

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

Baigiamasis bakalauro darbas

Mobilioji asmeninių finansų programėlė

Atliko:
Justina Šakalytė parašas

Vadovas:
Asist. dr. Alminas Čivilis

Vilnius
2019

Turinys

Sutartinis terminų žodynas **3**

Santrauka **4**

Summary **5**

Įvadas **6**

1. Mobiliosios programėlės tipo pasirinkimas **7**

2. Susijusių darbų analizė **8**

3. Teorinis programėlės modelis **10**
3.1. Veiklų diagrama 10
3.2. Naudojimo atvejai 11
3.3. Duomenų bazės diagrama 12

4. Programėlės įgyvendinimas **13**
4.1. Pasirinktos technologijos 13
4.2. Logotipo atpažinimo technologija 14
4.3. Įgyvendintų sprendimų analizė 15
4.3.1. Išlaidų modulis 15
4.3.2. Išlaidų čekio nuotraukos analizės metodas 17
4.3.3. Pajamų modulis 18
4.3.4. Biudžetų modulis 18
4.3.5. Taupymo planų modulis 20

5. Programėlės funkcionalumo testavimas **21**

Išvados ir rekomendacijos **23**

Ateities tyrimų planas **24**

Literatūros šaltiniai **25**

Sutartinis terminų žodynas

Android Studio IDE - oficiali Android programavimo aplinka, skirta programėlių kūrimui;
SQLite - duomenų bazė;
Šablono atitikimo metodas (angl. template matching) - kompiuterinės vizijos technika, naudojama aptikti šablono buvimo vietą pateiktame paveikslėlyje;
OpenCV - atviro naudojimo kompiuterinės vizijos sprendimų biblioteka;

Santrauka

Darbe pristatoma mobilioji Android programėlė, skirta asmeninių finansų planavimui ir registravimui. Ši programėlė suteikia galimybę naudotojui registruoti savo išlaidas ir pajamas, jas paskirstyti į pasirinktas kategorijas, matyti išlaidų pasiskirstymo analizę, planuoti savo biudžetus, skirtus kiekvienai iš kategorijų, bei susikurti taupymo planus. Programėlės kūrime naudojama kompiuterinės vizijos biblioteka padeda įgyvendinti funkciją, palengvinančią duomenų įvedimą – tai čekio nuotraukos analizė, aptinkanti čekyje pavaizduotą pardavėjo logotipą ir priskirianti išlaidas atpažintai išlaidų kategorijai.

Summary

Personal Finances Mobile Application

This work presents an Android mobile application for user's personal finances accounting. This application provides an ability for a user to register his own expenses and incomes, categorize them, see the visualized report of user's added expenses, plan his budgets for each of the categories and create saving plans. In the development of the application a computer vision library was used, which provides a way to develop a function of logo detection in the user's provided picture of the receipt. The template matching method is used for detection of the logo and then the category of the expenses is set. This function helps the user to insert the data of his expenses. The application was developed using Java programming language, Android Studio IDE, SQLite database and OpenCV open source computer vision library.

Įvadas

Šiuo metu yra sukurta daugybė įvairių finansinių programėlių - nuo oficialių bankinių programėlių iki paprastų, kasdienių finansų skaičiavimą ir sekimą palengvinančių programėlių. Kai rinkoje yra tokia didelė pasiūla, sunku įgyvendinti funkcionalumą, kurio dar neturi kita panašaus tipo programėlė. Šio darbo tikslas yra sukurti tokią asmeninių finansų programėlę, kuri padėtų naudotojui registruoti savo išlaidas ir pajamas, planuoti biudžetą, susikurti taupymo planus, ir turėtų papildomą funkcionalumą – galimybę nuskenuoti ir atpažinti čekyje pavaizduotą logotipą, taip priskiriant čekio išlaidas atitinkamai atpažintai kategorijai. Ši funkcija įgyvendinama pasitelkiant kompiuterinės vizijos sprendimus. Atlikus panašių programėlių analizę, buvo bandoma sukurti visas būtiniausias ir vartotojui naudingas funkcijas turinčią programėlę. Šiam darbo tikslui įgyvendinti buvo išsikelti tokie uždaviniai:

- Išnagrinėti kelias susijusias ir vienas populiaresnių finansų valdymo programėlių;
- Suprojektuoti teorinį programėlės modelį;
- Praktiškai įgyvendinti numatytą programėlės funkcionalumą ir dizainą;
- Įgyvendinti logotipų atpažinimo funkciją;
- Atlikti aplikacijos testavimą;

Programėlė kuriama naudojantis Android Studio IDE ir Java programavimo kalba, SQLite duomenų bazę, bei pasitelkiant OpenCV kompiuterinės vizijos biblioteką.

1. Mobiliosios programėlės tipo pasirinkimas

Programėlės, pagal savo sukūrimo tipą, yra skirstomos į tris kategorijas [7]. Tai vietinės (angl. native), hibridinės (angl. hybrid) ir mobilaus internetinio puslapio (angl. Mobile web apps) programėlės. Pastarojo – mobilaus internetinio puslapio – tipo programėlės veikia mobiliojo įrenginio naršyklės lange. Šios programėlės yra kuriamos naudojantis HTML, Javascript ir CSS programavimo kalbomis. Tokių programėlių funkcionalumą ir grafinę sąsają yra lengva atnaujinti, nes visi atnaujinimai atsiranda žiniatinklio UI, nereikia siųsti užklausos vartotojui, su prašymu atnaujinti programinę įrangą. Ši programėlė, skirtingai nei vietinio tipo programėlė, negali išnaudoti mobiliojo įrenginio komponentų, tokių kaip kamera, kalendorius, naudotojo buvimo vieta ir panašiai. Taip pat mobilaus internetinio puslapio programėlės veikia įrenginio naršyklės lange ir negali būti pasiekiamos be interneto ryšio.

Vietinio tipo programėlės, kurios turi būti parsisiųstos iš internetinės programėlių parduotuvės ir įdiegtos į mobilųjį įrenginį, gali naudoti visas gamyklines įrenginio funkcijas, naudotojui suteikus prieigą. Pagrindinis vietinių programėlių pliusas – jos gali veikti be interneto ryšio, arba bent jau svarbiausios funkcijos apsieina be interneto ryšio, taip pat šių programėlių vartotojo sąsaja gali būti kuriama neribotomis galimybėmis.

Hibridinio tipo programėlių naudotojo sąsajos galimybės yra mažesnės nei vietinio tipo, nes šios programėlės negali naudoti visų gamyklinių įrenginio funkcijų, tačiau taip pat turi savų pliusų. Tokia programėlė gali būti lengvai perkuriama kitoms operacinėms sistemoms. Hibridinės programėlės yra programuojamos HTML, Javascript ir CSS. Šios programėlės apjungia vietinio ir mobilaus internetinio puslapio aplikacijų galimybes.

Atlikus programėlių kūrimo technologijų analizę, buvo pasirinkta kurti Android operacinės sistemos vietinę (angl. native) programėlę. Šiame darbe aprašomos programėlės funkcijai – čekio logotipo atpažinimui - yra reikalinga prieiga prie įrenginio vidinės atminties, o vietinio tipo programėlė gali gauti prieigą visų įrenginio komponentų. Planuojama kurti programėlė neprivalo turėti nepertraukiamo interneto ryšio, nes visus duomenis įveda naudotojas ir jie yra laikomi lokalioje SQLite duomenų bazėje.

2. Susijusių darbų analizė

Nors ir dažnai girdime, kad šiuolaikinis jaunimas su pinigais tvarkosi žymiai geriau, nei senesnių kartų atstovai, Lietuvos bankų asociacijos tyrimas parodė, kad mūsų šalies finansinio raštingumo indeksas siekia tik 43 balus iš galimų 100 [3]. Tai reiškia, kad didžioji dalis gyventojų sugeba tvarkytis su paprastomis kasdieninėmis piniginėmis operacijomis, tačiau prastai planuoja savo biudžetus, nemoka taupyti. Norint išmokti taupyti, svarbu yra užsibrėžti tikslą, kurio būtų norima pasiekti. Taip pat, norint pagerinti savo finansinį raštingumą, yra siūloma [5] planuotis savo biudžetus, skaičiuojant kiek pinigų išleidžiame būtiniausiems dalykams ir kiek galime atidėti ateities planams įgyvendinti. Vienas iš įrankių, galintis palengvinti apskaičiuoti ir tvarkyti savo asmeninius finansus yra mobiliosios aplikacijos.

