

ŠIAULIŲ UNIVERSITETAS
INFORMATIKOS KATEDRA

Matas Dudoravičius

Informatikos specialybės IV kurso nuolatinių studijų studentas

**Šiaulių apskrities Povilo Višinskio viešosios bibliotekos
mobiliosios aplikacijos atnaujinimas**

**Upgrading of Android Mobile Application for Šiauliai County
Povilas Višinskis Public Library**

BAKALAURO DARBAS

Darbo vadovas: doc. dr. V. Giedrimas

Recenzentas: doc. dr. Kęstutis Žilinskas

Šiauliai 2018 m.

„Tvirtinu, jog darbe pateikta medžiaga nėra plagijuota ir paruošta naudojant literatūros sąrašą pateiktus informacinius šaltinius bei savo tyrimų duomenis“

Darbo autorius _____

(vardas, pavardė, parašas)

Darbo tikslas – atnaujinti bibliotekos lankytojų skirtą mobiliąją „Android“ aplikaciją.

Uždaviniai:

1. Išanalizuoti bibliotekos lankytojų poreikius programai.
2. Ištirti ir palyginti kitose bibliotekose ar viešosiose įstaigose realizuotas panašias mobiliąsias aplikacijas.
3. Apžvelgti programinės įrangos kūrimo įrankius skirtus „Android“.
4. Palyginti ir pasirinkti geriausiai tinkančią „backend“ kūrimo ir valdymo platformą.
5. Parengti programos projektą ir jį realizuoti.

Darbo vadovas _____

(vardas, pavardė, parašas)

TURINYS

1	Įvadas	5
2	Analitinė dalis.....	6
2.1	Temos analizė.....	6
2.2	Darbo srities analizė.....	7
2.3	Darbinės srities modelis	8
2.4	Įrankių ir priemonių pasirinkimo analizė	9
3	Projektinė dalis.....	18
3.1	Projekto vykdymo planas	18
3.2	Pradinis projekto aprašymas	19
4	Darbo eigos aprašymas.....	23
4.1	Darbų eigos grafikas	23
4.2	Problemų ir jų sprendimai	23
4.3	Galutinė projekto būseną	24
4.4	Testavimas.....	24
4.5	Darbo rezultatų analizė	25
4.6	Plėtimo galimybės.....	25
5	Išvados.....	27
6	Literatūros ir informacinių šaltinių sąrašas.....	28
	Anotacija.....	29
	Summary.....	29
	Priedai	30

1 ĮVADAS

Šiandien kompiuterį vis dažniau gali pakeisti mobilusis įrenginys. Išmanieji telefonai ir planšetės savo galimybėmis vis labiau prilygsta mums įprastiems kompiuteriams. Todėl žmonės vis dažniau renkasi atlikti darbus elektroninėje erdvėje naudodamiesi patogesni, mažesni mobilių įrenginių. Išmanusis telefonas tampa pagrindine priemone naudotis internetu. Pasak „Kantar TNS“ rinkos tyrimų ir analizės duomenų, interneto vartotojų, naudojančių mobiliųjų įrenginių, kiekis sparčiai auga, o naudojančių įprastus stacionarius ar nešiojamus kompiuterius kiekis mažėja. Išmaniuoju telefonu prie interneto jungiamasi ne tik dažniau, bet juo naudojamasi vis daugiau laiko. Vidutiniškai darbo dienomis per parą naudodamiesi internetu išmaniajame telefone lietuviai praleidžia 54 minutes – 16 minučių daugiau nei 2016 m. pradžioje. [1]

Interneto naudojimas išmaniaisiais telefonais toliau turėtų tik augti. Tokius įrenginius turi vis daugiau Lietuvos gyventojų. Mobiliojo interneto ryšio infrastruktūra šalyje taip pat yra gerai išvystyta, o daugelis technologinių inovacijų šiandien yra orientuotos būtent į mobiliuosius įrenginius. Šias tendencijas svarbu turėti omenyje verslui teikiant paslaugas internetu, planuojant reklamos kampanijas ar tiesiog komunikuojant su klientais, kuriems išmanieji telefonai su interneto ryšiu suteikia galimybę realiu laiku dalintis savo komentarais ir įvertinimais, atsiliepimais apie įmonę ar įstaigą. Minėtos tendencijos taip pat aktualios ir kultūros įstaigoms siekiant užtikrinti ir išplėsti informacijos apie renginius bei kitų teikiamų paslaugų prieinamumą platesnei visuomenės daliai. „Kantar TNS“ atliktas tyrimas taip pat parodė, kad internetą išmaniuosiuose telefonuose lietuviai naudoja vis įvairesniems tikslams – ne tik naršymui socialiniuose tinkluose ir naujienų skaitymui, bet ir elektroninei bankininkystei, paslaugoms ar prekėms įsigyti. [1, 7]

Šiuo darbu siekiama atnaujinti Šiaulių apskrities Povilo Višinskio viešosios bibliotekos mobiliąją programėlę, kuri padėtų pritraukti jauną žmogų, bei šeimas, auginančias mažamečius vaikus, į biblioteką, supažindinti su teikiamomis paslaugomis, naujomis knygomis bei renginiais.

