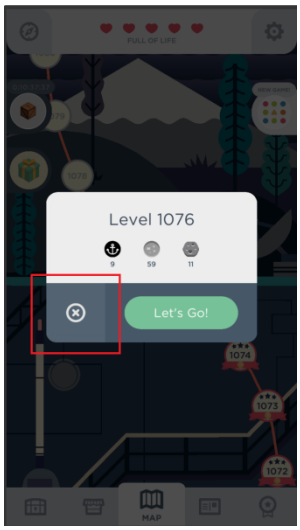
Spalvų neskiriančiam asmeniui labai svarbu, jog programėlės vartotojo sąsajos objektai turėtų didelį kontrastą su programėlės fonu. Kontrasto santykis neturėtų būti mažesnis nei 4.5:1. Jog būtų lengviau suprasti, kaip pavyzdys yra juodo paveikslėlio kontrastas baltame fone. Jo santykis yra 21:1 [I3]. Šis aukšto kontrasto pasirinkimas palengvintų ne tik daltonizmu sergančiam žmogui, tačiau ir labai silpną regą turinčiam asmeniui programėlės naudojimąsi. Silpnai matančiam asmeniui būtų suteikta galimybė atpažinti vartotojo sąsajos elementus bei suteikta galimybė greičiau ir lengviau susiorientuoti išmaniojoje programėlėje.

Pateiktas tekstas programėlėje turi būti trumpas bei lengvai suprantamas. Jei informacijos yra daug, patartina naudoti garsinius failus [Ego17] – naudotojas paspaudęs ant lengvai suprantamos piktogramos (paveikslėlio) vartotojo sąsajoje galėtų išgirsti norimą informaciją. Garso išvesties mygtukas turi būti prie norimos išgirsti informacijos temos. Jei nėra garsinio failo, patartina naudoti ekrano skaitymo funkcijas, kurios perteiktų parašytą tekstą ekrane, garsiniu signalu. Šios funkcijos leistu naudotis programėle ne tik pusakliam žmogui, bet ir visiškai regą praradusiam asmeniui.

Regėjimo sutrikimų turinčiam žmogui, viena iš didžiausių problemų yra informacijos radimas programėlėje bei navigavimas joje. Pirma, platus nuorodų naudojimas palengvintų greitą informacijos radimą. Antra, programėlės vartotojo sąsaja gali būti sudaryta iš nedaug mygtukų, jog būtų lengva juos įsiminti. Jų lengvai suprantamas išdėstymas ekrane sumažintų supainiojimo tikimybę. Trečia, „stacionarusis“ navigavimimo būdas gali būti puikus palengvinimas aklam asmeniui. Programėlė skenuoja esamo lango vartotojo sąsajos fragmentus/objektus bei pranešinėja, kuris objektas yra šiuo momentu pažymėtas. Vartotojas pasirenka pažymėtą objektą [Ego17]. (3 pav.). Taip pat, labai svarbu įgyvendinti lengvą bei aiškų programos modulio išjungimą pagal Ayesha Zafar [Zaf17]. Žr. Pav. Nr. 4. Šis funkcionalumas yra dažnai pamirštamas, tačiau yra labai svarbus sklandžiam, regos negalią turinčiam asmens, keliavimui po programėlę.



3 pav. „Stacionarusis“ navigavimo būdas.



4 pav. Programos modulio išjungimo pavyzdys.

Vienas iš esminių standartų moksliniame tyrime [ST12] aprašytų yra išlaikyti sukurtos programėlės skirtingus langus vienodos struktūros. Kaip pavyzdys būtų vienodas mygtukų su skirtingais funkcionalumais išdėstymas programos skirtinguose languose. Naudotojas įsimintų mygtukų lokacijas ekrane, ir tai leistų jam lengviau susiorientuoti programoje. Ši taisyklė galioja visiems programos vartotojo sąsajos fragmentams. Taip pat, autoriai teigia, jog reiktų vengti sudėtingesnių programos fragmentų kaip įrankių juostos, slankiklių bei naujai išsiskleidžiančių lange fragmentų. Patartina, jog kiekvienas programos langas būtų stacionarus.

Programėlės bendravimas su regos negalią turinčiu asmeniu paprastai yra garsiniu signalu arba telefono vibracija. Yra jau sukurtų išmaniųjų programėlių kaip „GetThere“, kurios savyje turi tam tikrą suprogramuotą funkcionalumą, kaip pavyzdžiui, kratymo judesio atpažinimą. Tokio tipo funkcionalumo įgyvendinimas programėlėje išplėstų jos ir naudotojo komunikacijos sugebėjimus, t.y. telefono kratymo atpažinimo funkcijai būtų galima pridėti dar papildomą funkcionalumą.

2. „Android“ operacinės sistemos standartinis funkcionalumas

Standartinis „Android“ operacinės sistemos funkcionalumas, regos negalią turintiems asmenims, yra šio gaminio kūrėjų, „Google“ korporacijos, sukurtas ir įdiegtas funkcionalumas į operacinę sistemą. Vartotojas tai gauna kartu su telefonu ir neturi galimybės to pašalinti. Jo paskirtis yra palengvinti negalių turinčiam asmeniui telefono naudojimąsi. Vartotojas gali pats atsirinkti, koks funkcionalumas jam yra priimtinas ir jį nusistatyti. Visą regos negaliai pritaikytą funkcionalumą „Android“ telefone galima rasti nukeliavus į programėlių meniu (Apps), paspaudus ant nustatymų (Settings) piktogramos bei paspaudus prieinamumo (Accessibility) pasirinkimą. Naujai atidarytame lange, vartotojas turi galimybę matyti tris neįgalumo kategorijas: regos (Vision), klausos (Hearing), vikrumo bei judėjimo (Dexterity and Interaction). Taip pat žemiau lange yra pateikta daugiau nustatymų, nepriskirtų konkrečiai negaliai, tačiau jie yra naudingi, bet kokią negalią turinčiam asmeniui. Šiame skyriuje bus išnagrinėtas ir apžvelgtas funkcionalumas, padedantis lengviau naudotis „Android“ telefonais, regos negalią arba sutrikimų turinčiam asmeniui. Tiriamos sistemos funkcionalumas nuo versijos – 6.0.1.

2.1. „Android“ standartinis priemonės regos negalią turinčiam asmeniui

Vizualinis sutrikimas, taip pat žinomas kaip regos sutrikimas arba visiškas regos praradimas yra sumažėjęs gebėjimas matyti, kuris sukelia problemų, kurios nėra išsprendžiamos įprastomis priemonėmis kaip, pavyzdžiui, akiniai ar kontaktiniai lęšiai. Tokiu atveju visiškai sutrinka tikrovės regimasis suvokimas. Šie vizualiniai sutrikimai paprastai sukelia žmogui sunkumų atlikti įprastus kasdieninius darbus kaip motorinės transporto priemonės vairavimas, skaitymas, navigavimas mieste ar socializavimas. Pagal „World Health Organization“ 2018 metų galo duomenimis apytiksliai 1.8 milijardo žmonių visame pasaulyje gyvena su viena iš regos sutrikimo formų [I7]. Tad „Android“ operacinės sistemos kūrėjai, matydami, šios sistemos pritaikymą regos negaliai turinčiam žmogui, poreikį, įdiegė funkcionalumą, kuris padeda kasdieniniame gyvenime naudotis šia sistema milijonams žmonių. Šio įdiegto funkcionalumo aprašymas bus pateiktas toliau.

2.1.1. „Text – to – speech“ integruota galimybė

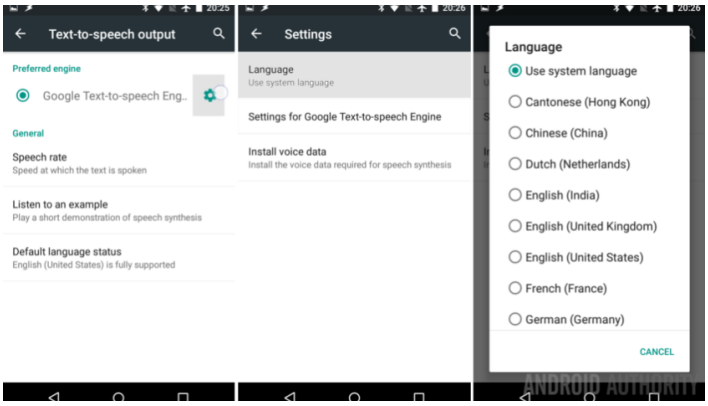
Ši priemonė labai gelbsti visiškai regos negalią turintiems asmenims. Jį galima rasti „Android“ prieinamumo (Accessibility) lange paspaudus „Text – to – speech“ pasirinkimą. Jis leidžia vartotojui pasirinkti norimą TTS (Text – to – speech) funkcionalumo variklį, bei įkelti išorinį. Visuose šios operacinės sistemos telefonuose, pagal nutylėjimą, yra įrašyta „Google“ korporacijos TTS variklis. Žemiau bus palyginti bei aptarti „Google“ bei “Samsung“ varikliai:

- „Samsung“ „Text – to - speech“ variklis. Jis pateikia galimybę naudotojui pasirinkti, kokia kalba išmanusis telefonas bendraus su naudotoju. „Samsung pateikia vartotojui galimybę pasirinkti iš septynių kalbų: prancūzų, vokiečių,

italų, rusų, ispanų, britų anglų bei amerikiečių anglų. Taip pat yra galimybė, jog šis funkcionalumas naudotų numatytą telefono sistemos kalbą. Nunavigavus į šio funkcionalumo nustatymus, naudotojas mato, jog galima keisti ne tik komunikavimo su vartotoju kalbą, bet ir balso nustatymus. Naudotojui yra suteikta galimybė pasirinkti telefoninio įrenginio balso lytį. Taip pat, yra pateikta galimybė atsisiųsti bei įdiegti naujus balso nustatymus/duomenis. Yra net 16 pasirinkimų, tarp kurių korėjiečių bei kinų kalbos.

- „Google“ „Text – to – speech“ variklis nedaug skiriasi nuo „Samsung“, tačiau jis turi daug daugiau norimų kalbų pasirinkimų. Jų - pačiame įrenginyje - yra įrašyta net 51. Galima rinktis ne tik tarp kalbų, bet ir jų tarmių kaip pavyzdžiui, brazilų portugalų arba portugalų tarmės. Taip pat, šio įrenginio kalbos variklio nustatymuose yra daugiau pasirinkimų nei „Samsung“. Šio funkcionalumo naudotojas gali, pirma, parsisiųsti daugiau nei 40 naujų balso duomenų/kalbų, antra, siuntimui gali pasirinkti, ar siųsti naudodamasis Wi – Fi – suteikta galimybė taupyti Internetinius duomenis, trečia, yra galimybė nusistatyti įrenginio kalbėjimo garsumą – tai leistu lengviau suprasti, ką pranešinėja telefonas, jei tuo pačiu metu įrenginys lestų kitą įrašytą signalą, pavyzdžiui, grotų muzika. Taip pat, šis variklis pateikia galimybę naudotojui pasirinkti kalbėjimo tono intonaciją iš trijų pasirinkimų: išraiškingo (Expressive), šiek tiek išraiškingo (Somewhat expressive) bei lygaus (Flat). Naudotojui yra suteikta galimybė įjungti skaičių apdorojimo kalboje funkcionalumą (Number processing), kuris pagerina laiko ir datos naudotojo supratimą ir leisti „Google“ rinkti konfidencialią informaciją, kaip dažnai „Text – to – speech“ funkcionalumas yra naudojamas, koks yra šio funkcionalumo užklausų našumas, kuri kalba yra naudojama dažniausiai ir kaip dažnai ši funkcija nesuveikia tinkamai.