Oficialioje šios operacinės sistemos programėlių internetinėje parduotuvėje Google Play Store, 2019 metų kovo mėnesio duomenimis [1], yra patalpinta virš 2,6 milijono programėlių. Šios programėlės yra suskirstytos į 36 skirtingas kategorijas, o finansų kategorija turi virš 69 tūkst. programėlių. Iš minėtos internetinės parduotuvės finansų kategorijos buvo išsirinktos kelios programėlės ir atlikta jų analizė. Programėlės buvo pasirinktos pagal populiarumą, taip pat atsižvelgiant į programėlės kalbą, ir randamos į paiešką įvedus raktažodžius „budget“, „finance“, „money“.

Pirmoji programėlė, pasirinkta analizei – Bluecoins Finance [2]. Šios programėlės pagalba naudotojas gali planuoti savo biudžetą, registruoti savo išlaidas ir pajamas, matyti vizualias ataskaitas apie jas - stulpelines diagramas, kreives ir skritulines diagramas. Taip pat naudotojui yra suteikiama galimybė registruojant savo išlaidas prisegti čekio nuotrauką. Ši programėlė suteikia galimybę išlaidas priskirti atitinkamoms kategorijoms. Egzistuojančių kategorijų kiekis yra nedidelis – tik 12 kategorijų, tačiau naudotojui yra suteikta galimybė pačiam susikurti savo kategorijas ir priskirti joms spalvas bei ikonas.

Dar viena programėlė, pasirinkta analizei – Money Lover [6]. Ši programėlė yra labai panaši į pirmąją analizuotą programėlę – visos išlaidos gali būti išskirstytos pagal kategorijas, tačiau šioje programėlėje yra gausus jau sukurtų kategorijų pasirinkimas, taip pat galima įplaukas išskirti kategorijomis – tokio funkcionalumo neturi pirmoji analizuota programėlė. Kaip ir didžioji dalis finansų programėlių, ši taip pat leidžia naudotojui peržiūrėti išlaidų ir pajamų ataskaitas, susiplanuoti biudžeto paskirstymą. Money Lover programėlė turi papildomą funkcionalumą – naudotojas, registruodamas savo patirtas išlaidas, gali nufotografuoti išlaidų čekį. Programėlė išanalizuoja nufotografuotą čekį, aptinka ir nustato galutinę apmokėtą sumą ir priskiria išlaidų dydį. Nufotografuoto čekio analizė užtrunka ilgai - apie 5 minutes, analizė vyksta fone, naudotojas gali toliau naudotis mobiliuoju įrenginiu ir gauna pranešimą, kai analizė būna užbaigta.

Kita analizuojama programėlė – Wallet [13]. Ši programėlė yra labai paprasta naudoti, turi daug sukurtų kategorijų, kurioms galima priskirti išlaidas ir įplaukas. Naudotojas neužtrunka daug laiko išlaidų registravimui, nes programėlės grafinė sąsaja nėra perkrauta funkcijomis. Egzistuoja funkcija - prisegti čekio nuotrauką prie registruojamos išlaidos. Tai pat yra galimybė planuoti ir paskirti biudžetą atitinkamai kategorijai, peržiūrėti išlaidų ir įplaukų diagramas. Šioje programėlėje galima pasirinkti turinį matyti lietuvių kalba, tačiau kai kurie meniu punktai išlieka anglų kalba.

Ketvirtoji analizuojama programėlė – Spendee [10]. Šios programėlės funkcionalumas yra labai panašus į kitų analizuojamų programėlių funkcionalumą. Šioje programėlėje taip pat yra galimybė priskirti išlaidas kategorijoms, paskirstyti savo biudžetą, peržiūrėti išlaidų ir įplaukų diagramas. Programėlė suteikia galimybę prisegti čekio nuotrauką išlaidų registravimo metu. Funkcionalumas, kurio neturi ši programėlė, tačiau turi keletas iš prieš tai analizuotų – taupymo tikslo kūrimas. Taupymo planavimas yra stipri motyvacija žmogui pradėti taupyti jo norimais tikslais,

1 lentelė. Panašių programėlių analizė

	Bluecoins finance	Money lover	Wallet	Spendee	Mano piniginė
Išlaidų priskyrimas kategorijai	+	+	+	+	+
Biudžeto paskirstymas pagal kategorijas	+	+	+	+	-
Pirkimo čekio duomenų analizė	-	+	-	-	-
Išlaidų diagramos peržiūra	+	+	+	+	+
Taupymo tikslo kūrimas	-	+	+	-	-

todėl šios funkcijos trūkumas kitose programėlėse yra gana didelis minusas.

Paskutinė analizuojama programėlė – Mano Piniginė [4]. Tai programėlė, vienintelė iš analizuojamų, yra įgyvendinta lietuvių kalba. Ši programėlė suteikia galimybę išlaidas priskirti kategorijoms. Jau sukurtų kategorijų kiekis nėra didelis, naudotojas gali sukurti savo kategorijas, tačiau kategorijos nėra suskirstytos spalvomis ar pagal ikonas. Naudotojas taip pat gali pamatyti savo išlaidų ir įplaukų ataskaitas. Keletas trūkumų, kurie kol kas neįgyvendinti šioje programėlėje – naudotojui nėra suteikta galimybė prie išlaidų pridėti nufotografuoto čekio nuotrauką, taip pat nėra galimybės susikurti taupymo tikslą, paskirstyti savo biudžetą skirtingoms pirkinių kategorijoms.

Visų analizuotų programėlių funkcionalumo palyginimas yra vaizduojamas 1 lentelėje. Atlikus šią analizę buvo pastebėta, kad egzistuoja funkcijos, kurios įgyvendintos visose aukščiau aptartose programėlėse – tai išlaidų diagramų peržiūra ir išlaidų priskyrimas kategorijai. Biudžeto paskirstymo pagal kategorijas funkcionalumas egzistuoja didžiojoje dalyje analizuotų programėlių. Taip pat buvo pastebėta, kad daugumoje aplikacijų yra galimybė nufotografuoti ar prisegti čekio nuotrauką, tačiau tik viena iš jų išanalizuoja pateiktą čekio nuotrauką ir panaudoja duomenis išlaidų įvedimo palengvinimui. Taip pat taupymo tikslo kūrimas įgyvendintas tik keliose iš analizuotų programėlių.

Šiuo darbu planuojama sukurti tokį teorinį ir praktinį programėlės modelį, kuris turėtų visą 1 lentelėje paminėtą funkcionalumą. Esminis skirtumas tarp analizuotų programėlių ir šiame darbe kuriamos programėlės bus čekio nuotraukos analizė, kuri turėtų palengvinti naudotojo išlaidų kategorijos įvedimą.