Darbo tikslas – atnaujinti bibliotekos lankytojams skirtą mobiliąją „Android“ aplikaciją.

Uždaviniai:

1. Išanalizuoti bibliotekos lankytojų poreikius programai.
2. Ištirti ir palyginti kitose bibliotekose ar viešose įstaigose realizuotas panašias mobiliąsias aplikacijas.
3. Apžvelgti programinės įrangos kūrimo įrankius skirtus „Android“.
4. Palyginti ir pasirinkti geriausiai tinkančią „backend“ kūrimo ir valdymo platformą.
5. Parengti programos projektą ir jį realizuoti.

2 ANALITINĖ DALIS

2.1 Temos analizė

Šio darbo metu atnaujinama bibliotekos programėlė mobiliems įrenginiams. Programa bus skirta tik „Android“ operacinę sistemą turintiems telefonams ir planšetiniams kompiuteriams.

„Android“ – tai populiariausia atviro kodo operacinė sistema, daugiausia naudojama išmaniuosiuose telefonuose. 2005 m. „Google“ korporacija įsigijo pirminį operacinės sistemos kūrėją, „Android Inc.“ Tai „Linux“ operacinės sistemos ir įvairių, daugiausia „Google“, sukurtų papildomų plėtinių kombinacija. Pagrindinė šios sistemos programavimo kalba yra „Java“, kurios sintaksė ir daugelis pagrindinių klasių nesiskiria nuo „Oracle“ siūlomos platformos, tačiau grafinės sąsajos API yra kitokia. [8, 9]

Priežastys, dėl kurių pasirinkta toliau dirbti su Android platforma:

1. Įrenginių įvairumas:

Su „iOS“ yra išleidžiama tik keliolika skirtingų įrenginių, pvz.: „iPhone“, „iPod“, „iPad“, „Apple TV“ ir kiti. O įrenginių su „Android“ yra sukuriami ženkiai daugiau. Galutinis tokių įrenginių skaičius nėra žinomas, nes be oficialiai leidžiamų išmaniųjų telefonų, laikrodžių, planšečių, televizorių ir automobilio multimedijos sistemų, kinai leidžia įvairias žinomų gamintojų originalinių įrenginių kopijas, į kurias diegia modifikuotas „Android“ versijas. Be to, „iOS“ įrenginiai yra prieinami tik labiau pasiturintiems vartotojams.

2. Gamintojų konkurencija:

Skirtingi mobiliųjų įrenginių gamintojai, tokie kaip „Samsung“, „LG“, „Sony“ ir kiti, naudojantys „Android“, konkuruoja ne vien su „Apple“, bet ir tarpusavyje. Didelė konkurencija skatina greitesnę progresą ir inovacijas.

3. Atvirasis kodas, laisva rinkos sistema:

„Android“ galima lengvai modifikuoti. Tai dar labiau didina šių įrenginių skverbtį. Be to, mikroprogramų kūrėjams nepalyginamai lengviau publikuoti savo darbus „Google Play“ parduotuvėje. [2]

2.2 Darbo srities analizė

Kauno apskrities viešoji biblioteka pirmoji Lietuvoje savo skaitytojams pateikė mobiliąją aplikaciją „KAVB“. Sukurta programėlė, kurioje pateikiama pagrindinė informacija apie bibliotekoje vykdomą veiklą. Žmonės patogiau ir greičiau gali gauti naujausią informaciją apie bibliotekoje vykstančius renginius, parodas bei kitus svarbius įvykius, pasinaudoti patogia prieiga prie mokymų medžiagos, el. katalogo, kontaktinės informacijos. Naudodamiesi minėta programėle, žmonės taip pat gali pateikti klausimus, dalyvauti įvairiose apklausose, dalytis informacija socialiniuose tinkluose. [5]