Iš šių dviejų „Text – to speech“ variklių palyginimo galima spręsti jog „Google“ sukurtas funkcionalumas yra pranašesnis nei „Samsung“, tačiau jo nustatymų/pasirinkimų langas yra sudėtingesnis nei konkurento, dėl nustatymų gausos. Taip pat „Text – to – speech“ leidžia pasirinkti norimą kalbėjimo greitį. Vizualus šio funkcionalumo pavyzdys pateiktas 5 pav.



5 pav. „Text – to – speech“.

2.1.2. „Voice assistant“. Standartinis funkcionalumas

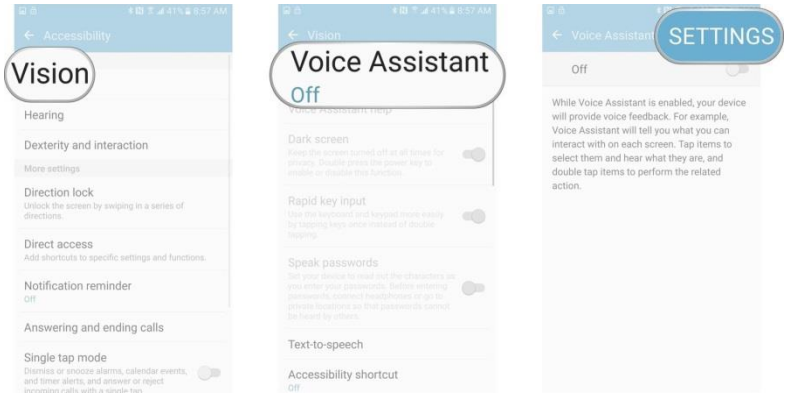
Šį standartinį funkcionalumą taip pat galima rasti „Accessibility“ lange paspaudus „Vision“ pasirinkimą. Šio funkcionalumo įjungimas yra gana paprastas, tiesiog reikia su pirštu paspausti ant šio pasirinkimo mygtuko „Vision“ lange. (6 pav.). Šis telefono savybė yra labai svarbi visišką regos negalią praradusiam asmeniui. Pasirinkus ją, telefonas pradeda garsiniu signalu pranešinėti naudotojui kiekvieną jo žingsnį telefone, kokiame jis yra lange, lango sudarymo fragmentus, jų skaičių bei įrenginio būklę, kaip pavyzdžiui, užgesus telefono ekranui, naudotojui bus pranešta, jog „Screen turned off“, kas lietuvių kalba reiškia, kad ekranas išsijungė. Taip pat, ši standartinė telefono funkcija praneša ant kokio naudotojo sąsajos fragmento naudotojo pirštas yra nuvestas bei perskaito visą to fragmento tekstą, taip vartotojas nusprendžia, ar nori pasirinkti esamą pažymėtą pasirinkimą, ar ne. Pasirinkimų skaičiui netilpus į vieną telefono langa, funkcija praneša naudotojui kiek pasirinkimų yra rodoma lange ir kiek langų dar galima pasiekti su kitais pasirinkimais. Šio funkcionalumo minusas yra, jog jis labai sulėtina naudotojo darbą telefonu bei vartotojui reikia laiko priprasti prie naujo telefono valdymo. Jis pasikeičia nuo įprastinio. Taip pat „Android“ pateikia šios funkcijos naudojimosi apmokymus apie:

- Ekrano tyrinėjimą.
- Navigavimą per langus.
- Navigavimą per vieną langą.
- Teksto keitimą.
- Kitas mažesnes funkcijas.

Šis funkcionalumas turi kelias funkcijas, kurios pagerina „Voice Assistant“ naudojimą:

- „Dark screen“ funkciją, kuri leidžia telefonu naudotis, jei telefono ekranas yra išsijungęs. Taip užtikrina aklo žmogaus privatumą, jog aplinkiniai žmonės nematytų, kas vyksta ekrane.

- „Rapid key input“ funkcija, kuri leidžia greičiau ir lengviau aklam asmeniui naudotis įrenginio klavietūra, leisdamas pasirinkti norimą simbolį ar raidę, paspaudus ant jo tik vieną kartą.
- „Speak passwords“ funkcija diktuoja telefono naudotojui jo vedamą slaptažodį, kad jis būtų įvestas be klaidų iš pirmo karto, taip išvengdami naudotojo profilio blokavimo. Naudotojas turi įsitikinti, jog aplink jokių pašalinių asmenų nėra arba naudoti telefonines ausines.



6 pav. „Voice Asssitant“ vizualus pavyzdys.

2.1.3. „Accessibility shortcut“ standartinė priemonė

Tai akliems žmonėms sukurtas funkcionalumas, kuris leidžia greitai telefone įjungti aukščiau aprašytą „Voice assistant“ funkciją bei nunaviguoti iki regos negalią turinčiam asmeniui „Android“ sukurto standartinio funkcionalumo lango. Įjungus šią funkciją jos naudojimas tampa gana paprastas, susideda iš dviejų vartotojo žingsnių:

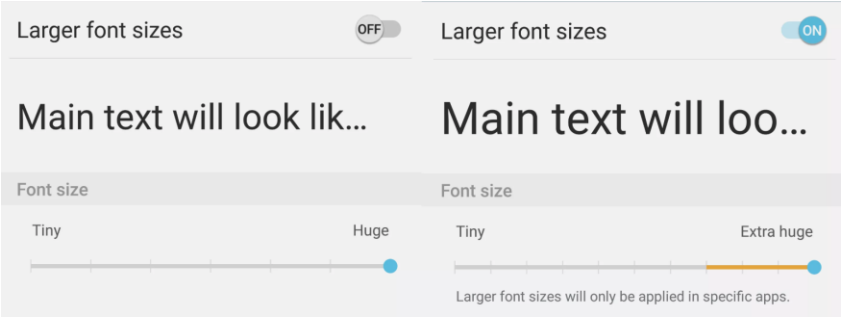
1. Naudotojas turi paspausti ir laikyti paspaudus įjungimo mygtuką, kol išgirs garsinį signalą arba pajaus telefono vibraciją.
2. Tuomet naudotojas turi su dviem pirštais paspausti ir laikyti telefono ekraną, kol išgirs garsinį patvirtinimą.

Įvykdžius šiuos du žingsnius, naudotojas bus nunaviguotas iki funkcionalumų, regos negalią turinčiam asmeniui, lango bei, kaip jau buvo minėta, bus įjungta „Voice asssitant“ funkcija.

2.1.4. „Font size“ integruota funkcija

Ši „Android“ operacinės sistemos standartinė funkcija yra reikšminga regos sutrikimų turinčiam asmeniui, tačiau ne visiškai ją praradusiam naudotojui. „Font size“ leidžia telefono naudotojui nesunkiai pasikeisti rašomo teksto ekrane dydį. Pagal „Android“ dokumentaciją, geriausias teksto dydžio matavimo vienetas yra SP (scale – independent pixels), tad visas

tekstas šioje operacinėje sistemoje yra matuojamas šiuo matavimo vienetu. Jei operacinėje sistemoje teksto dydis būtų matuojamas standartiniu vienetu – PX (pikseliais), tai jis atrodys didelis mažos rezoliucijos ekrane bei mažas didelės rezoliucijos ekrane. Matavimas SP vienetais šią problemą panaikina. Naudotojas gali pasirinkti iš septynių teksto dydžių, nuo pat mažiausio iki didelio. Taip pat, jei vartotojui nepakanka esamo teksto dydžio, jis gali įjungti „Larger font sizes“ funkciją, kuri leis dar keturiomis padalomis padidinti ekrane rodomo teksto dydį (7 pav.).



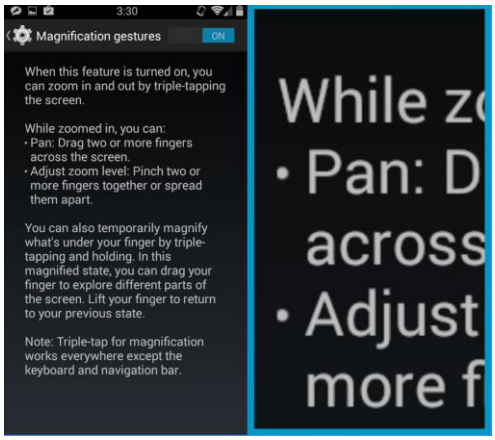
7 pav. „Font size“ vizualus pavyzdys.

2.1.5. „Magnifier window“ standartinės galimybės

„Magnifier window“ – „Android“ funkcija, leidžianti telefono naudotojui papildome lange ekrane matyti priartintą norimą vaizdą. Vartotojas turi galimybę papildomą langą, kuris atrodo kaip priartinimo stiklas, slankioti su pirštu per ekraną. Taip pat keisti lango dydį tarp trijų pasirinkimų: mažo (Small), vidutinio (Medium) bei didelio (Large). Vartotojas, taip pat, turi galimybę didinti priartinimo procentą.

2.1.6. „Magnification gestures“ integruotas funkcionalumas

„Magnification gestures“ – funkcionalumas, kuris leidžia naudotojui priartinti norimą ekrano elementą. Veikia su beveik visais naudotojo sąsajos ekrano fragmentais, išskyrus klaviatūrą. Vartotojui priartinus tam tikrą ekrano fragmentą, jis gali naudotis telefonu tame fragmente kaip ir prieš tai. Priartintas fragmentas yra rodomas visame ekrane. Priartinti vaizdą galima tris kartus paspaudus ant ekrano. Rodomą vaizdą priartinus slankioti po ekraną galim dviem pirštais braukiant. Taip pat, naudojant du pirštus, galima didinti arba mažinti priartinimo procentą (8 pav.).



8 pav. „Magnification gestures“ vizualus pavyzdys.

2.1.7. „Colour adjustment“ standartinis funkcionalumas

Tai funkcionalumas, leidžiantis daltonizmu sergantiems asmenims redaguoti ekrano spalvas, jei naudotojai pastebi, jog jiems yra sunku skaityti ekraną bei atskirti vartotojo sąsajos ekrano fragmentus – jie susilieja. Naudotojas pasirinkęs, jog nori koreguoti esamas ekrano spalvas, jis yra nunaviguojamas į naują langą, kuriame yra pateikiama 16 spalvų ir paprašoma jas sudėti eilės tvarka (9 pav.).

9 pav. „Colour adjustment“ vizualus pavyzdys.

Naudotojui sudėjus spalvas, funkcija nusprendžia, kokio tipo daltonikas jis yra ir pateikia spalvų spektrą su pavyzdžiu (nuotrauka), kuris palengvins jam naudojimąsi telefonu.

Taip pat „Android“ operacinė sistema daltonizmu sergantiems žmonėms pateikia funkcionalumą, kuris padaro visus telefono komponentus bespalviais – visi komponentai tampa skirtingu pilkos spalvos atspalviu. Šio funkcionalumo pavadinimas yra „Greyscale“. Taip pat, „Android“ pateikia neigiamų spalvų funkciją, kuri visas spalvų spektrą apverčia ir taip ekrano skaitomumas, daltonizmu sergantiems asmenims, tampa lenvesnis. Funkcijos pavadinimas - „Negative colours“.