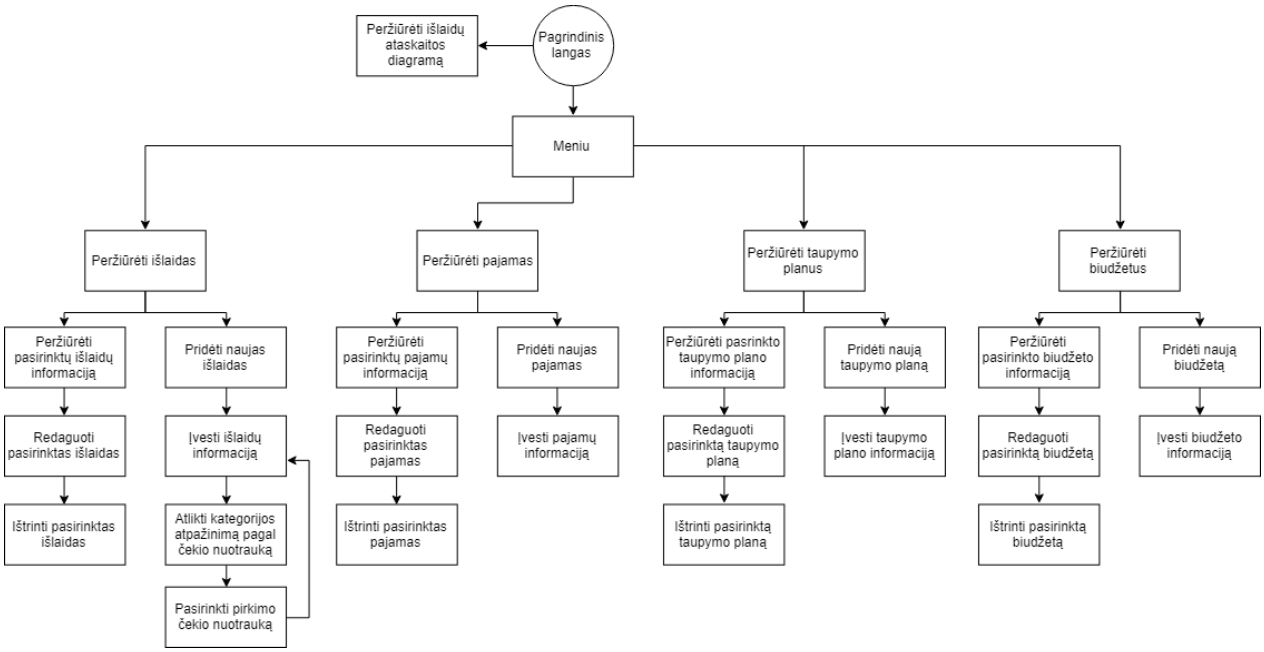
3. Teorinis programėlės modelis

Atlikus panašių programėlių analizę, buvo nuspręsta, kokie reikalavimai turėtų būti išsikelti ir įgyvendinti. Programėlė turėtų turėti išlaidų bei pajamų registravimo ir peržiūros modulius, biudžetų ir taupymo planų kūrimo ir peržiūros modulius, bei išlaidų ataskaitos peržiūros modulį. Vienas iš pagrindinių šios programėlės funkcionalumų yra pirkimo čekio logotipo atpažinimas ir automatinis priskyrimas išlaidų kategorijai - šis funkcionalumas taip pat aprašomas šiame skyriuje.

3.1. Veiklų diagrama

Toliau pateikiama veiklos diagrama (1 paveikslėlis), vaizduoja visus naudotojui galimus atlikti veiksmus. Kadangi kol kas nėra planuojama programėlės suasmeninti ir kurti naudotojo prisijungimo funkcijos, visos 1 paveikslėlyje pavaizduotos veiklos yra prieinamos mobilaus Android įrenginio, kuriame suinstaliuota ši programėlė, naudotojui. Egzistuoja tik vienas naudotojo tipas. Pirmoji veikla, kurią mato naudotojas, yra langas, kuriame pavaizduota einamojo mėnesio išlaidų ataskaita skritulinės diagramos pavidalu. Taip pat matomas meniu pasirinkimo mygtukas. Spustelėjęs meniu mygtuką, naudotojas gali pasirinkti vieną iš keturių meniu veiklų. Pasirinkus išlaidų veiklą, naudotojui pateikiamas langas, kuriame išrikiuotos visos jo užregistruotos išlaidos, taip pat matomas mygtukas, leidžiantis pridėti naujas išlaidas. Pasirinkus pridėti išlaidas, susigeneruoja nauja veikla, kurioje naudotojas suveda informaciją, reikalingą patirtoms išlaidoms užregistruoti, ir suvedęs duomenis, naudotojas mygtuko paspaudimu registruoja naują įrašą. Tada naudotojas yra grąžinamas į išlaidų sąrašo langą. Šiame lange dar egzistuoja dvi galimos veiklos - kiekvieno individualaus įrašo peržiūra ir šio įrašo redagavimas arba ištrynimas.

1 pav. Veiklų diagrama



Likusios veiklos, kurias galima pasirinkti meniu juostoje yra pajamos, biudžetai ir taupymo planai. Kiekviena iš šių veiklų taip pat gali būti peržiūrėta kaip sąrašas, elementai gali būti redaguojami arba pašalinami. Visi galimi atlikti veiksmai pavaizduoti 1 paveikslėlyje.

3.2. Naudojimo atvejai

Kadangi programėlė nereikalauja naudotojo registracijos ar prisijungimo, egzistuoja tik vienas naudotojo tipas. Šiam naudotojui galimi veiksmai pavaizduoti 2 paveikslėlyje ir aprašyti toliau:

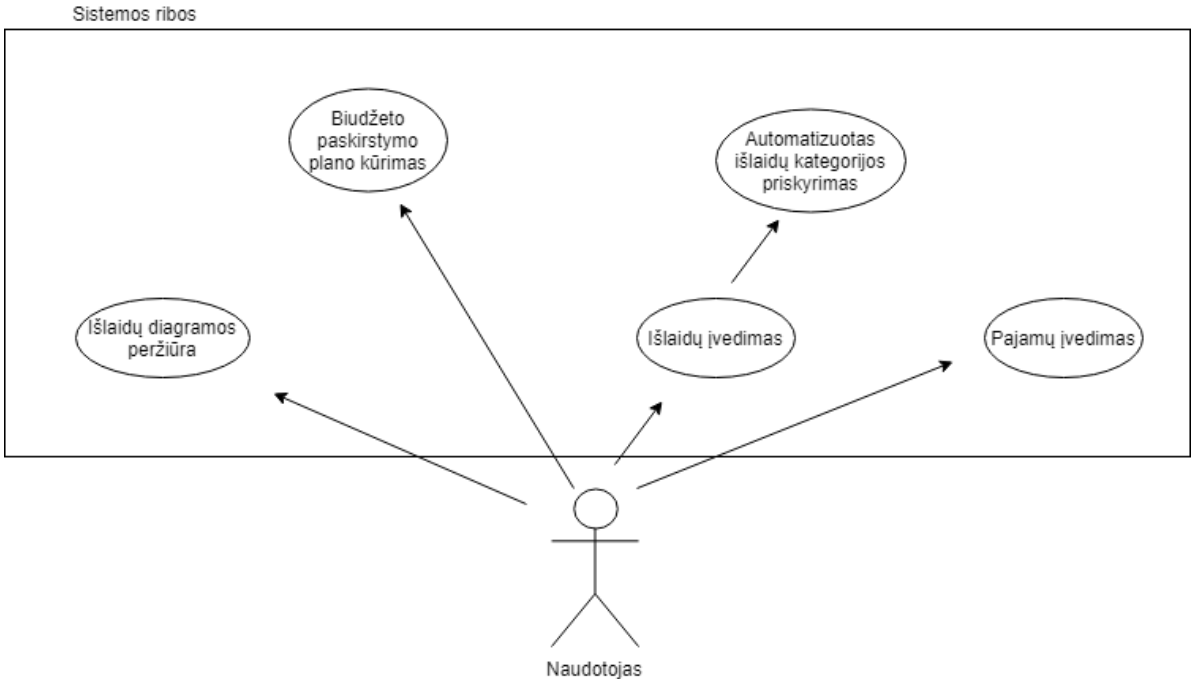
Išlaidų įvedimas – kiekvieną savo atliktą pirkinį naudotojas gali įvesti ir priskirti atitinkamai kategorijai (**Išlaidų priskyrimas kategorijai**). Egzistuoja alternatyvi funkcija – automatinis kategorijos priskyrimas naudojant čekio nuotrauką. Naudotojui pasirinkus nuotrauką iš mobilaus įrenginio laikmenos, programėlė, išanalizavusi ir atpažinusi čekyje pavaizduotą logotipą, nustato atitinkamą išlaidų kategoriją.

Pajamų įvedimas – naudotojas gali savo gautas pajamas užregistruoti, priskirdamas joms pavadinimą, pateikdamas jų dydį, aprašymą, bei kategoriją, pavyzdžiui, atlyginimas.