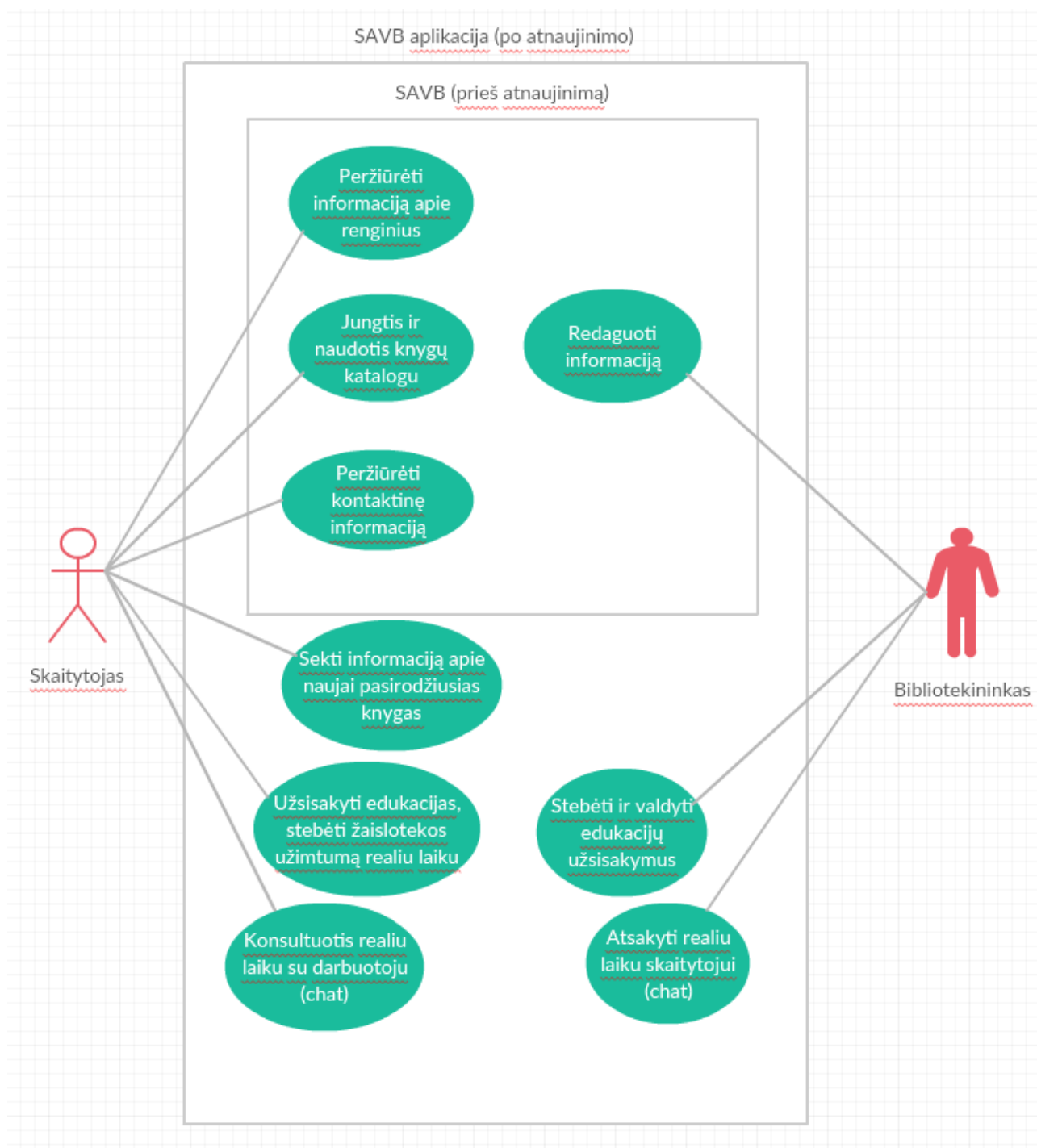
Mobiliąją aplikaciją skaitytojams sukūrė ir Trakų viešoji biblioteka. Programėlės paskirtis labai panaši į „KAVB“ programėlę – suteikti informaciją apie: naujas knygas; vykstančius renginius; bibliotekos veiklą ir teikiamas paslaugas; bibliotekos ir jos filialų kontaktinę informaciją; bibliotekos fondo knygas.

Visai kitokią aplikaciją „VilniusGO“ 2016 metais sukūrė Vilniaus miesto savivaldybės centrinė biblioteka. Tai programėlė, kurios tikslas – sudominti kuo platesnę auditoriją Vilniaus miesto istoriniu, gamtiniu ir kultūriniu palikimu. Programėlė neapsiriboja gerai žinomais, pvz. Senamiesčio, objektais, bet populiarina ir kitus ne mažiau svarbius, tačiau mažai žinomus objektus, pvz.: bunkerius, šiuolaikinio meno kūrinius ar gamtinį palikimą, esančius toliau nuo miesto centro.

Vilniaus pavyzdžiu pasekė ir Klaipėdos apskrities viešoji I. Simonaitytės biblioteka, išleisdama programėlę, skirtą skaitytojams, norintiems atrasti ir pažinti Klaipėdos krašto paveldą. [4]

Šiuo projektu kuriama aplikacija, daugeliu atžvilgių artimesnė Kauno ar Trakų bibliotekų programėlėms. Aplikacijoje taip pat bus publikuojama svarbiausia kontaktinė informacija, bibliotekos darbo laikas, naujienos, renginiai. Kuriama programa pasižymės tuo, jog joje bus publikuojamos specifinės bibliotekos paslaugos: kinas, mobilies bibliotekos laboratorijos užsiėmimai, supažindinimas su 3D grafikos modeliavimu ir spausdinimu. Aplikacijoje bus pateikiama informacija apie būsimas naujas bibliotekos erdves: sensorinį kambarį, skirtą autizmo spektro sutrikimą turintiems vaikams bei žaisloteką trumpalaikiai vaikų priežiūrai. Naudojantis aplikacija, žmonės galės registruotis į norimus užsiėmimus, matyti edukacinių renginių grafiką, sekti naujienas, teikti klausimus. Atsinaujinus informacijai, programėlės vartotojas bus informuojamas pranešimu.

2.3 Darbinės srities modelis



pav. 2-1 Use case diagrama

Vartotojų grupės: Skaitytojai, bibliotekininkai.