3. „Android“ operacinės sistemos nestandartinis funkcionalumas

Šiame skyriuje bus nagrinėjama, kokios jau yra sukurtos, ne „Android“ operacinės sistemos, kūrėjų programėlės, lengvinančios aklo arba silpnaregio žmogaus gyvenimą kasdieninėje aplinkoje. Tyrinėjant, kokios jau yra sukurtos programėlės, buvo pastebėta jų gausa, dėl to teko atsirinkti keletą programėlių pagal kriterijus:

- Programėlės funkcionalumas turi padėti žmogui susigaudyti esamoje aplinkoje.
- Programėlė turi padėti žmogui nukakti iš taško A į tašką B.

Buvo pasirinkti šie du kriterijai, jog atrinktas programėles būtų galima palyginti su šio baigiamojo bakalaurinio darbo kuriamu projektu.

3.1. „Ariadne GPS“ programėlė

Ši programėlė yra informatyvinio pobūdžio. „Ariadne GPS“ suteikia galimybę susipažinti su jo dabartine lokacija ir gauti informaciją apie esamos gatvės pavadinimą, artimiausio namo numerį ir pan. Ši programėlė leidžia neregiui, interaktyvia lietimu bendrauti su ekrane pažymėtu žemėlapiu, kuriame yra pavaizduotas dabartinės lokacijos vietos žemėlapis. Programėlės naudotojas braukdamas pirštu per miesto planą, garsiniu signalu gauna informaciją apie kiekvieną gatvę bei namo numerius, kuriuos pažymėjo pirštu. Taip pat, naudotojas gali žodžiu pranešti programėlei miestą bei kitos gatvės pavadinimą, kurios regioną nori ištirti. Programėlės naudotojui norint sužinoti savo lokacijos vietą – užtenka paspausti vieną mygtuką. Taip pat, galima aktyvuoti funkciją, kuri periodiškai pranešinėja naudotojui jo lokacijos vietą. Vienas didžiausių pliusų, jog ji seka naudotoją ir kai jis keliauja viešuoju transportu. Turi funkcionalumą, kuris pranešinėja kertamas viešojo transporto stoteles.

Navigavimas per šią programėlę yra gana paprastas. Naudotojui programėlė pranešinėja kiekvieną vartotojo žingsnį garsiniais signalais bei vibracija. Žmogui norint atlikti tam tikrus veiksmus programėlėje, ji pranešinėja jam kiekvieną jo žingsnį joje. Tuomet asmuo piršto slinkimu į kairę arba į dešinę naudotojas naviguoja per pasirinkimus ir gaudamas informaciją apie kiekvieną iš jų, gali išsirinkti. Dvigubu greitu paspaudimu per ekraną naudotojas pasirenka norimą programos fragmentą.

Programėlės pagrindinio funkcionalumo sąrašas:

- Praneša dabartinę naudoto lokacijos vietą žemėlapyje.
- Leidžia naudotojui tyrinėti aplinką aplink jį.
- Leidžia tyrinėti pasirinktą specifinę zoną.

- Periodiškai skanuoja dabartinę lokacijos vietą ir jeigu ji pasikeitė nuo praeito skenavimo – praneša naudotojui jo naujos lokacijos informaciją.
- Leidžia naudotojui pridėti mėgstamiausių vietų sąrašą.
- Praneša vartotojui, kai jis priartėja prie mėgstamos vietos.



10 pav. „Ariadne GPS“ vizualus pavyzdys.

Ištestavus programėlę buvo pastebėtos kelios teigiamos bei neigiamos savybės.

3.1.1. „Ariadne GPS“ teigiamos savybės

1. Galimybė naviguoti programėle keliais būdais: navigavimas pirštu braukiant į kairę arba dešinę ir navigavimas įsimenant ir spaudžiant ant norimo mygtuko.
2. Programėlė priima naudotojo garsinius signalus.
3. Garsiniu signalu, aklam naudotojui pranešinėja informaciją.

3.1.2. „Ariadne GPS“ neigiamos savybės

1. Programėlės mygtukų skaičius yra didelis – sunku įsiminti visus mygtukus.
2. Programėlės mygtukai yra neproporcingai maži – regos negalią turinčiam asmeniui sunku jais naudotis.
3. Programėlėje yra funkcionalumo, kuris dažniausiai nėra aktualus ją naudojančiam asmeniui. Pavyzdžiui, programėlė taip pat pateikia naudotojui tikslias jo lokacijos koordinates. Paprastam asmeniui koordinatės, pateiktos skaitine verte, nebūtų prasmingos.

3.2. „GetThere“ programėlė

„GetThere“ programėlė yra sukurta akliems bei regos sutrikimų turintiems žmonėms. Ši programėlė neturi žemėlapio, kaip daugelis, tačiau ji nuolatos GPS padedama seka jos naudotoją bei gali jam pranešti jo lokacijos vietą. Ši programėlė skiriasi nuo kitų navigavimo programėlių akliems žmonėms tuo, jog ji sąveikauja su naudotoju ne tik vibracijos bei garsinių signalų pagalba, tačiau ir reaguoja, kai naudotojas pakrato telefoną. Ši programėlė apskaičiuoja kelią link tikslo ir pranešinėja naudotojui, kai jis nuklysta nuo jo. Taip pat vienas iš naudingesnių funkcionalumų yra, jog naudotojui pasirinkus kelionės tikslą, programėlė praneš naudotojui, kai jis priartės prie jo. Tai leidžia, taip pat, naudotojui naudotis viešuoju transportu, nes šis funkcionalumas leis jam sužinoti išlipimo stotelę.



11 pav. „GetThere“ vizualus programėlės pavyzdys.

Išbandant programėlę buvo pastebėtos jos teigiamos bei neigiamos savybės.

3.2.1. „GetThere“ programėlės teigiamos savybės

1. Programėlės funkcionalumas nėra perteklinis.
2. Programėlės mygtukų skaičius nėra didelis, juos lengva įsiminti. Jie išdėstyti ekrane taip, kad būtų lengva rasti ir jų dydis atititinka programėlių, skirtų neįgaliesiems, standartus.
3. Programėlės funkcionalumą galima aktyvuosi pakračius telefoną.
4. Ja lengva išmokti naudotis ir tai neužtruks daug laiko.

3.2.2. „GetThere“ programėlės neigiamos savybės

1. Programėlėje naudotojui reikia pasirinkti, kuriame pasaulio regione jis šiuo metu yra.
2. Programėlė turi skaitymo iš failo ir rašymo į failą funkcionalumą, kas autoriaus manymų yra bevertis dalykas.

4. Programėlės „Blindify“, regos negalią turinčiam asmeniui, kūrimas

Remiantis išnagrinėtais standartais bei išnagrinėjus keletą programėlių, padedančių regos negalią turintiems asmenims, naviguoti mieste buvo stengtasi sukurti lengvai naudojamą programėlę, kuri palengvintų neįgalaus asmens navigavimą link kito asmens/draugo. Projekte buvo stengtasi perteikti geriausias išnagrinėtų programėlių savybes bei sukurti tokią programėlę, kuri atitiktų neįgaliųjų programėlių standartus.

Programėlės instaliacinį failą galima rasti https://mif.vu.lt/~myba3045/blindify.apk.

4.1. Programėlės tikslas

Programėlės tikslas yra palengvinti aklo arba regos sutrikimų turinčio asmens navigavimą mieste iki pasirinkto asmens/draugo. Prireikus neįgalusis galės pasinaudoti programėle, kuri pateiks jam greičiausią kelią link norimo asmens - kito programėlės naudotojo. Šios programėlės tikslas, jog regos negalią turintis asmuo galėtų be aplinkinių pagalbos naviguoti mieste iki pasirinkto kelionės taško.

4.2. „Android“ standartinių ir išplėstinių galimybių panaudojimas

Visų pirma, programėlė darbui su lokacija naudoja „Google Maps“ įskiepį. Aplinkinių objektų atpažinimui buvo pasinaudota atviro kodo „Android Tensorflow Machine Learning Example“ [I5] programėle/funkcionalumu. Jis buvo integruotas į kuriamą programėlę bei buvo patobulintas, jog atpažintų ne tik objektą, bet ir jo spalvą. Buvo suprogramuotas funkcionalumas, jog atpažinimo rezultatus praneštų programėlės naudotojui garsiniu signalu. Taip pat, prisijungimui į išmaniąją programėlę buvo naudojamas „Firebase“ [I6] šablonas. Prisijungusių naudotojų būseną bei informacija yra laikoma išorinėje „Firebase“ duomenų bazėje. Greičiausio kelio radimui buvo integruotas funkcionalumas, kuris kreipiasi į „Google“ korporacijos sukurtą „Directions API“ [I7].

Programėlė buvo programuota su „Java“ programavimo kalba. Naudojama 8 „Java“ versija.

Žemiau bus pateiktas programėlės pagrindinio funkcionalumo paaiškinimai su kodo pavyzdžiais. Nebus keliamos šalutinės funkcijos, tokios kaip GPS/Interneto leidimų tikrinimas ir pan.

4.2.1. Programėlės prisijungimas

Prisijungimui buvo pasirinktas „Firebase“ šablonas, kuris tik suteikia galimybę naudotojui prisijungti bei prisiregistruoti prie sukurtos programėlės. Autorius naudoja šabloną, kuris gražina rezultatą, ar vartotojas prisijungė, ar ne (20 pav.). Programėlės naudotojui yra pranešama, ar jam pavyko prisijungti. Garsiniu signalui/pranešimui yra naudojama „Text – to – speech“ klasė (21 pav.).

```
    btnLogin.setOnClickListener((v) → {
            startActivityForResult(
                    AuthUI.getInstance().createSignInIntentBuilder()
                            .build(),LOGIN_PERMISSION
            );
    });
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if(requestCode == LOGIN_PERMISSION) {
        startNewActivity(resultCode, data);
    }
}

private void startNewActivity(int resultCode, Intent data) {
    if (resultCode == RESULT_OK) {
        Intent intent = new Intent( packageContext: Login.this, ListOnline.class);
        textToSpeechMethod( speechh: "Prisijungti pavyko.");
        startActivity(intent);
        finish();
    } else {
        textToSpeechMethod( speechh: "Prisijungti nepavyko.");
        Toast.makeText( context: this,  text: "Prisijungti nepavyko.", Toast.LENGTH_SHORT).show();
    }
}
```

20 pav. Naudotojo prisijungimo funkcionalumas.

```
private void textToSpeechMethod(final String speechh) {
    textToSpeech = new TextToSpeech(getApplicationContext(), (status) → {
        if (status == TextToSpeech.SUCCESS) {
            int ttsLang = textToSpeech.setLanguage(Locale.US);

            if (ttsLang == TextToSpeech.LANG_MISSING_DATA
                    || ttsLang == TextToSpeech.LANG_NOT_SUPPORTED) {
                Log.e( tag: "TTS",  msg: "The Language is not supported!");
            } else {
                Log.i( tag: "TTS", speechh);

                String data = speechh;
                Log.i( tag: "TTS",  msg: "button clicked: " + data);
                int speechStatus = textToSpeech.speak(data, TextToSpeech.QUEUE_FLUSH,  params: null);

                if (speechStatus == TextToSpeech.ERROR) {
                    Log.e( tag: "TTS",  msg: "Error in converting Text to Speech!");
                }

            }
            Log.i( tag: "TTS",  msg: "Initialization success.");
        } else {
            Toast.makeText(getApplicationContext(),  text: "TTS Initialization failed!", Toast.LENGTH_SHORT).show();
        }
    });
}
```

21 pav. „Text – to – speech“ funkcija.