Biudžeto paskirstymo plano kūrimas – egzistuoja galimybė sukurti biudžetą atitinkamai išlaidų kategorijai. Naudotojui sukūrus biudžetą ir priskyrus jį kategorijai, matomas biudžeto likutis. Kiekvieną kartą naudotojui užregistravus išlaidas, biudžetas mažėja.

Išlaidų diagramos peržiūra – vaizduojama skritulinė diagrama, kurią sudaro kategorijų išlaidų sumos. Kiekviena dalis atspindi skirtingą kategoriją ir ši diagrama yra atnaujinama po kiekvieno naudotojo naujos išlaidos užregistravimo.

2 pav. Naudojimo atvejų diagrama



3.3. Duomenų bazės diagrama

Duomenų bazėje, kuri pavaizduota 3 paveikslėlyje, talpinama naudotojo įvesta išlaidų, pajamų, biudžetų ir taupymo planų informacija, bei šablonai, reikalingi naudotojo nufotografuoto čekio logotipo palyginimui ir kategorijos priskyrimui. Taip pat pavaizduoti sąryšiai tarp lentelių. Dauguma sąryšių yra vienas su daug, pavyzdžiui, pajamos gali turėti tik vieną priskirtą kategorija, tačiau atitinkama kategorija gali turėti daug priskirtų pajamų. Taip pat šis ryšys nėra privalomas, nes kategorija gali neturėti priskirtų pajamų.

Duomenų bazėje egzistuoja šešios lentelės:

Pajamos – pajamų lentelė, sauganti pajamų identifikavimo numerį, priskirtą pajamų šaltinio kategoriją, naudotojo pasirinktą pajamų pavadinimą, įvestą aprašymą ir pajamų dydį.

Išlaidos – išlaidų lentelė, sauganti išlaidų identifikavimo numerį, priskirtą išlaidų kategoriją, naudotojo pasirinktą išlaidų pavadinimą, aprašymą ir išlaidų dydį.

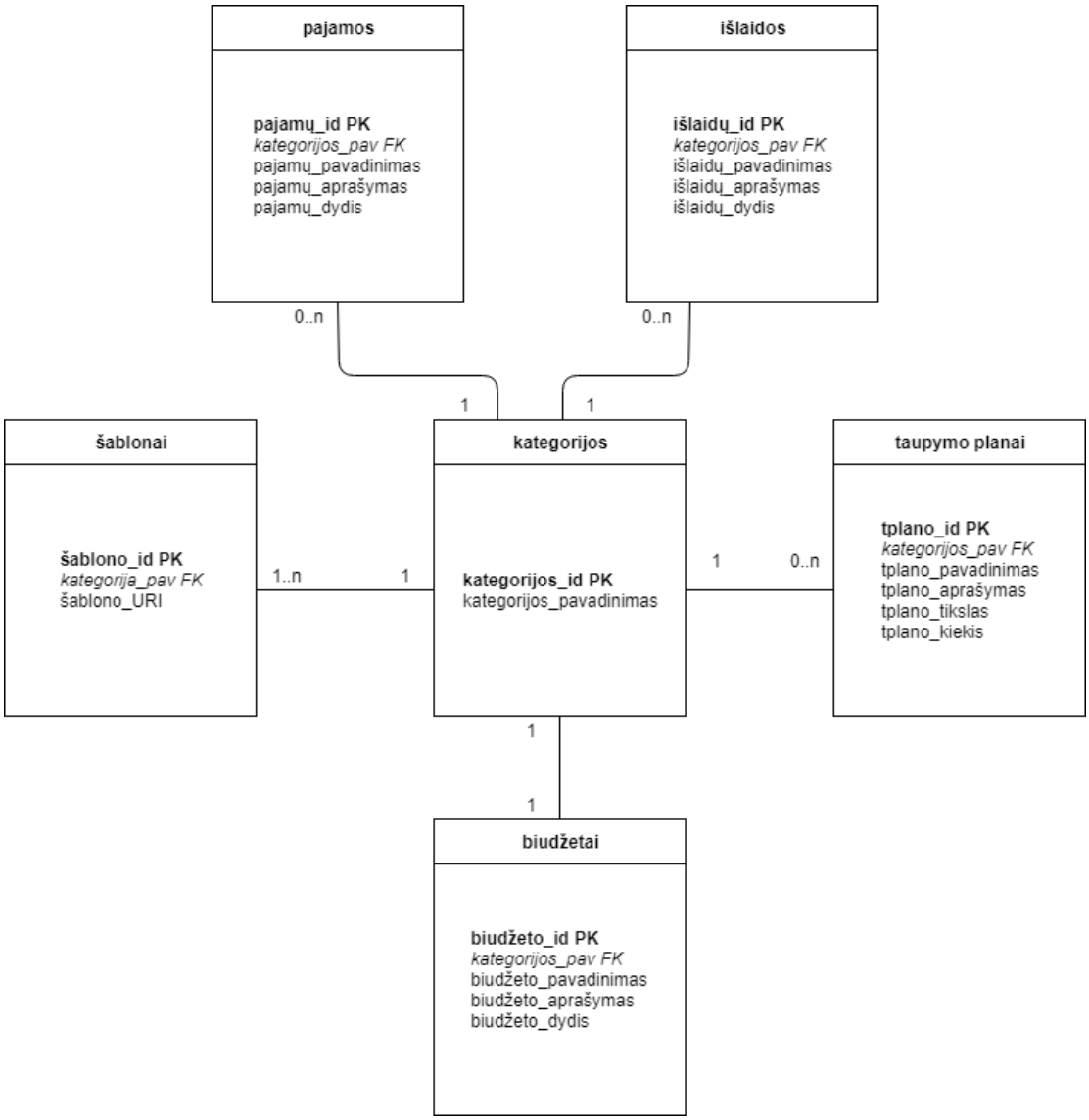
Biudžetai - biudžetų lentelėje laikoma informacija apie naudotojo sukurtus biudžeto paskirstymo planus. Joje laikomas biudžeto identifikavimo numeris, kategorija, kuriai paskirtas suplanuotas biudžetas, taip pat naudotojo paskirti pavadinimas ir aprašymas, bei biudžeto dydis.

Taupymo planai - taupymo planų lentelė, sauganti informaciją apie naudotojo sukurtus taupymo planus. Ši lentelė sudaryta iš taupymo plano identifikavimo numerio, taupymo plano pavadinimo, dydžio, naudotojo priskirto aprašymo, kategorijos ir dydžio, nurodančio naudotojo sutaupytų pinigų kiekį.

Kategorijos - kategorijų lentelė, talpinanti kategorijos identifikavimo numerį ir kategorijos pavadinimą.

Šablonai - šablonų lentelė, talpinanti šablono identifikavimo numerį, pavadinimą ir patalpinimo nuorodą.

3 pav. Duomenų bazės diagrama



4. Programėlės įgyvendinimas

Šiame skyriuje aprašomos programėlės įgyvendinimui pasirinktos technologijos, programėlės technologinis tipas ir aprašyti įgyvendinti sprendimai. Pateikiamas aplikacijos testavimo aprašymas ir pasirinkto metodo testavimo rezultatai.

4.1. Pasirinktos technologijos

Šiomis dienomis populiariausiomis mobiliųjų įrenginių operacinėmis sistemomis išlieka iOS ir Android [9]. iOS operacinė sistema, kuri veikia Apple įrenginiuose, turi savų įgyvendinimo pliusų ir specifikų, tačiau programėlės, kuriamos šiai operacinei sistemai, yra rašomos Objective-C arba Swift programavimo kalbomis naudojant Apple XCode IDE, kuri veikia tik ant Macintosh kompiuterių. Tai šiek tiek apsunkina darbo įgyvendinimo sąlygas, todėl buvo pasirinkta kurti Android programėlę. Android operacinės sistemos programėlės yra rašomos Java arba Kotlin programavimo kalbomis, o Eclipse arba Android Studio IDE gali būti paleidžiamos ant bet kokią operacinę sistemą turinčio kompiuterio. Dar viena iš priežasčių yra tai, kad Android yra pirma pagal popu-

liarumą mobilioji operacinė sistema [11]. Tai nurodo, kad naudotojų kiekis yra didelis ir ateityje būtų lengviau tokią programėlę pateikti rinkai.