Reikalavimai:

- Vartotojui patogiau ir greičiau forma pateikiama visa kontaktinė informacija.
- Duomenys į programėlę perduodami iš nutolusios duomenų bazės, sinchronizuojama su bibliotekos tinklalapiu.

- Informuoti apie naujausius bibliotekos renginius ir knygas naudojami pranešimai (*Notifications*).
- Vartotojui suteikiama galimybė jungtis prie bibliotekos knygų katalogo, naudotis jo funkcijomis.
- Galimybė skaitytojui registruotis į bibliotekos siūlomas edukacijas.
- Realio laiko stebėti žaislotekos užimtumą, konsultuotis su darbuotoju.

2.4 Įrankių ir priemonių pasirinkimo analizė

Programavimo aplinka „Android Studio“

„Android“ programas gali kurti kiekvienas programuotojas, nes pagrindiniai įrankiai („Android Studio“) yra nemokami. Kūrimui nėra nebūtina investuoti į „Android“ pagrindu veikiančią mobilųjį telefoną ar kitą prietaisą, nes tam galima naudoti emuliatorių. Jei telefonas ar kitas prietaisas yra prijungiamas prie programuotojo kompiuterio per USB jungtį, pakeitus „Android“ sąsajoje keletą dokumentuotų nuostatų jis tampa programavimo įrankiu. Taip nesunku patikrinti programos darbą su GPS ir kitais telefono davikliais bei matyti, ar užduočiai pakanka telefono atminties, procesoriaus resursų. Naudojantis emuliatoriumi galima patikrinti, kaip programa veikia esant skirtingai ekrano raiškai ir įvairioms „Android“ versijoms.

Naujesnės „Android“ versijos leidžia programuoti ir C/C++. „Android“ C/C++ sąsaja beveik tokia pat kaip ir pasiūlyta „Sun Microsystems“ (JNI), kuri yra gana sudėtinga ir naudojama tik norint pasiekti iš „Java“ neprieinamus resursus, integruoti didelį turimą C/C++ kodo fragmentą ar optimizuoti daug procesoriaus resursų reikalaujantį algoritmą.

Programos „Android“ ilgą laiką buvo kuriamos su „Eclipse“, kuri ir dabar yra naudojama. Tačiau viena pagrindinių ir daugiausia galimybių turinčių programavimo aplinkų šiuo metu laikoma „Android Studio“. „Android“ programą galima parengti vykdymui ir be jokios programavimo aplinkos, naudojantis vien terminalo komandomis. Programos kompiliavimui komandine eilute ilgą laiką buvo naudojama „Ant“ kompiliavimo sistema, kurią pastaruoju metu pakeitė pranašesnis „Gradle“. [6-8]

Programavimo kalba „Java“

„Java“ – tai bendrosios paskirties, objektiškai orientuota programavimo kalba, kurią 1991 m. sukūrė „Sun Microsystems“ inžinierius James Gosling. Nuo 2010 m. „Java“ priklauso „Oracle“ korporacijai. Šiai programavimo kalbai jau daugiau nei 25 metai ir ji vis dar yra viena populiariausių programavimo kalbų. Ši programavimo kalba iki šiol vis dar atnaujinama ir tobulinama, kad geriausiai atitiktų programuotojų poreikius. Apie 63 proc. pasaulio programuotojų dirba ar yra dirbę su „Java“, kuri veikia septyniuose milijarduose įrenginių pasaulyje.

Nors „Java“ sintaksė yra panaši į C++, tačiau daugeliu atžvilgių „Java“ naudoti yra lengviau ir paprasčiau. Tai viena tinkamiausių kalbų pradedantiesiems – būtent ją dažniausiai ir pasirenkama mokytis pirmiausia

„Java“ kalbos populiarumas susijęs ir su jos universalumu: naudojantis „Java“ galima sukurti tiek mobiliąsias programėles, tiek ir didelius komercinius portalus. „Google“, „Android Inc.“ pasirinko „Java“ kalbą kuriant „Android“ operacinę sistemą. Nors „Android“, veikianti „Linux kernel“ pagrindu, parašyta C kalba, „Android“ SDK naudoja „Java“ kaip „Android“ aplikacijų pamatą, tačiau tik sintaksei, o ne klasių bibliotekoms. Todėl programėlės, skirtos šiai sistemai, taip pat dažniausiai kuriamos su „Java“. Per pastaruosius keletą metų smarkiai išaugo ir vis dar auga naudojimasis „Android“.

Svarbu paminėti, kad „Java“ turi vieną didžiausių ir stipriausių bendruomenių iš visų programavimo kalbų. Tokiuose daugybę narių turinčiuose forumuose kaip „StackOverflow“ galima greitai rasti visus atsakymus į iškilusius klausimus, patyrę nariai visada bus pasiruošę padėti naujokams savo programavimo žiniomis ir patirtimi. [13]

Duomenų bazė

Kuriant Šiaulių apskrities Povilo Višinskio viešosios bibliotekos programėlę svarbu, kad duomenys visiems programos vartotojams būtų pateikiami ir atnaujinami vienu metu, administratoriui iškart padarius pakeitimus duomenų bazėje. Todėl svarbu turėti duomenų bazę nutolusiame serveryje. Tarp nutolusios ir lokalsios duomenų bazių turi vykti reguliari duomenų sinchronizacija.