4.2.2. „Firebase“ išorinės duomenų bazės duomenų atnaujinimas

Žemiau pateiktame pavyzdyje (23 pav.) yra pavaizduota, kaip dabartinė naudotojo lokacijos vieta yra išsaugoma “Firebase“ išorinėje duomenų bazėje. Panašiu principu yra saugomi ir prisijungusių naudotojų sąrašai.

```
locations = FirebaseDatabase.getInstance().getReference( path: "Locations");
```

22 pav. Nuoroda į „Firebase“ duomenų bazę.

```
mLastLocation = LocationServices.FusedLocationApi.getLastLocation(mGoogleApiClient);
if(mLastLocation != null) {

    //Update to firebase
    locations.child(FirebaseAuth.getInstance().getCurrentUser().getUid())
        .setValue(new Tracking(FirebaseAuth.getInstance().getCurrentUser().getEmail(),
                FirebaseAuth.getInstance().getCurrentUser().getUid(),
                String.valueOf(mLastLocation.getLatitude()),
                String.valueOf(mLastLocation.getLongitude())));
}
else {
    textToSpeechMethod( speechh: "Couldn't load location.");
    Log.d( tag: "TEST",  msg: "Couldn't load location");
}
```

23 pav. Sinčiami duomenys į „Firebase“ duomenų bazę.

4.2.3. Veiksmai su „Google maps“

4.2.3.1. Naudotojo sekimas žemėlapyje

Naudotojo sekimui žemėlapyje būtina patikrinti ar jis yra davęs programėlei sutikimą gauti jo lokacijos vietą. „DisplayCurrentLocation()“ metode aprašomas funkcionalumas, kuris seka naudotoją žemėlapyje.

```
mMap.getUiSettings().setZoomControlsEnabled(true);
if (ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION) !=
        PackageManager.PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_COARSE_LOCATION) !=
                PackageManager.PERMISSION_GRANTED) {
    ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION}, LOCATION_REQUEST);
    return;
}

mMap.setMyLocationEnabled(true);
mMap.getUiSettings().setMyLocationButtonEnabled(false);
mMap.setOnMyLocationChangeListener((arg0) → {

        mLocation = arg0;

        try {
            displayCurrentLocation();
        } catch (IOException e) {
            e.printStackTrace();
        }
```

24 pav. Naudotojo sekimas žemėlapyje.

```
public  void displayCurrentLocation() throws IOException {
    CameraPosition cameraPosition = new CameraPosition.Builder().
            target(new LatLng(mLocation.getLatitude(), mLocation.getLongitude())).
            tilt(60).
            zoom(19).
            bearing(0).
            build();
    mMap.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPosition));
}
```

25 pav. „DisplayCurrentLocation“ metodas.

4.2.3.2. Kelionės tikslo koordinačių gavimas bei pažymėjimas žemėlapyje

Žemiau pateiktame programėlės kodo pavyzdyje yra gaunamos kelionės tikslo koordinatės, kiekvieną kartą kai jos pasikeitė, iš „Firebase“ duomenų bazės. Žemėlapyje yra uždedamas žymeklis (26 pav.). Koordinatės yra gaunamos iš „Firebase“ duomenų laikmenos. Koordinačių identifikavimas yra elektroninis paštas. Duomenų bazėje, kiekvieno naudotojo koordinatės yra susietos su jo elektroniniu paštu. Kitaip tariant, gaunamo kito aklo/sveiko asmens koordinatės bei jos randamos žemėlapyje.

```
Query user_location = locations.orderByChild("email").equalTo(email);
user_location.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        for (DataSnapshot postSnapShot : dataSnapshot.getChildren()) {
            dataChanged = true;
            Tracking tracking = postSnapShot.getValue(Tracking.class);
            //Add marker for friend location

            LatLng friendLocation = new LatLng(Double.parseDouble(tracking.getLat()), Double.parseDouble(tracking.getLng()));
            friendsLocation = friendLocation;

            Location currentUser = new Location( provider: "");
            currentUser.setLatitude(lat);
            currentUser.setLongitude(lng);

            //create location from friend coordinate;
            Location friend = friendsLocationBuilder(tracking.getLat(), tracking.getLng());
            fLocation = friend;
            //clear old markers
            mMap.clear();

            //Add friend marker on map
            mMap.addMarker(new MarkerOptions().position(friendLocation)
                    .title(tracking.getEmail())
                    .snippet("Distance " + new DecimalFormat( pattern: "#.#").format( number: (currentUser.distanceTo(friend)) / 1000))
                    .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_BLUE)));
```

26 pav. Kelionės tikslo koordinačių gavimas bei pažymėjimas žemėlapyje.

4.2.3.3. Funkcionalumas naudotojui nenuklysti nuo kelionės tikslo

Žemiau pavaizduotas kodas, kuris reaguoja į kiekvieną telefono pasukimą ir pradeda vibruoti, jei telefonas nusisuka nuo tikslo per 10 laipsnių į kairę arba dešinę pusę (27 pav.). Buvo pasirinka 10 laipsnių paklaida į kiekvieną pusę, nes programėlės naudotojas neišlaikys

visada telefono laikydamas tiesiai tikslo link. Autorius mano, jog šis funkcionalumas tikrai palengvintų tikslo radimo krypties žinojimui.

Šis funkcionalumas, taip pat, buvo panaudotas ir kai naudotojas pasiekia vieną iš kelionės taškų. Tuo metu programėlė įjungia kompaso fragmentą bei praneša, jog naudotojui reikia suktis aplink savo ašį, kol programėlė nepraneš jam, kad naudotojas atsisuko į kitą kelio tašką.

```
@Override
public void onSensorChanged(SensorEvent event) {
    if (wayDirectionFound) {
        float degree = Math.round(event.values[0]);
        if (degree <= way + 10f && degree >= way - 10f ) {
            try {
            } catch (Exception e) {
                e.printStackTrace();
            }
        } else {
            if (Build.VERSION.SDK_INT >= 26) {
                vibrator.vibrate(VibrationEffect.createOneShot( milliseconds: 10, VibrationEffect.DEFAULT_AMPLITUDE));
            } else {
                vibrator.vibrate( milliseconds: 10);
            }
        }
    }
}
```

27 pav. Funkcionalumas naudotojui nenuklysti nuo tikslo.

4.2.3.4. Sąsaja su „Google“ „Directions API“

Jog programėlė gautų teisinga atsakymą iš „Directions API“ reikėjo sukurti bei sukonfiguruoti projektą „Google Developer Console“ [I8]. Konsolėje sugeneruoti API (Application programming interface) raktą bei jį naudoti sukurtoje programėlėje, jos identifikavimui. Šį raktą įdėjome į sukonstruoto adreso parametrus.

Programėlėje buvo įgyvendintas funkcionalumas, kuris sukonstruoja tinkamą Internetinį adresą, kuriuo galima pasiekti „Directions API“.

Sukonstruoto adreso pavyzdys:

https://maps.googleapis.com/maps/api/directions/json?origin=locationFrom.latitude,locationFrom.longitude&destination=locationTo.latitude,locationTo.longitude&key=API key&mode=walking.

- https://maps.googleapis.com/maps/api/directions - „Directions API“ Internetinis adresas.
- /json – rodiklis, jog gražinamo atsakymo kūno formatas bus „Json“ formato [I9] (27 pav.).

- origin=locationFrom.latitude,locationFrom.longitude – paduodami du koordinačių parametrai nuo kurių bus sakičiuojamas greičiausias kelias žemėlapyje.
- destination=locationTo.latitude,locationTo.longitude – paduodami du koordinačių parametrai iki kurių bus skaičiuojamas greičiausias kelias žemėlapyje.
- key=API key – parametras, pagal kurį „Directions API“ identifikuos programėlę, iš kurios yra kreipiamasi.
- mode=walking – parametras, pagal kurį nustato, kokio tipo greičiausio kelio programėlė tikisi. Tipai: važiavimo transporto priemone (driving), ėjimo (walking), važiavimas dviračiu (bicycling), keliavimas viešuoju transportu (transit).

Ši programėlė pateikia aklam naudotojui greičiausią kelią pėsčiomis, dėl to parametras „mode“ buvo pasirinktas „walking“.

```
{
  "Id": 0,
  "FirstName": "string",
  "LastName": "string",
  "Name": "string",
  "EmailAddress": "string",
  "TerritoryId": 0
}
```

27 pav. „Json“ formato pavyzdys.

```
StringBuilder stringBuilder = new StringBuilder();
Object[] dataTransfer = new Object[4];
GetDirectionsData getDirectionsData = new GetDirectionsData(getApplicationContext());

stringBuilder.append("https://maps.googleapis.com/maps/api/directions/json?");
stringBuilder.append("origin=" + mLocation.getLatitude() + "," + mLocation.getLongitude());
stringBuilder.append("&destination=" + fLocation.getLatitude() + "," + fLocation.getLongitude());
stringBuilder.append("&key=" + "AIzaSyA7MYUzcejMd4BrzJC1YoWvqifYDtvdliU");
stringBuilder.append("&mode=" + "walking");
```

28 pav. Internetinio adreso konstravimas.

4.2.3.5. Reikiamų duomenų gavimas bei iškirpimas iš „Directions API“ atsakymo

„Directions API“ atsakymo formatas yra vientisas simbolių masyvas, kuriame, „Json“ formatu, yra laikoma informacija. Dėl to reikėjo įgyvendinti funkcionalumą, kuris iš gražinto masyvo iškirptų reikiamą informaciją ir išsaugotų programėlėje.

„Directions API“ gražinamo simbolių masyvo reikiamos informacijos pavyzdys (29 pav.).


```
"routes": [ {
  "summary": "I-40 W",
  "legs": [ {
    "steps": [ {
      "travel_mode": "DRIVING",
      "start_location": {
        "lat": 41.8507300,
        "lng": -87.6512600
      },
      "end_location": {
        "lat": 41.8525800,
        "lng": -87.6514100
      },
      "polyline": {
        "points": "a~l~Fjk~uOwHJy@P"
      },
      "duration": {
        "value": 19,
        "text": "1 min"
      },
      "html_instructions": "Head \u003cb\u003enorth\u003c/b\u003e on \u003cb\u003eS
      Morgan St\u003c/b\u003e toward \u003cb\u003eW Cermak Rd\u003c/b\u003e",
      "distance": {
        "value": 207,
        "text": "0.1 mi"
      }
    },
```


29 pav. „Directions API“ atsakymo pavyzdys.

Programėlėje naudojama informacija iš gražinamo atsakymo: linija (polyline), trukmė (duration), tikslo lokacija (end_location). Šios informacijos kiekis priklauso nuo viso kelio, kelio taškų skaičiaus.

Išsiųsti užklausai buvo pasinaudota „HttpUrlConnection“ klasė.