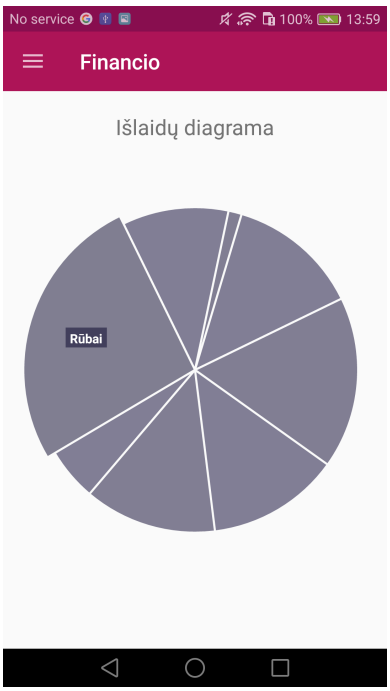
Šios programėlės kūrimas buvo įgyvendinamas Java programavimo kalba, naudojantis Android Studio aplinka. Ši aplinka yra oficiali Android programėlių aplinka ir suteikia patogią papildomų bibliotekų integravimo galimybę. Programėlėje naudojama OpenCV [8] kompiuterinės vizijos sprendimų biblioteka, kuri yra integruota į programėlės kodą, o jai naudoti turi būti įrašytas papildomas OpenCV Manager modulis, kuris įrašomas į įrenginį komandinės eilutės pagalba. Programėlės testavimo darbai buvo atliekami prijungus Android įrenginį USB jungtimi prie kompiuterio. Visiems duomenims, kuriuos įveda naudotojas, talpinti pasirinkta SQLite lokali duomenų bazė. Buvo pasirinktas vietinės (angl. native) programėlės tipas tam, kad reikalingi įrenginio komponentai būtų lengvai pasiekiami, naudotojui suteikus prieigą prie jų, bei užtikrinti programėlės veikimą be interneto ryšio.

4.2. Logotipo atpažinimo technologija

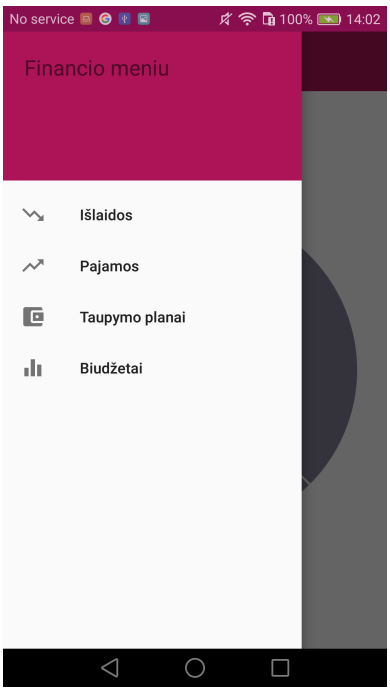
Kadangi buvo užsibrėžtas tikslas įgyvendinti logotipų atpažinimo funkcionalumą, buvo ieškoma būdų tam įgyvendinti. Išanalizavus visas atviro naudojimo bibliotekas, buvo pasirinkta viena populiariausių ir daugiausiai skirtingų funkcijų turinti kompiuterinės vizijos biblioteka - OpenCV [8]. Ši atviro naudojimo biblioteka talpina kelis šimtus kompiuterinės vizijos algoritmų ir gali būti panaudojama labai plačiai. Biblioteka yra sudaryta iš kelių skirtingų modulių, skirtų duomenų struktūrų kūrimui, paveikslėlių ir vaizdo įrašų apdorojimui, objektų ir veidų atpažinimui, bei daugybės kitų kompiuterinių sprendimų įgyvendinimui. Šioje programėlėje, iš prieš tai išvardintų pagalbinių modulių, bus reikalingi duomenų struktūrų moduliai paveikslėlių nuskaitymui, apdorojimui ir pateikimui, bei objektų atpažinimo modulis. Objekto atpažinimas vykdomas pasitelkus šablono atitikties techniką [12].

Šablono atitikties technika (angl. template matching) – tai technika, kurios tikslas yra pateiktame paveikslėlyje surasti atitinkančias vietas, kurios sutampa (arba yra panašios) su pateiktu šablono paveikslėliu. Šiai technikai reikalingi du pirminiai komponentai – pradinis paveikslėlis, kuriame tikimasi rasti šabloninį paveikslėlį, ir šabloninis paveikslėlis, kuris bus lyginamas su pirminiu paveikslėliu. Tam, kad identifikuoti sutampantį vaizdo plotą, šablonas yra lyginamas su pirminiu paveikslėliu slinkimo (angl. sliding) principu. Šio principo esmė yra slinkti šabloninį paveikslėlį per visą analizuojamą paveikslėlį, pradedant viršutiniame kairiajame kampe ir slenkant iš kairės į dešinę, bei lyginant viršutinį (šablono) ir apatinį (analizuojamo paveikslėlio) sluoksnius.

Atliekant šablono atitikties techniką, galima naudoti vieną iš 6 bibliotekos suteikiamų atitikties metodų. Šie metodai skiriasi atitikimo apskaičiavimo būdais ir rezultato pateikimu. Pavyzdžiui, metodai SQ_DIFF ir SQ_DIFF_NORMED, geriausiai šabloną atitinkančią paveikslėlio vietą žymi mažiausia reikšme, o likę keturi metodai – didžiausia. Atlikus šablono atitikties techniką, metodas grąžina rezultato masyvą, kuriame talpinamos kiekvieno pikselio atitikimo reikšmės. OpenCV biblioteka taip pat suteikia metodus šiam masyvui apdoroti. Egzistuoja funkcija minMaxLoc, kuri analizuoja rezultato masyvą, randa visus jame esančius ekstremumus ir jų pozicijas. Priklausomai nuo pasirinkto metodo, mažiausią arba didžiausią reikšmę turintys taškai yra labiausiai šabloną atitinkantys pikseliai.



4 pav. Pagrindinis programėlės langas



5 pav. Programėlės meniu pasirinkimų langas

4.3. Įgyvendintų sprendimų analizė

Pagrindinis programėlės langas, kuris rodomas paleidus programėlę, vaizduoja naudotojo užregistruotų išlaidų analizę pagal kategorijas. Pavaizduota skritulinė diagrama yra suskirstyta į skirtingas dalis pagal kategorijas, o pirštu palietus vieną iš dalių, pasirodo kategorijos pavadinimo etikėtė(4 paveikslėlis). Duomenys, naudojami diagramos braižymui, yra paimami iš lokalios SQLite duomenų bazės. Kiekvieną kartą naudotojui sukūrus naują išlaidą ir priskyrus ją kategorijai, arba pašalinus išlaidą iš sąrašo, šios kategorijos dalies dydis skritulinėje diagramoje pasikeičia atitinkamai.

Kairiajame viršutiniame lango kampe matomas meniu pasirinkimo mygtukas. Šis meniu nėra fiksuotas, jis matomas tik spustelėjus ant meniu ikonos arba paslinkus pirštu į dešinę. Meniu (5 paveikslėlis) matomi keturi skirtingų programėlės modulių pasirinkimai. Kiekvieno iš šių modulių dizaino principas yra panašus, kai kurie iš jų turi papildomų funkcijų, kurios aprašomos tolimesniuose poskyriuose.

4.3.1. Išlaidų modulis

Pirmasis iš meniu pasirinkimo modulių yra išlaidų modulis. Pasirinkus šį modulį, atidaromas visų naudotojo sukurtų išlaidų sąrašas (6 paveikslėlis). Šiame išlaidų sąraše vaizduojami du išlaidų elementai – išlaidų pavadinimas ir dydis. Visos išlaidos yra paimamos iš lokalios SQLite duomenų bazės ir atvaizduojamos sąraše. Naudotojui spustelėjus ant pasirinkto išlaidų elemento, atidaromas langas su pavaizduota pasirinktų išlaidų informacija (7 paveikslėlis). Šiame lange matoma išsamesnė informacija apie išlaidas, kaip pavyzdžiui, naudotojo įvestas išlaidų aprašymas ir pasirinkta, arba pagal išlaidų čekio paveikslėlį nustatyta išlaidų kategorija. Taip pat viršuje, pasirinkimų juostoje atsiranda pasirinkimas grįžti į prieš tai buvusį vaizdą, redaguoti pasirinktą išlaidą arba ją ištrinti. Naudotojui spustelėjus pasirinkimą redaguoti išlaidą, atidaromas langas, kuriame matomi laukai, kuriuos gali redaguoti naudotojas.