Pasirinktos duomenų bazių valdymo sistemos:

- Nutolusi duomenų bazė:

o „MySQL“ – viena iš reliacinių duomenų bazių valdymo sistemų, dirbanti „SQL“ kalbos pagrindu. „MySQL“ yra atviro kodo programinė įranga (GPL ir kitos licencijos), vystoma ir palaikoma švedų kompanijos „MySQL AB“. „MySQL“ RDBVS veikia daugelyje platformų ir yra dažnai pasirenkama programuojant internetines svetaines. Šiame sektoriuje su „MySQL“ konkuruoja su „PostgreSQL“. Pastaruoju metu „MySQL“ vis dažniau pritaikoma didelėse informacinėse sistemose, pvz.: „Vikipedija“, kurios apkrovimas kartais viršija 10 tūkstančių užklausų per sekundę, arba vienas iš didžiausių JAV kabelinės televizijos tinklų „Cox Communications“, kurio duomenų bazėje daugiau kaip 3600 lentelių. Nors prieigai prie MySQL duomenų bazių dažniausiai pasirenkama PHP kalba, kaip ir šiuo atveju, ją taip pat galima pasiekti įvairiomis kitomis programinėmis priemonėmis – C, C++, C#, „Java“, „Perl“, „Python“ ir kitomis. [3,12]

Backend platformos

„Firebase“

„Firebase“ – tai mobilių ir žiniatinklio aplikacijų kūrimo platforma, leidžianti kūrėjui rinktis iš daugybės įrankių ir servisų, padedančių kurti aukšto lygio programas.

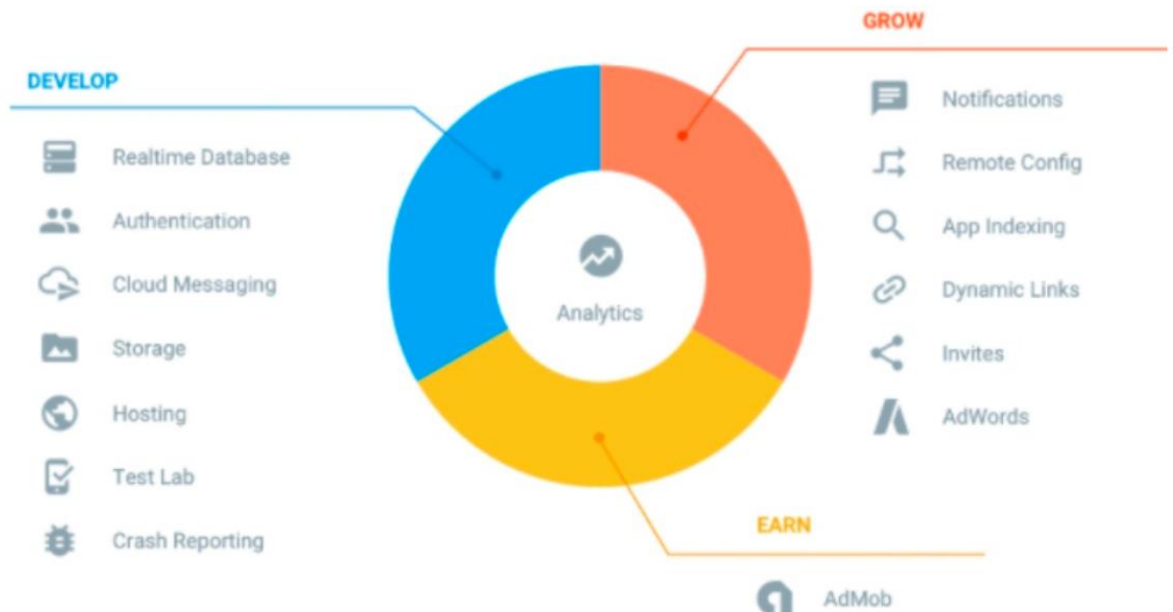
Istorija

„Firebase“ sukurtas 2011 metais pavadinimu „Envolve“. Šio įrankio pagrindinė paskirtis buvo integruoti susirašinėjimo funkciją tinklalapiuose. Šioje sistemoje siunčiami duomenys buvo ne vien susirašinėjimo žinutės, tačiau kūrėjai naudojo „Envolve“ sinchronizuoti duomenis tarp vartotojų realiu laiku, kaip žaidimo būseną. Tai paskatino kūrėjus James Tamplin and Adnrew Lee atskirti susirašinėjimo sistemą ir realaus laiko architektūrą. 2012 metais kaip atskira technologija buvo įkurta „Firebase“, skirta aplikacijų serverio dalies (*Backend*) serviso integravimui su realaus laiko funkcionalumu.

2014 metais „Google“ kompanijai įsigijus „Firebase“, ši sistema sparčiai išsivystė į milžinišką multifunkcinę platformą, kokia ji yra ir šiandien.