Žemiau pateiktas pavyzdys, kaip informacija yra iškerpama iš „Directions API“ atsakymo.

```
JSONObject jsonObject = new JSONObject(s);
jsonArray = jsonObject.getJSONArray( name: "routes").getJSONObject( index: 0)
        .getJSONArray( name: "legs").getJSONObject( index: 0).getJSONArray( name: "steps");


arrayLenght = jsonArray.length();
String[] pollyArray = new String[arrayLenght];
JSONObject jsonObject2;

for (int i = 0; i < arrayLenght; i++) {
    jsonObject2 = jsonArray.getJSONObject(i);
    String polygone = jsonObject2.getJSONObject("polyline").getString( name: "points");
    pollyArray[i] = polygone;
}
```

30 pav. „Polyline“ informacijos iškirpimas iš atsakymo.

Panašiu principu buvo gaunama ir kita informacija.

4.2.3.6. Tikslios lokacijos vietos adreso gavimas/ visų gatvių iki kelionės tikslo radimas

Tikslios naudotojo lokacijos vietos adresui gauti yra naudojama „Geocoder“ klasė. Jai užtenka paduoti lokaciją – platumą bei ilgumą – ir ši klasė gražina paduotos lokacijos tikslų adresą, kurį sudaro: miestas, gatvės pavadinimas, šalis, šalies kodas, pašto kodas, namo numeris (31 pav.). Ši integruota galimybė į programėlę padėtų asmeniui, kuris orientuojasi mieste, lengviau suprasti, kurioje miesto vietoje jis yra.

```
private List<Address> getAddress (LatLng coordinates) throws IOException {
    Geocoder geocoder = new Geocoder( context: this, Locale.getDefault());
    return geocoder.getFromLocation(coordinates.latitude, coordinates.longitude,  maxResults: 1);
}
```

31 pav. Tikslaus adreso radimo funkcionalumas.

Visų gatvių pavadinimų iki kelionės tikslo radimui taip pat buvo naudota „Geocoder“ klasė. Šio funkcionalumo esmė, žmogui, kuris yra susipažinęs su miestu leisti lengviau įsivaizduoti kelionės maršrutą, pateikiant jam visų kertamų gatvių pavadinimus. Pasikartojančios gatvės nebus antrą kartą kartojamos.

```
private List<String> fetchStreets (LatLng current, LatLng goal) throws IOException {
    List <String> streets = new ArrayList<~>();
    Double currentLat = Double.parseDouble(new DecimalFormat( pattern: "##.###").format(current.latitude));
    Double currentLon = Double.parseDouble(new DecimalFormat( pattern: "##.###").format(current.longitude));

    double goalLat = Double.parseDouble(new DecimalFormat( pattern: "##.###").format(goal.latitude));
    double goalLon = Double.parseDouble(new DecimalFormat( pattern: "##.###").format(goal.longitude));
    while (currentLat != goalLat && currentLon != goalLon) {
        if (goalLat > currentLat) {
            goalLat = goalLat - 0.001;
            goalLat = Double.parseDouble(new DecimalFormat( pattern: "##.###").format(goalLat));
        }
        if (goalLon > currentLon) {
            goalLon = goalLon - 0.001;
            goalLon = Double.parseDouble(new DecimalFormat( pattern: "##.###").format(goalLon));
        }
        if (goalLat < currentLat) {
            goalLat = goalLat  + 0.001;
            goalLat = Double.parseDouble(new DecimalFormat( pattern: "##.###").format(goalLat));
        }
        if (goalLon < currentLon) {
            goalLon = goalLon + 0.001;
            goalLon = Double.parseDouble(new DecimalFormat( pattern: "##.###").format(goalLon));
        }
        LatLng streetCoordinates = new LatLng(goalLat, goalLon);
        List<Address> address = getAddress(streetCoordinates);
        streets.add(address.get(0).getThoroughfare());
    }
    return removeDuplicates((ArrayList<String>) streets);
}
```

32 pav. Funkcionalumas visų kertamų gatvių radimui.

4.2.3.7. Naudotojo sekimo funkcionalumas

Šis funkcionalumas įsijungs, kai naudotojas bus nukreipiamas į programėlės žemėlapio fragmentą. Bus vykdomi du dalykai:

- Programėlės naudotojui bus kiekvieną kartą pranešama, kada jis pradės artėti link kelionės tikslo/kelio taško.
- Programėlė kas ketvirtį atstumo nuo likusio atstumo iki kelionės tikslo/kelio taško pranešinės naudotojui, koks atstumas liko nueiti

Šis funkcionalumas baigia savo darbą, kai naudotojas priartėja prie kelionės tikslo/kelio taško per 5 metrus.

4.2.3.8. Aplinkinių objektų atpažinimo funkcionalumas

Programėlėje buvo įdėtas patobulintas atviro kodo „TensorFlow Machine Learning example“ programėlės funkcionalumas, kuris suteikia galimybę jam atpažinti daugiau nei 1000 kasdieninėje aplinkoje sutinkamų daiktų bei objektu, pradedant: namai, mašinomis, kelio ženklais, baigiant šaukšteliais bei panašiais dalykais. Šis funkcionalumas buvo patobulintas iki tiek, jog garsiniu signalu galėtų programėlės naudotojui pateikti kokį objektą jis nufotografavo ir kokia yra to objekto spalva. Atpažinamų spalvų spektrą sudaro iš pagrindinių 12 – os spalvų bei atspalvių.

5. Sukurtos programėlės „Blindify“ nefunkciniai reikalavimai

Kad programėlės naudotojas galėtų naudotis šia sukurta programėle, įrenginys turi atitikti šiuos reikalavimus:

- Mobilusis įrenginys turi palaikyti „Android“ operacinę sistemą.
- „Android“ operacinės sistemos versija neturi būti žemesnė nei 6.0.0.
- Mobilusis įrenginys turi leisti programėlei turėti prieigą prie įrenginio kameros.
- Mobilusis įrenginys turi leisti išmaniajai programėlei prieiti prie Internetinių duomenų.
- Mobilusis įrenginys turi leisti išmaniajai programėlei pasiekti GPS nustatymus.

6. Kuriamos programėlės („Blindify“) funkciniai reikalavimai

Susipažinus su neregiems kuriamų programėlių standartais bei išnagrinėjus egzistuojančiu programėlių galimybesbuvo įžvelgti jų funkcionalumo trūkumai. Atsižvelgus į juos, suformuluoti reikalavimai, kuriuos turėtų tenkinti pažangesnė, neregio poreikius tenkinanti, programa.

6.1. Prisijungimas

1. Įvedamas elektroninis paštas.
2. Įvedamas slaptažodis.
3. Tikrinama, ar egzistuoja vartotojas su pateiktais duomenimis.
4. Jei vartotojas egzistuoja, jis yra nukreipiamas į programėlės langą, kuriame yra rodomi visi prisijungę programėlės naudotojai.
5. Jei vartotojas neegzistuoja, jis yra nukreipiamas į registracijos langą.
6. Nuo antro karto jungiantis programėlė įsimena prisijungimo vardą bei slaptažodį ir automatiškai prijungia naudotoją prie programėlės – automatinis nukreipimas į langą, kur yra pateikiami visi prisijungę naudotojai.
7. Slaptažodis negali būti trumpesnis nei 6 simbolių.

Žr. 12 pav.

6.2. Registracija.

1. Įvedamas elektroninis paštas.
2. Įvedamas slaptažodis.
3. Įvedamas, antrą kartą, slaptažodis.
4. Patikrinama, ar įvesti slaptažodžiai sutampa.
5. Patikrinama, ar įvestas elektroninis paštas atitinka elektroninio pašto struktūrą.
6. Slaptažodžių ilgiai negali būti trumpesni nei 6 simboliai.

Žr. 13 pav..

6.3. Naudotojų sąrašas

1. Programėlės naudotojų lange išrikiuoti visi prisijungę naudotojai.
2. Išvardinti visi prisijungusių naudotojų elektroninio pašto adresai.
3. Patekus į naudotojų sąrašo langą, programėlė priskiria kiekvienam naudotojui identifikacijos numerį.
4. Programėlė išvardija naudotojui visus prisijungusius naudotojus su jų identifikacijos numeriu.

5. Galimybė atsijungti, jog nerodytu naudotojo kitų programėlės naudotojų sąraše.
6. Atsijungus, galimybė prisijungti.
7. Galimybė surasti naudotoją žodžiu, pagal jo identifikacinį numeri.
8. Galimybė surasti naudotoją paspaudus ant jo naudotojų sąraše.
9. Garsiniu signalu neradus naudotojo, pranešama, jog jis nerastas.

Žr. 14, 15, 16 pav.

6.4. Navigavimas iki pasirinkto programėlės naudotojo

1. Programėlė apskaičiuoja artimiausią maršrutą pėsčiomis iki pasirinkto asmens.
2. Programėlė apskaičiuoja kelio taškus, kuriuose kelias drastiškai pakeičia kryptį.
3. Telefonas pradeda vibruoti, jei naudotojas nėra atsisukęs į artimiausią kelio tašką arba kelionės finišą.
4. Programėlė praneša naudotojui, kada jis pradeda artėti prie kelionės tikslo arba artimiausio kelio taško.
5. Programėlė praneša naudotojui, kada jis pradeda tolėti nuo kelionės tikslo arba artimiausio kelio taško.
6. Programėlė, kas ketvirtį atstumo nuo likusio atstumo iki kelionės tikslo arba artimiausio kelio taško, praneša likusį atstumą metrais.
7. Programėlė, pasiekus kelio tašką įjungia kompasą ir praneša naudotojui, jog sukdamasis jis surastų kito kelio taško arba kelionės tikslo kryptį.
8. Programėlės naudotojui suradus naują kryptį, pranešamas atstumas metrais iki kito kelio taško arba kelionės tikslo bei, kiek laiko minutėmis programėlės naudotojas užtruks, kol pasieks tikslą.
9. Pasiekus kelionės tikslą, garsiniu signalu pranešama naudotojui, jog jis pasiektas.
10. Vartotojui panorėjus, programėlė praneša jam dabartinį jo lokacijos vietos adresą.
11. Programėlė bendrauja su naudotoju garsiniais signalais ir telefono vibracija.

Žr. 18, 19 pav.

6.5. Kasdienių objektų atpažinimas

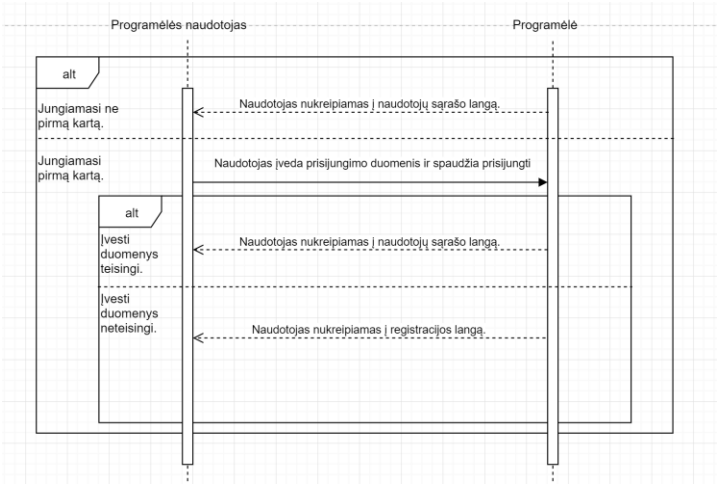
1. Programėlė gali atpažinti kasdienius objektus bei jų spalvą.
2. Programėlė atpažinimo rezultatus praneša naudotojui.

Žr. 17 pav.