Atlikus išlaidų redagavimo veiksmus ir spustelėjus varnelę pasirinkimų juostoje viršuje, pa-



6 pav. Išlaidų sąrašas



7 pav. Išlaidų informacija

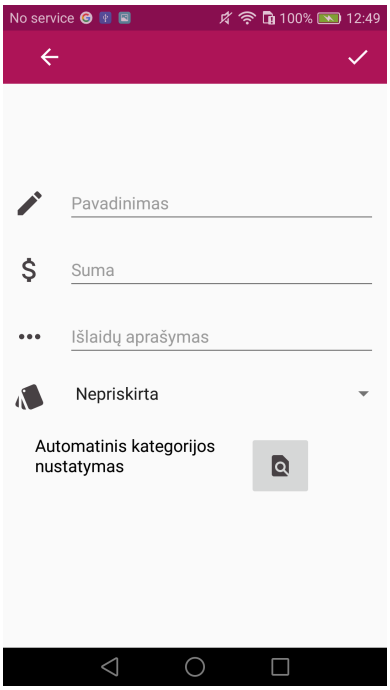
keitimai yra išsaugomi ir duomenų bazė yra atnaujinama naudojant metodą, pavaizduotą 1 išeities kode.Iš visų redaguotų arba likusių nepakeistų duomenų yra suformuojamas išlaidų objektas, kuris yra siunčiamas duomenų bazės pagalbinei klasei (angl. Database Helper) į ten esantį atnaujinimo metodą. Šis metodas išlaidų lentelėje atnaujina perduoto išlaidų objekto identifikavimo numerio įrašus.

1 išeities kodas. Išlaidų lentelės atnaujinimas

```
1 public boolean updateExpense(Expense expense, int id){
2         SQLiteDatabase db = this.getWritableDatabase();
3         ContentValues contentValues = new ContentValues();
4         contentValues.put(COL1, expense.getTitle());
5         contentValues.put(COL2, expense.getDescription());
6         contentValues.put(COL3, expense.getAmount());
7         contentValues.put(COL5, expense.getCategory_id());
8         int update = db.update(TABLE_NAME1, contentValues, COL0 + " = ?
              ", new String[] {String.valueOf(id)});
9         if(update != 1){
10            return false;
11            }
12        return true;
13    }
```

Išlaidų sąrašo lango (6 paveikslėlis) apatiniame dešiniajame kampe yra matomas pliuso ženklo mygtukas, kurį spustelėjus, naudotojui yra atidaromas išlaidų sukūrimo langas (8 paveikslėlis). Šiame lange naudotojas turi galimybę suvesti norimus duomenis, tokius kaip išlaidų pavadinimas, išlaidų dydis, aprašymas ir vienos iš kategorijų pasirinkimas. Naudotojui pageidaujant, galima pasirinkti išanalizuoti čekio paveikslėlį, pasirinktą iš įrenginio atminties, ir taip gauti automatiškai priskirtą išlaidų kategoriją pagal atpažintą čekio logotipą. Ši čekio paveikslėlio analizė plačiau aprašoma sekančiame poskyryje ir yra pasiekiama spustelėjus padidinamojo stiklo ikonos mygtuką lango apačioje.

8 pav. Išlaidų sukūrimo langas



4.3.2. Išlaidų čekio nuotraukos analizės metodas

Išlaidų registravimo metu, naudotojui pageidaujant, gali būti atlikta pasirinkto čekio paveikslėlio analizė. Ši funkcija bando atpažinti paveikslėlyje pavaizduoto čekio logotipą ir jeigu šis logotipas yra atpažintas, priskiria atitinkamą išlaidų kategoriją.

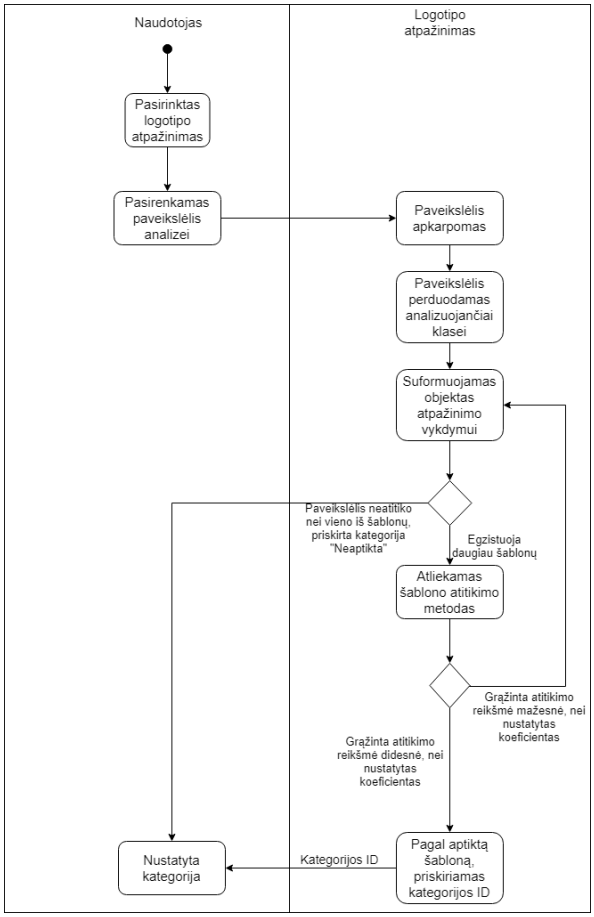
Programėlėje įgyvendintas logotipo atpažinimo metodas naudoja lokaliai patalpintus šabloninius logotipų paveikslėlius bmp formatu. Buvo pasirinkta logotipų šablonus laikyti lokalioje direktorijoje tam, kad jie būtų pasiekiami ir be interneto ryšio, o tai reiškia, kad visos programėlės funkcijos veikia nepriklausomai nuo interneto ryšio. Logotipo atpažinimui įgyvendinti ir išbandyti buvo naudojami 3 skirtingo dydžio logotipai, žymintys skirtingus paslaugų teikėjus.

Naudotojui pasirinkus išanalizuoti čekio nuotrauką ir automatiškai nustatyti kategoriją, jis yra nukreipiamas į langą, kuriame spustelėjus mygtuką, prašoma pasirinkti nuotrauką iš lokalios mobiliojo įrenginio atminties. Pasirinkus nuotrauką, ši nuotrauka yra nuskaitoma ir logotipų atpažinimo klasėje paverčiama į bitmap tipo objektą. Tam, kad pagreitinti logotipo atpažinimo veikimą, buvo nuspręsta analizei naudojamą čekio paveikslėlį apkirpti. Yra įprasta, kad visų pirkimo čekių logotipas dažniausiai būna pavaizduotas čekio viršuje, ir idealiu variantu, naudotojui nufotografavus čekį vertikaliu režimu, logotipas turėtų būti aptiktas viršutinėje čekio paveikslėlio dalyje. Priėjus tokią išvadą, buvo nuspręsta, kad 65% nuskaitytos nuotraukos turėtų būti nukerpama. Visi pakeitimai ir paveikslėlių transformacijos vyksta sukurtoje klasėje, kuri vėliau naudoja OpenCV bibliotekos suteikiamus metodus.