„Firebase“ paslaugos

„Firebase“ paslaugos gali būti skirstomos į dvi grupes:



pav. 2-2. Firebase paslaugos

1. Kurti ir testuoti aplikacijas:

- „Realtime Database“
- „Auth“
- „Test Lab“
- „Crashlytics“
- „Cloud Functions“
- „Firestore“
- „Cloud Storage“
- „Performance Monitoring“
- „Crash Reporting“
- „Hosting“.

2. Plėsti ir analizuoti vartotojų auditoriją:

- „Firebase Analytics“
- „Invites“
- „Cloud Messaging“
- „Predictions“
- „AdMob“
- „Dynamic Links“
- „Adwords“
- „Remote Config“
- „App Indexing“.

Realaus laiko duomenų bazė

„Firebase“ realaus laiko duomenų bazė – tai debesyse talpinama „NoSQL“ duomenų bazė, leidžianti talpinti ir sinchronizuoti duomenis tarp vartotojų realiu laiku.

Realaus laiko duomenų bazė (ang. *Realtime Database*) iš tiesų yra vienas didelis „JSON“ objektas, kurį kūrėjai gali redaguoti realiu laiku.



Realaus laiko mechanizmas padeda vartotojui lengviau pasiekti reikiamus duomenis iš bet kurio įrenginio bei susiekti su kitais vartotojais. Vienas iš privalumų naudojant šią sistemą – tai, jog nėra poreikio serveriui.

Kai vartotojas nėra prisijungęs, realaus laiko duomenų bazės SDK naudoja vidinę įrenginio talpyklą pokyčių saugojimui. Kai įrenginys prisijungia prie tinklo, vidiniai duomenys automatiškai sinchronizuojasi.

Ši duomenų bazė gali būti integruojama ir su „Firebase“ autentifikavimu.

Autentifikavimas

„Firebase“ autentifikavimas užtikrina greitą ir saugų vartotojų tapatybės nustatymą. Šiam procesui galima naudoti keletą skirtingų metodų:

- Email & Password
- Phone numbers
- Google
- Facebook
- Twitter

Šią autentifikavimo sistemą, sukurtą „Google Sign-in“, „Smart Lock“ ir „Chrome Password Manager“ kūrėjų, galima pritaikyti ir bibliotekos programėlės kūrimui.

„Firebase“ pranešimų siuntimas

„Firebase Cloud Messaging“ (FCM) užtikrina patikimą ir mažai resursų reikalaujantį susijungimą tarp serverio ir įrenginių siųsti bei gauti žinutes ir pranešimus „iOS“, „Android“ ir Web įrenginiuose.

Galima siųsti pranešimus pasirinktam įrenginių segmentui, kuris seka pasirinktą temą arba siųsti pasirinktam konkrečiam įrenginiui.

Pranešimai gali būti išsiunčiami iškart arba nustačius konkretų išsiuntimo laiką, atsižvelgiant į vartotojo laiko zoną.

„Firebase“ DB užklausos

Naudojantis specialiomis „Firebase“ DB užklausomis galima paprasčiau gauti duomenis iš duomenų bazės. Užklausos sukurtos susiejant vieną ar kelis filtrų metodus. Yra 4 rikiavimo funkcijos:

- orderByKey()
- orderByChild('child')
- orderByValue()
- orderByPriority().

Siekiant susiaurinti gaunamus duomenis, naudojamos šios papildomos funkcijos:

- startAt('value')
- endAt('value')
- equalTo('child_key')
- limitToFirst(10)
- limitToLast(10)

„Firebase“ kaip ir „SQL“ užklauskos susideda iš dviejų dalių, tačiau čia jos yra skirtingos. Pirmiausia sukuriamas nuoroda į tėvinį raktą ir tuomet naudojama rikiavimo funkcija. Papildomas funkcijas galima naudoti siekiant dar tikslesnių duomenų.

„Firebase“ testavimas

„Firebase“ testavimo laboratorija suteikia daugybę mobiliųjų bandymų įrenginių, padėsiančių išbandyti programas. Yra 3 testavimo būdai:

1. Instrumentinis testavimas – tai aplikacijos testavimo karkasas kaip „Espresso“ ir „UIAutomator 2.0“;
2. Robo testas – tai testas, kuriuo simuliuojamas vartotojo elgesys ir stebima, kaip veikia kiekvienas komponentas;
3. Žaidimo ciklo testas – testas, skirtas žaidimo programėlės testavimui. Šiuo testu simuliuojami galimi žaidėjo veiksmai.

Nuotolinė konfigūracija

Nuotolinis konfigūravimas leidžia siųsti atnaujinimus vartotojams. Naudojantis šia funkcija aplikacijoje galima padaryti bet kokią pakeitimą keičiant parametrus serverio pusėje nesisiunčiant naujos aplikacijos versijos.

Naudojantis nuotoline konfigūracija galima greitai ir lengvai atnaujinti aplikaciją nepublikuojant naujos versijos bei efektyviai stebėti vartotojų segmento elgesį į taikomus naujinimus.

„Firebase“ minusai

„Firebase“ palengvina ir pagreitina aplikacijos kūrimą, tačiau be daugybės privalumų ši platforma turi tam tikrų minusų.