7. Sukurtos programėlės „Blindify“ komponentų diagramos

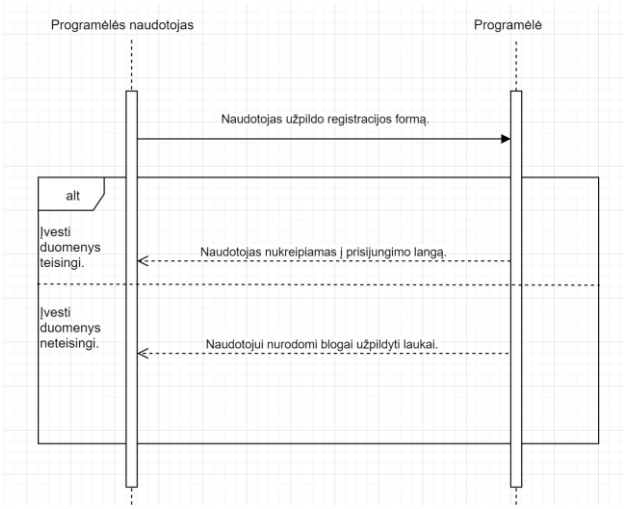
Šiame skyriube pateiktos programėlės komponentų diagramos. Jos vaizduoja programėlės sąveiką su naudotoju.

7.1. Prisijungimo lango diagrama



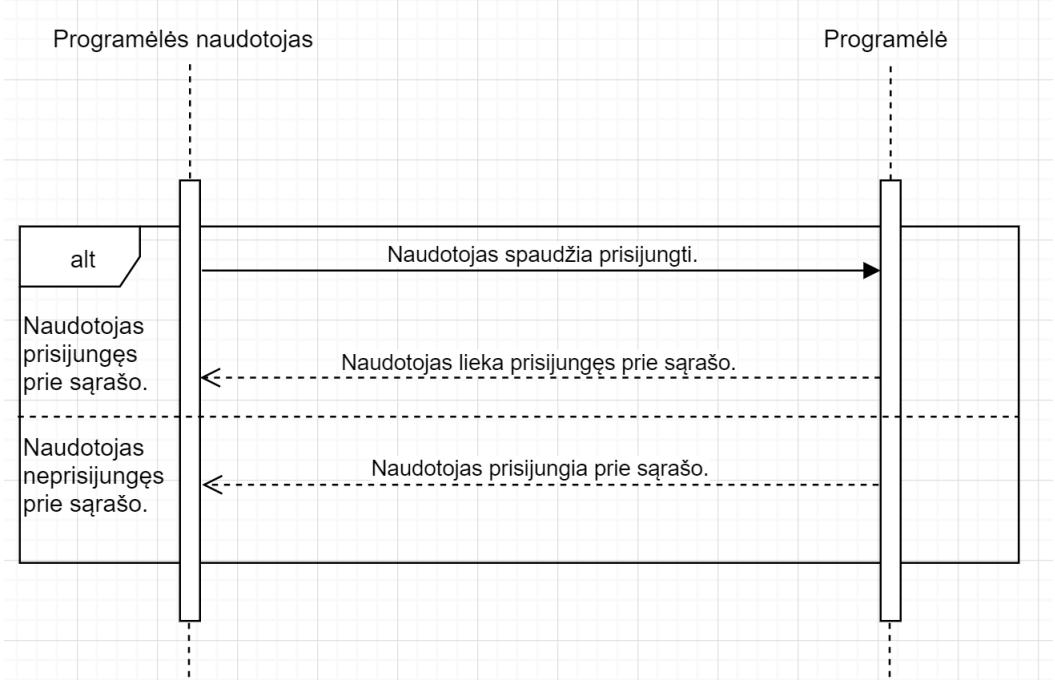
12 pav. Prisijungimo lango diagrama.

7.2. Registracijos lango diagrama



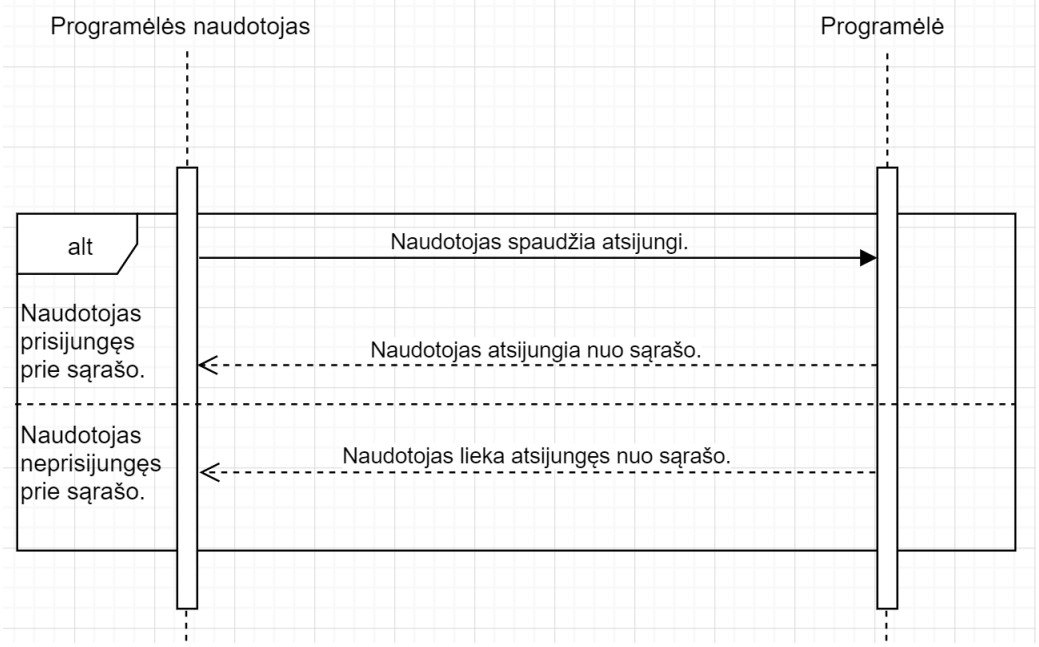
13 pav. Registracijos lango diagrama.

7.3. Naudotojo prisijungimo prie sąrašo diagrama



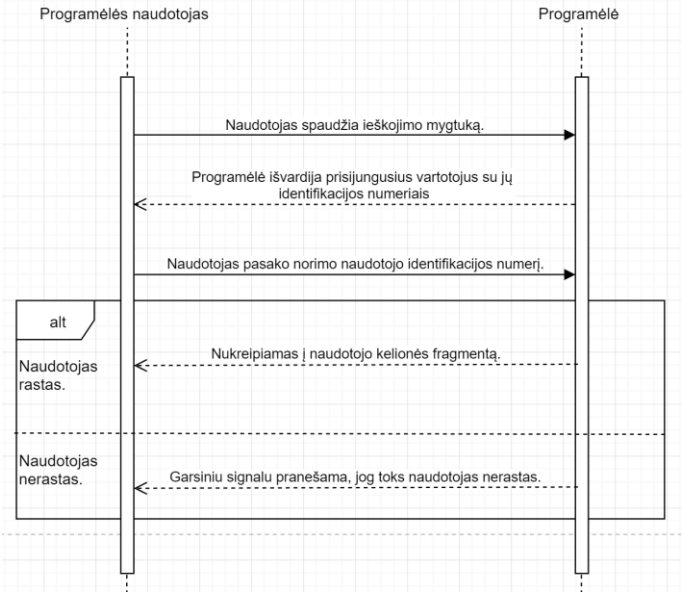
14 pav. Naudotojo prisijungimo prie sąrašo diagrama.

7.4. Naudotojo atsijungimo nuo sąrašo diagrama



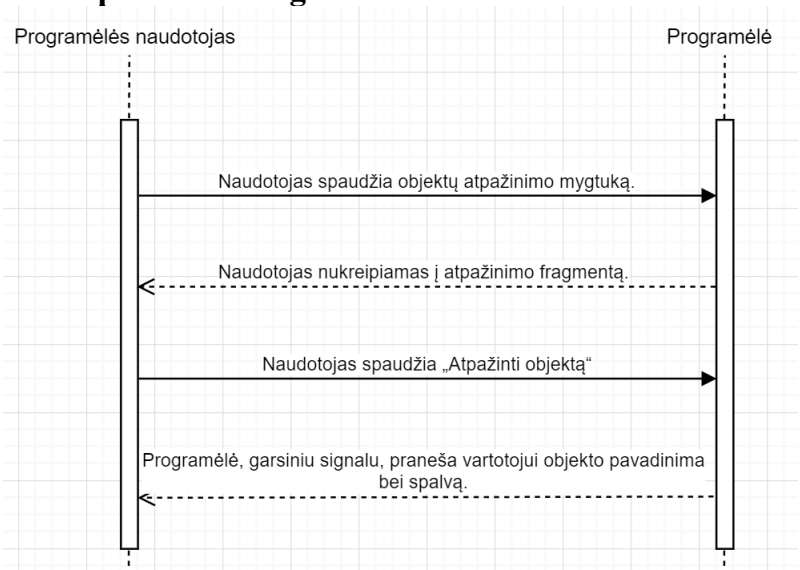
15 pav. Naudotojo atsijungimo nuo sąrašo diagrama.

7.5. Prisijungusio naudotojo radimas sąraše



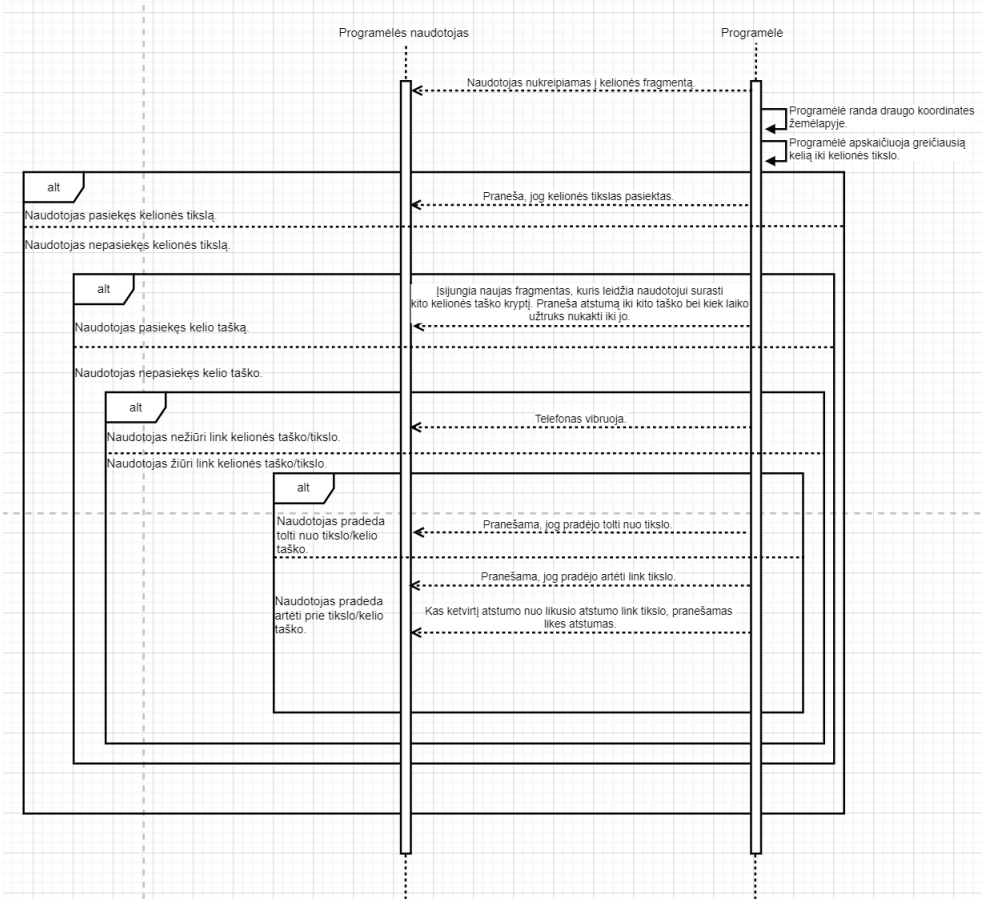
16 pav. Prisijungusio naudoto radimo sąraše diagrama.