Naudotojo pateikta nuotrauka slinkimo (angl. sliding) metodu yra lyginama su pirmuoju šablonu. Testavimo metu, aprašytu 5 skyriuje, buvo įvestas šablono atitikties koeficientas. Jeigu metodo grąžinamas atitikties koeficientas yra žemesnis nei pasirinktas 0.36 koeficientas, yra atliekamas slinkimo metodas su sekančiu šablonu, ir šie veiksmai yra kartojami tol, kol yra randamas atitikmuo, arba kol nebeturima daugiau šablonų, su kuriais galima atlikti palyginimus. Jeigu nuotraukoje esantis logotipas atitinka bent vieną iš turimų šablonų, yra grąžinamas kategorijos identifikavimo numeris, kuris nustato atitinkančią kategoriją išlaidų kūrimo lange (8paveikslėlis). Jeigu nuotrau-

koje esantis logotipas neatitinka nei vieno iš šabloninių logotipų, yra nustatoma išlaidos kategorija „Nepriskirta", o tada naudotojas gali pats pasirinkti kategoriją iš sąrašo.

9 pav. Išlaidų čekio analizės veiksmų diagrama



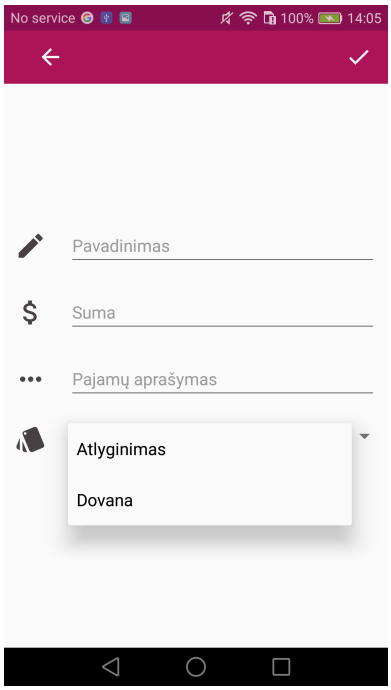
4.3.3. Pajamų modulis

Meniu pasirinkimo lange egzistuoja pasirinkimas "Pajamos". Tai modulis, kuriame naudotojas gali pamatyti visas savo užregistruotas pajamas, jas redaguoti, ištrinti arba pridėti naujas. Visa šio modulio struktūra yra labai panaši į prieš tai aprašytų išlaidų modulio struktūrą - yra vaizduojamas visų sukurtų pajamų sąrašas, matomas visų elementų pavadinimas ir pajamų dydis, galima pasirinkti ir atidaryti pasirinkta pajamų elementą. Pajamų sąrašo langas pavaizduotas 11 paveikslėlyje. Naudotojui spustelėjus ant pasirinkto sąrašo elemento, yra atvaizduojama informacija apie užregistruotas pajamas - naudotojo suteiktas pavadinimas, pajamų dydis, aprašymas ir kategorija, kuriai naudotojas priskyrė šias pajamas. Naudotojui pasirinkus sukurti naujas pajamas, yra atidaromas langas, kuriame prašoma įvesti pajamų duomenis. Esminis skirtumas tarp išlaidų ir pajamų registravimo yra tai, kad pajamų modulis neturi automatinio kategorijos nustatymo, bei iškrentančiame kategorijos pasirinkimų lange yra nurodytos kitos - pajamų kategorijos(10 paveikslėlis).

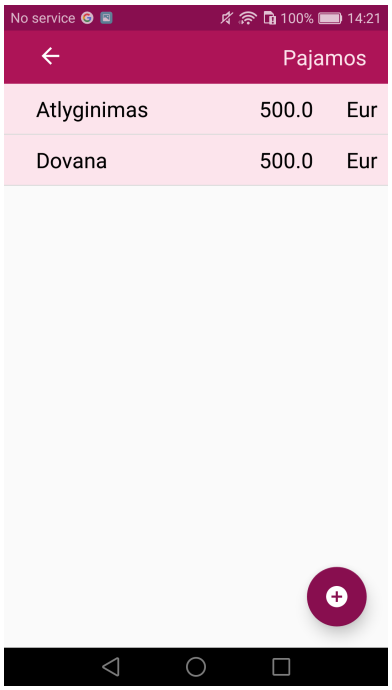
4.3.4. Biudžetų modulis

Trečiasis iš pateikiamų programėlės modulių yra biudžetų modulis. Tai modulis, kuriame naudotojas gali sukurti biudžetą jo pasirinktai kategorijai. Dizaino prasme, šis modulis mažai skiriasi nuo prieš tai aptartų modulių, tačiau turi vieną papildomą funkciją - biudžeto progreso peržiūros

10 pav. Pajamų sukūrimo langas

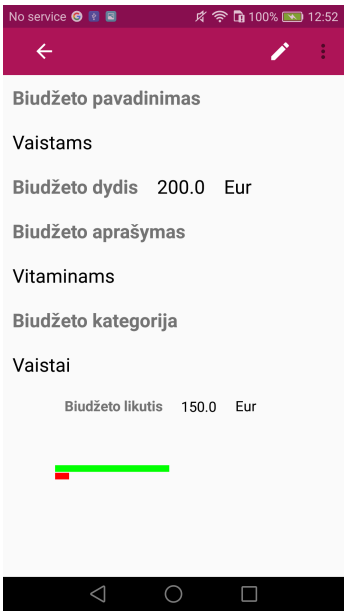


11 pav. Pajamų sąrašo langas



stulpelinę diagramą (12 paveikslėlis). Ši stulpelinė diagrama yra braižoma apskaičiuojant skirtumą tarp biudžeto ir visų, šio biudžeto kategorijai priskirtų išlaidų sumos. Žalias diagramos stulpelis vaizduoja koks buvo nustatytas biudžetas, o raudonos spalvos stulpelis vaizduoja kiek šio biudžeto jau yra išnaudota.

12 pav. Biudžeto informacija ir progreso diagrama

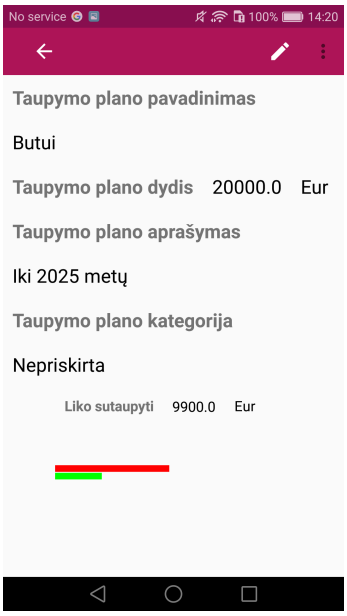


Taip pat rodomas ir likučio dydis eurais. Likučio reikšmė yra atvaizduojama apskaičiuojant skirtumą tarp naudotojo paskirto biudžeto konkrečiai kategorijai ir atliktų bei užregistruotų išlaidų, priskirtų šiai kategorijai. Duomenys, pavaizduoti šiai progreso diagramai, yra paimami iš lokalios SQLite duomenų bazės ir atvaizduojami atitinkamame struktūros elemente. Kiekvieną kartą naudotojui užregistravus išlaidas, ir priskyrus kategorijai, kuri turi sukurtą biudžetą, šio atitinkamo

biudžeto likutis mažėja. Taip naudotojas gali planuoti savo išlaidų paskirstymą ir stebėti, kaip jam sekasi laikytis šio plano.

4.3.5. Taupymo planų modulis

Paskutinis iš pateikiamų programėlės modulių yra taupymo planų modulis. Tai modulis, kuriame naudotojas gali sukurti savo taupymo planą. Šis modulis naudojamas kaip motyvacija skatinti naudotoją taupyti pasirinktam tikslui ir stebėti savo taupymo progresą. Meniu pasirinkimo juostoje pasirinkus taupymo planus, naudotojas yra nukreipiamas į taupymo planų sąrašo langą. Pasirinkus vieną iš sąrašo elementų, yra atveriamas taupymo plano langas (13 paveikslėlis). Šiame lange matoma visa naudotojo įvesta informacija, bei taupymo plano progreso diagrama. Ši diagrama kinta tada, kai naudotojas pasirenka redaguoti taupymo planą ir redagavimo lango apačioje esančiame langelyje įveda sumą, kurią jau paskyrė šiam taupymo fondui (14 paveikslėlis).