Vienas svarbiausių neigiamų aspektų – aplikacijos pririšimas prie paslaugos tiekėjo. Programos duomenys tampa priklausomi nuo „Firebase“ serverio, jos nebegalima perkelti. Be to, nėra palaikymo tarp skirtingų sistemų, todėl pradėjus naudotis šiuo paslaugos tiekėju vartotojas negalės lengvai pereiti prie alternatyvios paslaugos, o tai ateityje gali sukelti tam tikrų nepatogumų:

- Nėra galimybės optimizuoti programos vidinės serverio dalies kodo pagal projekto poreikius.

- Prarandama galimybė įsigyti paslaugas iš kito tiekėjo.
- Prarandama galimybė naudotis naujausiomis technologijomis, kurias taiko kiti tiekėjai.
- Aplikacija visam laikui tampa priklausoma nuo „Google“ atnaujinimų.

„Firebase“ alternatyvos

„Firebase“ platforma turi atviro kodo alternatyvų, kurių funkcionalumas labai panašus. Šias platformas pasirinkus galima laisvai kontroliuoti savo duomenis, jie neliks prirakinti prie paslaugos tiekėjo.

„Parse Server“

Tai „Facebook“ sukurta panašias galimybes turinti platforma su dideliu ir itin svarbiu skirtumu – tai atviro kodo sistema. Viena pagrindinių savybių, skiriančių nuo „Firebase“ ir kitų panašių sistemų, tai, jog „Parse“ gali dirbti su „Express Web Application“ karkasu.

Privalumai:

- Leidžia lengvai modifikuoti aplikaciją, kad būtų galima vykdyti kitoje OS.
- Platus įrankių rinkinys paspartinti „backend“ kūrimą ir lengvai valdyti aplikaciją.
- Suteikia galimybę padaryti pakeitimus nerašant kodo, siųsti pranešimus ir stebėti analitinius duomenis.
- Gali būti talpinama beveik bet kokioje aplinkoje.

Minusai:

- Nėra komandinės eilutės sąsajos automatizuoti kodo kūrimą.
- Nėra automatinių operacijų.

Analitika

Pagrindinės analizės priemonės, įskaitant išsaugojimų metriką, labai naudingos „Parse Server“ funkcijos. Jomis naudojantis gaunama informacija analizei apie naudotojų sąveiką su aplikacija, susijusi su siunčiamais pranešimais. Tai taip pat gali būti modifikuojama siekiant gauti smulkių su „Parse“ susijusių duomenų. Jei to nepakanka, yra ir keletas mokamų papildomų analitikos priemonių.

Autentifikavimas

Naudojant „Parse server“ kuriamoje aplikacijoje galima integruoti prisijungimo galimybę naudojant „Facebook“, „Twitter“, „Google“, „Meetup“, „Github“, „LinkedIn“ ir „Instagram“.

Globali konfigūracija

Tai „Parse Server“ funkcija, atitinkanti „Firebase“ nuotolinę konfigūraciją. Taip galima keisti aplikaciją, jos veikimą be programos perinstaliavimo. Vartotojas pastebės aplikacijos pokyčius be jokių atnaujinimų.

Serverio pranešimai

„Parse Server“ turi pranešimų (*Push notification*) siuntimo galimybę, tačiau jos įgyvendinimas nėra itin paprastas. Valdymo panelėje nėra įrankių, kuriais vartotojas galėtų valdyti pranešimus. Siuntimas vyksta „PushAdapter“ pagalba, kuris leidžia jungtis prie bet kokios paslaugos serviso, pateikiančios siuntimo API.

Foniniai darbai

„Parse Server“ turi galimybę vykdyti programiniame kode aprašytas funkcijas reguliariai fone, kas tam tikrą laiką. Tam naudojamas užduočių atlikimo laikmatis. Galima koreguoti funkcijos vykdymo dažnumą, pasirinkti pradžios laiką. Matomi vydomų ir anksčiau atliktų darbų statusai. Kita vertus, yra limitai, nurodantys, kiek daugiausia foninių darbų gali būti vykdomų vienu metu ir kiek ilgai jie gali trukti. Tai veikia tik su mokama „Parse Server“ alternatyva.

Sistemos el. laišakai

„Parse Server“ siūlo lengvą būdą siųsti pasveikinimo, slaptažodžio atkūrimo, el. pašto patvirtinimo ir kitus sistemos generuojamus el. laiškus.

„Horizon“

„Horizon“ – tai realaus laiko atviro kodo „backend“ kūrimo platforma, skirta web ir mobiliosioms aplikacijoms, naudojama „JavaScript“ API. „Horizon“ sudaro trys pagrindiniai komponentai:

1. Tarpinis „backend“ serveris, naudojantis „Node.js“ ir „RethinkDB“, kuris palaiko duomenų saugumą, vartotojo autentifikavimą, įvesties tikrinimą, realaus laiko duomenų apsikeitimą ir leidimus.
2. „JavaScript“ kliento biblioteka, kurią gali naudoti „frontend“ kūrėjai JSON dokumento saugojimui duomenų bazėje, vykdyti užklausas ir sekti atnaujinimus.
3. Komandinė eilutė, leidžianti startuoti vidiniam „Horizon“ serveriui, kurti projekto šablonus ir diegti „Horizon“ aplikacijas debesyje.