7.6. Objekto atpažinimo diagrama



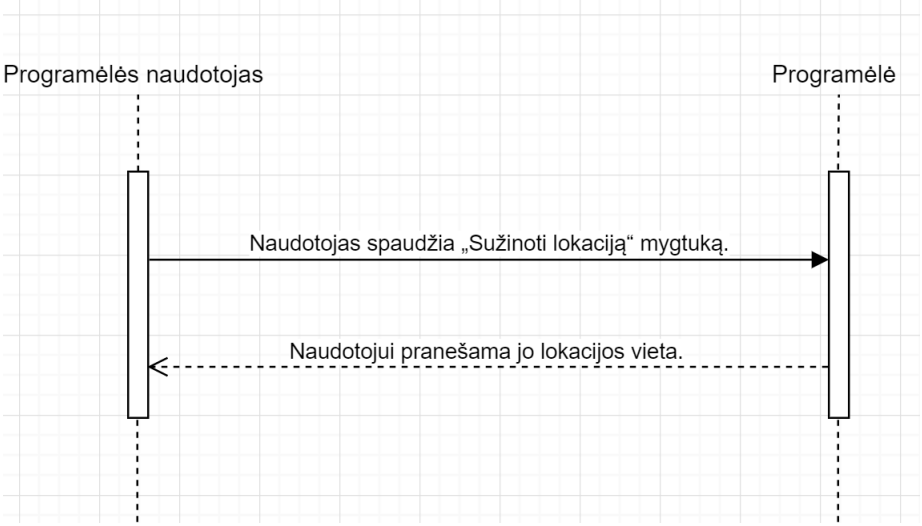
17 pav. Objekto atpažinimo diagrama.

7.7. Naudotojo kelionė link kelionės tikslo



18 pav. Naudotojo kelionės link kelionės tikslo diagrama.

7.8. Naudotojo lokacijos radimo diagrama



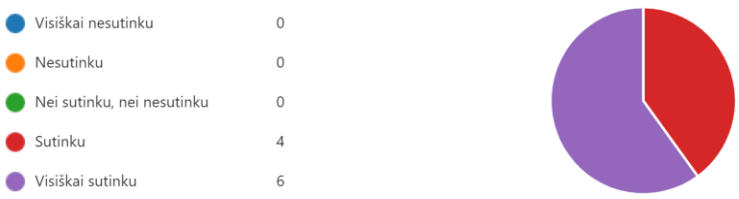
19 pav. Naudoto lokacijos radimo diagrama.

8. Programėlės testavimas

Išmanioji programėlė buvo ištestuota 10 - ies asmenų. Dėl aklų žmonių ypač sunkaus radimo, testuotojų aibė buvo pasirinkta sveiki asmenys. Programėlės bandymo metu, žmonėms buvo užrišamos akys, taip stengtasi imituoti aklo žmogaus pojūčius. Išbandžius programėlę, kiekvienas asmuo užpildė klausimyną, paruoštą pagal „Likterio“ skalę, kuriame buvo pateikti klausimai, susiję su neregio žmogaus naudojimosi sukurta programėle. Norintys asmenys, taip pat, parašė po trumpą atsiliepimą. Gautus rezultatus matysite žemiau šiame bakalauriniame darbe.

8.1. Atliktos apklausos rezultatai

1. Regos negalią turinčiam asmeniui yra lengva prisijungti programėlėje.



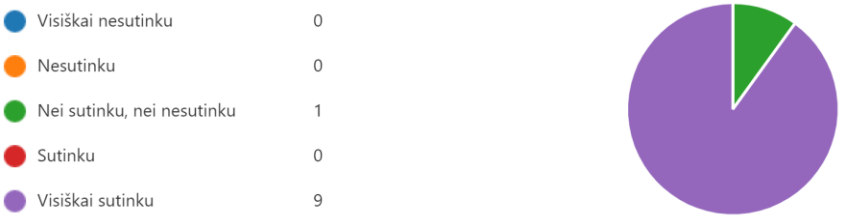
33 pav. Rezultatai apie prisijungimą.

2. Regos negalią turinčiam asmeniui yra lengva užsiregistruoti programėlėje.



34 pav. Rezultatai apie registravimąsi programėlėje.

3. Regos negalią turinčiam asmeniui yra lengva pasirinkti naudotoją, iki kurio nori nukeliauti.



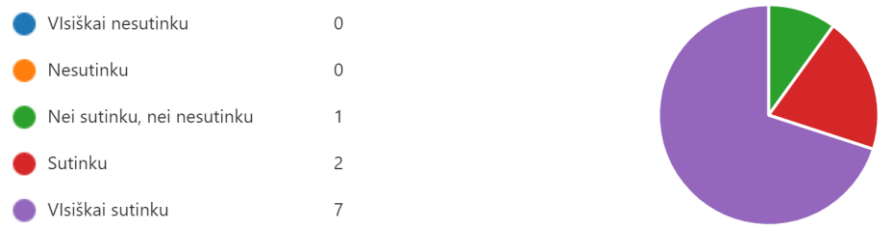
35 pav. Naudotojo pasirinkimas.

4. Aklajam yra lengva prisijungti/atsijungti iš prisijungusių asmenų sąrašo.



36 pav. Prisijungimas/atsijungimas iš sąrašo.

5. Regos sutrikimų turinčiam asmeniui yra aišku, kada jis pradeda artėti/tolėti link kelionės tikslo/kelio taško.



37 pav. Rezultatai apie artėjimą/tolėjimą link kelionės tikslo/kelio taško.

6. Regos negalią turinčiam asmeniui programėlė pateikia pakankamai informacijos, kad ja būtų lengva naudotis.



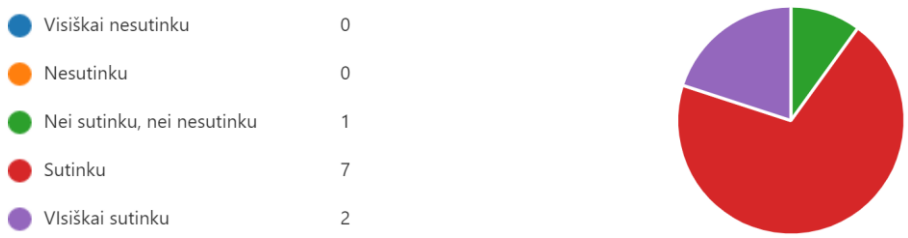
38 pav. Rezultatai apie informacijos pateikimą.

7. Programėlės naudotojui yra lengva sužinoti dabartinę lokacijos vietą.



39 pav. Lokacijos vietos sužinojimas.

8. Aklajam yra lengva atpažinti aplinkinius objektus.



40 pav. Aplinkinių objektų atpažinimo lengvumo rezultatai.

9. Programėlė tiksliai nuveda bandytoją link kelionės tikslo.



41 pav. Programėlės tikslumo rezultatai.

10. Programėlė yra lengvai perprantama, regos negalią turinčiam asmeniui, bei ja neužtruks išmokti naudotis.



42 pav. Naudotojų rezultatai apie programėlės perpratimo lengvumą.

8.2. Išmaniosios programėlės testavusių asmenų atsiliepimai

Žemiau pateikti žmonių, kurie sutiko parašyti atsiliepimą apie išbandytą programėlę nuomonę. Bandytojai aprašo savo išgyvenimus iš aklo asmens perspektyvos.

1. Testuotojas nr. 1.

Programėlė yra patogi ir greitai perprantama. Pagrindinis funkcionalumas - pagalba naviguojant iki draugo – veikia gerai, aplikacija puikiai atlieka savo paskirtį. Vienintelis minusas yra pradėjimas naudotis programėle, kadangi reikalingas žmogus, kuris padėtų užsiregistruoti ir supažindintų su programa.

2. Testuotojas nr. 2.

Programa „Blindify“ skirta žmonėms, turintiems regos negalią. Programa, su artimo žmogaus pagalba, lengva įrašyti į telefoną ir prisijungti prie jos. Po to, artimųjų pagalbos jau nereikia. Regos negalią turintys žmonės, naudodamiesi šia programa, turi galimybę tapti saugesni ir savarankiškesni, gali susisiekti ir susitikti pabendrauti tarpusavyje, programa, kameros pagalba, lengvai atpažįsta objektus, sutiktus pakeliui, tiksliai informuoja apie likusį atstumą iki numatyto kelionės tikslo, lengvai nustato lokacijos vietą. Programos trūkumas: ne visai tiksliai nustato lokacijos vietą, jeigu ja naudojamasi patalpų viduje.

3. Testuotojas nr. 3.

Programėlė yra sukurta aklam žmogui, todėl valdymas yra ypač paprastas ir lengvai išmokstamas. Labai gerai, kad yra objektų atpažinimo sistema, kuri leidžia naudotojui atpažinti aplinką. Su šita programėle, aklas žmogus žinos gatvę, kurioje yra bei daug lengviau pasieks kelionės tikslą.

4. Testuotojas nr. 4.

Programėlė „Blindify“ ne tik suteikia šansą negalią turinčiam asmeniui atlikti kasdienes užduotis, kurios, paprastai, butų neįgyvendinamos kaip nukakti iki draugo, bet taip pat yra itin paprastai naudojama, prieinama tiek jaunimui, tiek vyresnio amžiaus naudotojui. Programėlėje esant objektų atpažinimo funkcija naudotojui leidžia pasijusti saugesniu bei labiau užtikrintu keliaujant iš taško A į tašką B. Programėlė, paremta Google maps“, yra tiksli ir užtikrina

vartotojui saugumo jausmą, kad ir kur jis bebūtų, nesvarbu ar tai jam žinoma aplinka, ar nauja vietovė.

Taigi, atsižvelgiant į užpildytos apklausos rezultatus bei parašytus atsiliepimus, būtų galima teigti jog programėlė daugiau/mažiau yra lengvai perprantama naudotojui ir tikrai jam padėtų susiorientuoti mieste. Iš programėlės bandytojų sužinota, jog sunkiausia vieta yra registracija bei pirmasis prisijungimas, kai naudotojas turi pats užpildyti lange nurodytus laukus. Tai jam turėtų padėti padaryti kitas asmuo. Taip pat testuotojai buvo gana patenkinti, kaip programėlė bendrauja su naudotoju. Teigė, jog teikiamos informacijos ir programėlės GPS tikslumo pakanka, jog naudotojas galėtų daug saugiau nukakti iki kito naudotojo/draugo. Taip pat, didelė sukurto funkcionalumo teigiama savybė yra, jog programėlė daugelį darbo atlieka pati bei nereikalauja naudotojo įsikišimo.