13 pav. Taupymo plano informacijos langas



14 pav. Taupymo plano redagavimas

5. Programėlės funkcionalumo testavimas

Programėlė buvo testuojama atlikus kiekvieną funkcionalumo ir dizaino pakeitimą. Taip buvo stengiamasi ištaisyti iškilusias problemas, išvengti klaidų tolimesniuose etapuose, bei užtikrinti nepertraukiamą programėlės veikimą. Testavimas buvo atliekamas Android Studio aplinkoje, programėlę paleidžiant prie kompiuterio USB jungtimi prijungtame Android įrenginyje. Šis įrenginys palaiko API 23 ir turi įdiegtą Android 6.0 operacinę sistemą.

Taip pat buvo atliekamas logotipo atpažinimo metodų testavimas. Buvo bandoma atlikti logotipo atpažinimą, naudojant dvidešimt čekių nuotraukų. Pirkimo čekiai buvo nufotografuoti iš skirtingo nuotolio, bei šiek tiek pakreipiant fotografavimo kampą. Taip buvo siekiama išbandyti, ar metodas sugebės aptikti šiek tiek iškreipto logotipo vaizdą.

2 lentelė. Šablonų atitikimo testavimas ir rezultatai

Šablonas	Paveikslėlis	Panašumo koeficientas
kavinė	identiškas	0.999
kavinė	panašus	0.371
kavinė	kito čekio	0.256
vaistinė	identiškas	0.943
vaistinė	panašus	0.910
vaistinė	kito čekio	0.271
kosmetikos parduotuvė	identiškas	0.999
kosmetikos parduotuvė	panašus	0.588
kosmetikos parduotuvė	kito čekio	0.232

Kaip ir nurodyta OpenCV bibliotekos dokumentacijoje [8], kiekvienas iš šešių galimų šablono atitikties metodų, grąžina skirtingus rezultatus. Įprastai kūrėjai, naudojantys šią kompiuterinės vizijos biblioteką, pasirenka vieną labiausiai lūkesčius atitinkantį metodą, taigi buvo atlikti bandymai su visais metodais ir buvo nuspręsta naudoti CCOEFF_NORMED metodą. Šis metodas gražina normalizuotą atitikimo reikšmę imtyje [0;1]. Testavimams buvo naudojami trys skirtingi logotipai, atitinkatys kategorijas "Kavinė", "Vaistinė" ir "Kosmetikos parduotuvė". Nuotraukos, kuriose pavaizduoti atitinkamų produktų tiekėjų pardavimo čekiai, buvo nuskaitomos iš įrenginio galerijos ir lyginamos su logotipų šablonais, laikomais programėlės lokalioje duomenų bazėje. Atlikus šiuos bandymus, buvo pasirinkta koeficiento riba 0.36, kuri naudojamas šablono atitikimo įvertinimui. Buvo įdiegta tokia logika - jeigu šablono atitikimo metodo grąžinama reikšmė yra didesnė arba lygi 0.36, naudotojo pateiktame paveikslėlyje ieškomas logotipas buvo aptiktas, o jeigu koeficiento reikšmė yra žemesnė, ieškomas logotipas nebuvo aptiktas.

2 lentelėje vaizduojami trys stulpeliai. Pirmojo stulpelio reikšmės yra kategorijų pavadinimai, kurias atitinka šablonai. Antrajame stulpelyje yra pateikiami trys naudojamo paveikslėlio tipai: identiškas, panašus ir kito čekio.

• Identišku laikomas paveikslėlis, kuriame egzistuoja visiškai identiškas šablonui logotipas, t.y. testavimo metu iš nuotraukų buvo iškirpti logotipai, naudojami bandymams ir metodo veikimo principui suprasti.

• Reikšmė "panašus" reiškia, kad panašioje nuotraukoje buvo ieškoma logotipo, kuris gali šiek tiek skirtis nuo šabloninio - spalva, pakreipimu, dydžiu, tačiau atitinka ieškomo tiekėjo logotipą.

• "Kito čekio" reikšmė reiškia, kad buvo tikrinama, kokį koeficientą grąžina metodas, kai žinoma, kad ieškomo logotipo paveikslėlyje nėra.

Trečiajame stulpelyje pateikia šablono ir pateikto paveikslėlio palyginimo rezultatai.

Taip pat buvo atliekami logotipo aptikimo efektyvumo bandymai. Darant pakeitimus programėlės kode, keičiant pateikiamo paveikslėlio dydį arba apkerpant paveikslėlį, logotipo radimo laikas buvo sumažintas nuo 30 sekundžių iki 6 sekundžių.

Išvados ir rekomendacijos

Šio darbo tikslas buvo sukurti programėlę, kuri padėtų naudotojui registruoti ir prižiūrėti savo asmeninius finansus. Buvo sukurta vietinio tipo programėlė, kuri suteikia galimybę užregistruoti naudotojo pajamas, bei priskirti jas kategorijoms, taip pat užregistruoti išlaidas ir pasinaudoti galimybe automatizuotai nustatyti išlaidų kategoriją pagal pirkimo čekio nuotrauką. Taip pat buvo įgyvendintas biudžeto sukūrimo funkcionalumas, taupymo planų kūrimas, bei vaizdinė išlaidų diagramos peržiūra. Buvo sėkmingai panaudota kompiuterinės vizijos biblioteka, kuri pateisino visus lūkesčius ir padėjo įgyvendinti norimą funkcionalumą.

Darbo atlikimo metu buvo susipažinta su OpenCV biblioteka, buvo įsigilinta į Android programėlių kūrimo specifikas, susipažinta su pagalbinėmis Android klasėmis, metodais ir funkcijomis.

Ateities tyrimų planas

Kiekvienas sukurtas technologinis produktas turėtų būti nuolat atnaujinamas ir tobulinamas. Turėtų būti atliekami testavimai, kurie parodytų, ar kiekviena funkcija veikia be trukdžių, ar nėra jokių spragų, kurios trukdytų naudotojui sklandžiai ir sėkmingai naudotis produktu.

Nors ir buvo įgyvendinti užsibrėžti tikslai, tačiau programėlės funkcionalumas ir našumas dar gali būti stipriai tobulinamas. Ateityje būtų galima įdiegti tokį papildomą funkcionalumą:

- Didesnė šablonų duomenų bazė, kuri išplėstų išlaidų kategorijų nustatymo galimybes;
- Patobulinta išlaidų čekio analizės klasė ir optimizuotas jos veikimas, pagreitintas logotipo atpažinimo laikas;
- Įgyvendinta priminimų apie taupymo planų ir biudžetų būseną sistema;
- Galimybė susikurti pirkinių sąrašą;
- Patobulintas programėlės dizainas;

Visi šie patobulinimai yra realiai įgyvendinami, nes sukurta programėlė yra vietinio tipo, o tokios programėlės tobulinimo galimybės yra labai plačios dėl neriboto įrenginio gamyklinių komponentų panaudojimo.

Literatūros šaltiniai

[1] Appbrain statistics. https://www.appbrain.com/stats/number-of-android-apps.

[2] Bluecoins finance. https://play.google.com/store/apps/details?id=com.rammigsoftware.bluecoins.

[3] Lietuvos bankų asociacija. https://www.lba.lt/lt/asociacijos-naujienos/2019/psichologe-finansinis-isprusimas-labai-susijes-su-emociniu-intelektu.

[4] Mano piniginė. https://play.google.com/store/apps/details?id=lt.algimka.manopiniginelt.

[5] Money advice service. https://www.moneyadviceservice.org.uk/en/articles/beginners-guide-to-managing-your-money.

[6] Money lover. https://play.google.com/store/apps/details?id=com.bookmark.money.

[7] Nngroup. https://www.nngroup.com/articles/mobile-native-apps/.

[8] Opencv biblioteka. https://opencv.org/.

[9] Os market share. http://gs.statcounter.com/os-market-share/mobile/worldwide.

[10] Spendee. https://play.google.com/store/apps/details?id=com.cleevio.spendee.

[11] Statcounter. http://gs.statcounter.com/os-market-share/mobile/worldwide.

[12] Template matching method. https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html.

[13] Wallet. https://play.google.com/store/apps/details?id=com.droid4you.application.wallet.