9. Išnagrinėtų bei sukurtos programėlės funkcionalumo palyginimas

Pasirinktos programėlės, dvi jau anksčiau darbe išnagrinėtos programėlės: „AriadneGPS“ ir „GetThere“, bei sukurta - „Blindify“. Tikslas – palyginti programėles bei sužinoti panašumus bei skirtumus. „Blindify“ programėlės vertinimų rezultatai yra gaunami iš testuotojų atsiliepimų bei apklausos rezultatų, o likusių programėlių vertinimas yra šio bakalaurinio rašto darbo autoriaus.

	„Blindify“	„GetThere“	„AriadneGPS“
Programėle yra lengva naudotis.	+	+/-	+/-
Programėlė nesunkiai perprantama.	+	+	-
Navigavimas po funkcionalumą yra lengvas.	+	+	+/-
Programėlė padeda susiorientuoti mieste.	+	+	+
Programėlė pranešinėja likusį nukeliauti atstumą.	+	+	-
Programėlė atpažįsta aplinkinius objektus.	+	-	-
Programėlė leidžia išsaugoti norimas lokacijos vietas.	-	+	-
Programėlė yra gana savarankiška, nereikalauja naudotojo įsikišimo.	+	+	+
Programėlės mygtukai atitinka standartus.	+/-	+	-

Pagal nustatytus kriterijus visos trys programėlės buvo įvertintos. Galima teigti, jog šio bakalaurinio darbo tikslas – sukurti programėlę, tobulesnę už jau esamas, yra pateisintas.

10. Rekomendacijos

Remdamasis įgauta patirtimi, kuriant programėlę regos negalią turinčiam asmeniui, galiu suformuluoti eilę rekomendacijų. Sukūręs šią programėlę bei remiantis įgauta patirtimi bei informacija rekomenduojama, kuriant programėlę, regos negalią turinčiam asmeniui, būti atsakingais. Vartotojo sąsaja privalo būti lengvai perprantama. Aklas asmuo visiškai nemato mobiliojo įrenginio ekrano, dėl to turi būti išlaikytas programėlės fragmentų minimalizmas. Tai padidintų funkcionalumo įsisavinimo greitį. Vartotojo sąsaja turėtų atitikti kuriamų programėlių neįgaliesiems standartus, tokius kaip dideli mygtukai, stacionarūs programėlės fragmentai.

Programėlės skaičiavimų tikslumas ir patikimumas regos negalią turinčiam asmeniui privalo būti pagrindinis prioritetas. Netikslumai gali sukelti pavojų naudotojui realiame gyvenime. Taip pat, keliavimo programėlė taptų bevertė, jei ji nenuveda asmens į norimą vietą.

Trečia, sukurtas gaminys turi „gyventi nepriklausomai“ – nereikalauti didelio programėlės naudotojo įsikišimo. Funkcionalumas turėtų būti įgyvendintas taip, jog programėlei nereikėtų didelio neįgalaus asmens veiksmų skaičiaus. Taip regos negalią turinčiam asmeniui reikės mažiau įsiminti mygtukų bei programėlė taps lengviau naudojama.

Sukurtoje programėlėje yra aplinkybių, kurios galėtų būti tobulinamos. Pirmoji - registracijos nepritaikymas neįgaliam asmeniui. Registravimo fragmento duomenų užpildymą būtų galima padaryti žodžiu, t.y. jos naudotojas galėtų diktuodamas raides užsiregistruoti sistemoje. Antra netobula vieta yra prisijungimas pirmąjį kartą. Naudotojas turi suvesti prisijungimo duomenis savo rankomis – jam turi padėti kitas asmuo. Šį funkcionalumą taip pat būtų galima perdaryti į žodžiu valdomą prisijungimą. Trečia, kai kuriuose programos fragmentuose, pagal naudotojo atsiliepimus, galima padidinti mygtukus, jog jais būtų lengviau naudotis. Ketvirta, galvojant apie žymesnius patobulinimus, būtų galima įgyvendinti funkcionalumą, kuris pranešinėtų programos naudotojui, kada jam reikia kirsti kelią arba įgyvendinti funkciją, kuri palengvintų naudojimąsi viešuoju transportu.

11. Rezultatai bei išvados

Šio baigiamojo bakalaurinio darbo metu buvo norima sukurti išmaniąją programėlę „Android“ operacinei sistemai, kuri atneštų realią naudą regos sutrikimų turinčiam žmogui bei padėtų nukeliauti iki kito asmens/draugo mieste. Jog šį išsikeltą tikslą būtų galima įgyvendinti, visų pirma, reikėjo ištirti, kokie yra kuriamų programėlių akliems žmonėms standartai. Tai atlikus, buvo konstatuota, jog pritaikytos programėlės regos negalią turintiems asmenims privalo būti lengvai perprantamos. Jų grafinės sąsajos fragmentai turi išlaikyti tam tikrus dydžio standartus, jog būtų lengva neįgaliam asmeniui lengva naudotis. Pastebėta, jog didelis išmaniosios programėlės savarankiškumas (autonominis darbas) sumažina galimybę ja pasinaudoti klaidingai bei atneša didesnę vertę.

Taip pat, ištyrinėjus integruotą „Android“ operacinės sistemos funkcionalumą, galima teigti, jog šios sistemos turimų priemonių kartu su išoriniais įskiepiais, tokiais kaip „Google Maps“ bei „Firebase“, pakanka neregio asmens navigavimui aplinkoje.

Išanalizavus keletą sukurtų programėlių pritaikytų aklo asmens navigavimui, buvo pastebėti pagrindiniai funkcionalumo pliusai bei minusai kuriant šias programėles. Atsižvelgiant į juos bei išanalizuotus standartus, matant vertinamas alternatyvas, buvo sukurta nauja išmanioji programėlė. Jos tikslas yra palengvinti neregio asmens navigavimą keliaujant iki kito programėlės naudotojo. Sukurtos programėlės naudojimosi galimybės bei savybės yra peteiktos žemiau.

- Programėlės naudotojas turi galimybę prisijungti prie jos ir matyti prisijungusiu žmonių sąrašą.
- Aklas asmuo gali atsijunti/prisijungti prie prisijungusių asmenų sąrašo – tapti nematomas kitiems naudotojams.
- Asmuo turi galimybę užsiregistruoti programėlėje.
- Naudotojas gali žodžiu pasirinkti asmenį iš sąrašo, iki kurio nori nueiti.
- Programėlė apskaičiuoja greičiausią kelią iki norimo asmens ir padeda naudotojui iki jo nukakti.
- Telefonas vibruoja, jei asmuo neina tinkama kryptimi.
- Programėlė naudotojui pranešinėja likusį atstumą kas ketvirtadalį kelio. Ji, taip pat, praneša, kiek jis užtruks nukeliauti iki kito kelio taško arba kelionės tikslo ir koks atstumas jo laukia.
- Programėlė apskaičiuoja kelio taškus, kuriuos naudotojui reikės pasiekti, jog būtų pasiektas kelionės tikslas.
- Naudotojas bet kada gali sužinoti savo dabartinės lokacijos vietą.
- Su programėle aklas asmuo gali atpažinti aplinkinius objektus bei jų spalvą.

Įgyvendinus aukščiau aprašytą funkcionalumą, programėlė buvo duota išbandyti 10 – čiai asmenų. Atsižvelgus į įvertinimus, ši programėlė yra naudinga vartotojui naviguojant iki kito programėlės naudotojo. Pagal bandytojų nuomonę, vienintelis trūkumas yra naudotojo registracija programėlėje, tačiau kitas funkcionalumas yra išdirbtas gerai. Visi bandytojai pasiekė kelionės tikslą.

Taip pat, sukurta programėlė pagal tam tikrus kriterijus buvo palyginta su kitomis išnagrinėtomis navigavimo programėlėmis ir buvo pastebėta, jog ji yra pranašesnė nei jau egzistuojančios. Paruošus programėlę platinimui, būtų galima ja pasidalinti tokiose platformose kaip „Google Play“.

12. Summary

According to fast human population growth, it is well known that the number of blind people is increasing too. They need to adapt in our fast suciety. Mobile applications can help them do that. There are a lot of developed mobile applications but only few are adapted for visual impaired people. This batchelor theses‘s purpose is:

- Investigate the standart “Android” operation system functionality for blind people.
- Try a few third party provider’s created applications.
- Find out best mobile application creation standards.
- Create mobile application to help user navigate to a friend.
- Test the created application with 10 volunteers.
- Compare the application with others.

During the work the mobile application was created. Application’s purpose is to help blind user navigate to his friend (other user). Application’s functionality:

- User has a possibility to register.
- User can login. Application saves login data so user does not need to enter data every time he goes online.
- User can select travel destination by voice.
- Customer can get the address of his location.
- Application lets customer detect surrounding objects.
- Application leads user to the final destination.
- Application does not let to stray from the direction.
- Customer knows when he starts to get closer to the goal and opposite.
- User is always informed about the distance and time.

Created application was tested with 10 volunteers. Feedback was positive, except the registration part. According the volunteers the registration part needs to be improved and adapted more for visual impaired people. The main functionality performs well. Every volunteer reached their final destination.

13. Literatūros sąrašas

[I1] „Statista“. URL: https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/

[I2] „Statista“. URL: https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads/

[MFT05] Jennifer Mankoff, Holly Fait and Tu Tran. Is Your Web Page Accessible? A Comparative Study of Methods for Assessing Web Page Accessibility for The Blind.https://www.researchgate.net/publication/221515801_Is_your_web_page_accessible_A_comparative_study_of_methods_for_assessing_web_page_accessibility_for_the_blind. 333 KB, Portland, Oregon, USA, 2005.04.2-7.

[Dob17] Krzysztof Dobosz. Designing Mobile Applications For Visually Impaired People. In: *Visually Impaired: Assistive Technologies, Challenges and Coping Strategies*, Nova Science Publishers. Editors: Judy Estrada.

[Ego17] Anastasia Egorina. Accessibility: what you need to know about mobile app development for people with special needs. URL:https://medium.com/rosberryapps/accessibility-what-you-need-to-know-about-mobile-app-development-for-people-with-special-needs-b52d713f8c95. 2017.12.14.

[I3] „ W3C“. URL:https://www.w3.org/TR/UNDERSTANDING-WCAG20/visual-audio-contrast-contrast.html.

[ST12] Javier Sánchez Sierra, Joaquín Selva Roca de Togores. Designing Mobile Apps for Visually Impaired and Blind Users. https://www.researchgate.net/publication/290677649_Designing_mobile_apps_for_visually_impaired_and_blind_users_Using_touch_screen_based_mobile_devices_IPhoneiPad. 2012.

[Zaf17] Ayesha Zafar. How to design accessible mobile experiences for the blind. URL: https://willowtreeapps.com/ideas/how-to-design-accessible-mobile-experiences-for-the-blind. 2017.12.

[I4] „World Health Organization“. Blindness and vision impairment. URL:https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment. 2018.11.11.

[I5] https://github.com/MindorksOpenSource/AndroidTensorFlowMachineLearningExample

[I6] „Firebase“. URL:https://firebase.google.com/docs/android/setup.

[I7] „Google Maps“ URL: https://developers.google.com/maps/documentation/directions/intro.

[I8] „Google Developer Console“ URL: https://console.developers.google.com/

[I9] „Vespa“. Json. URL: https://docs.vespa.ai/documentation/reference/document-json-format.html