Privalumai:

- „RethinkDB“ galima ne tik talpinti savo duomenis, bet ir vykdyti sudėtingas užklausas.
- Daugiaplatformėje aplinkoje galima diegti ir vykdyti realaus laiko web ir mobiliąsias aplikacijas.

- Atviro kodo sistema.

Minusai:

- Trūksta daug naudingų funkcijų, kurias teikia „Firebase“.
- Skirtingai nei „Firebase“, yra maža kūrėjų bendruomenė.
- „Horizon“ nėra toks stabilus, kaip „Firebase“.

Įvertinus visus privalumus ir trūkumus, bibliotekos aplikacijos „backend“ dalies kūrimui pasirinkta „Google Firebase“ platforma.

3 PROJEKTINĖ DALIS

3.1 Projekto vykdymo planas

Projekto vykdymo etapai:

1. Išmaniųjų telefonų naudojimo tyrimas.
2. Užsakovų reikalavimų išsiaiškinimas.
3. Panašių sistemų analizė.
4. Įrankių darbui pasirinkimas.
5. Sistemos maketų kūrimas.
6. Duomenų bazės kūrimas, pildymas duomenimis.
7. Vartotojo sąsajos kūrimas.
8. Aplikacijos loginės dalies programavimas.
9. Atskirų programos dalių apjungimas į visumą.
10. Testavimas.
11. Klaidų taisymas.

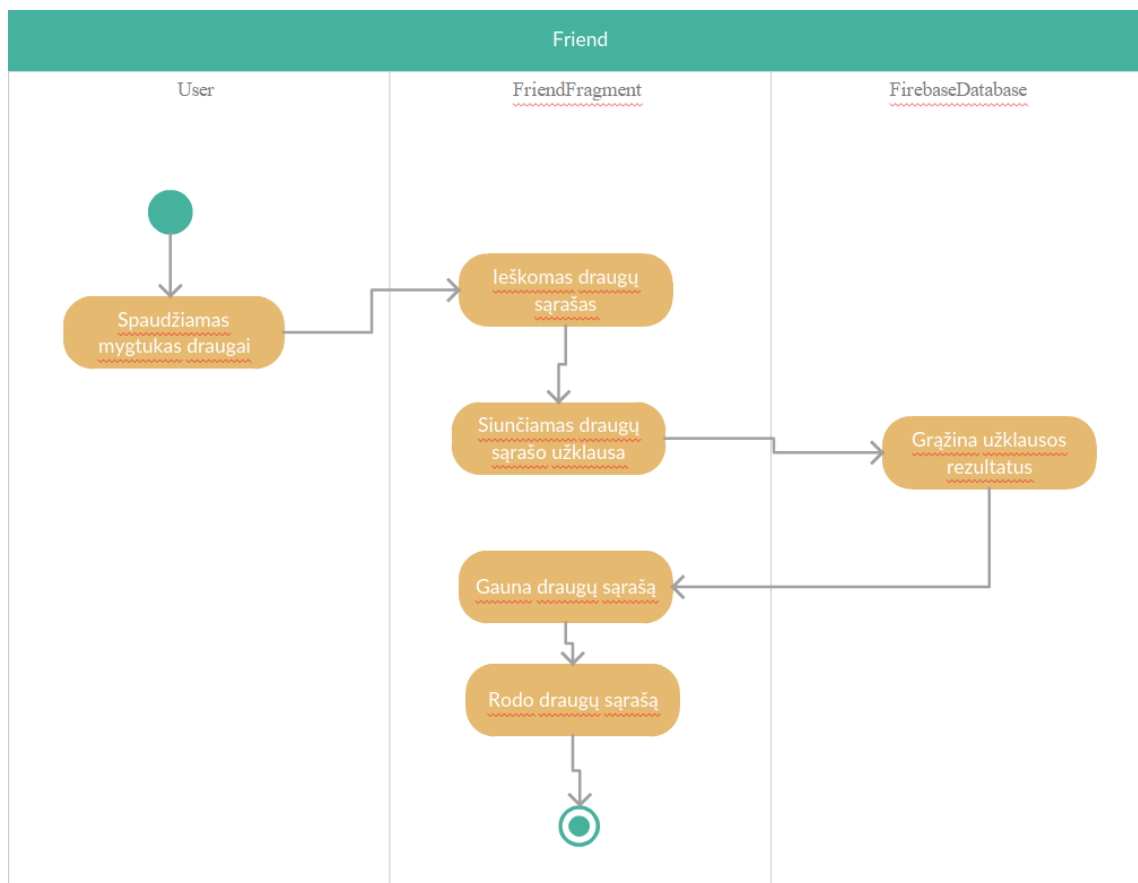
		Vasaris		Kovas				Balandis				Gegužė			
		Savaitės													
Nr.	Val. sk.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	10														
2	5														
3	10														
4	18														
5	32														
6	70														
7	45														
8	120														
9	10														
10	25														
11	25														
Iš viso:	370														

Lentelė 3-1 Projekto vykdymo planas.

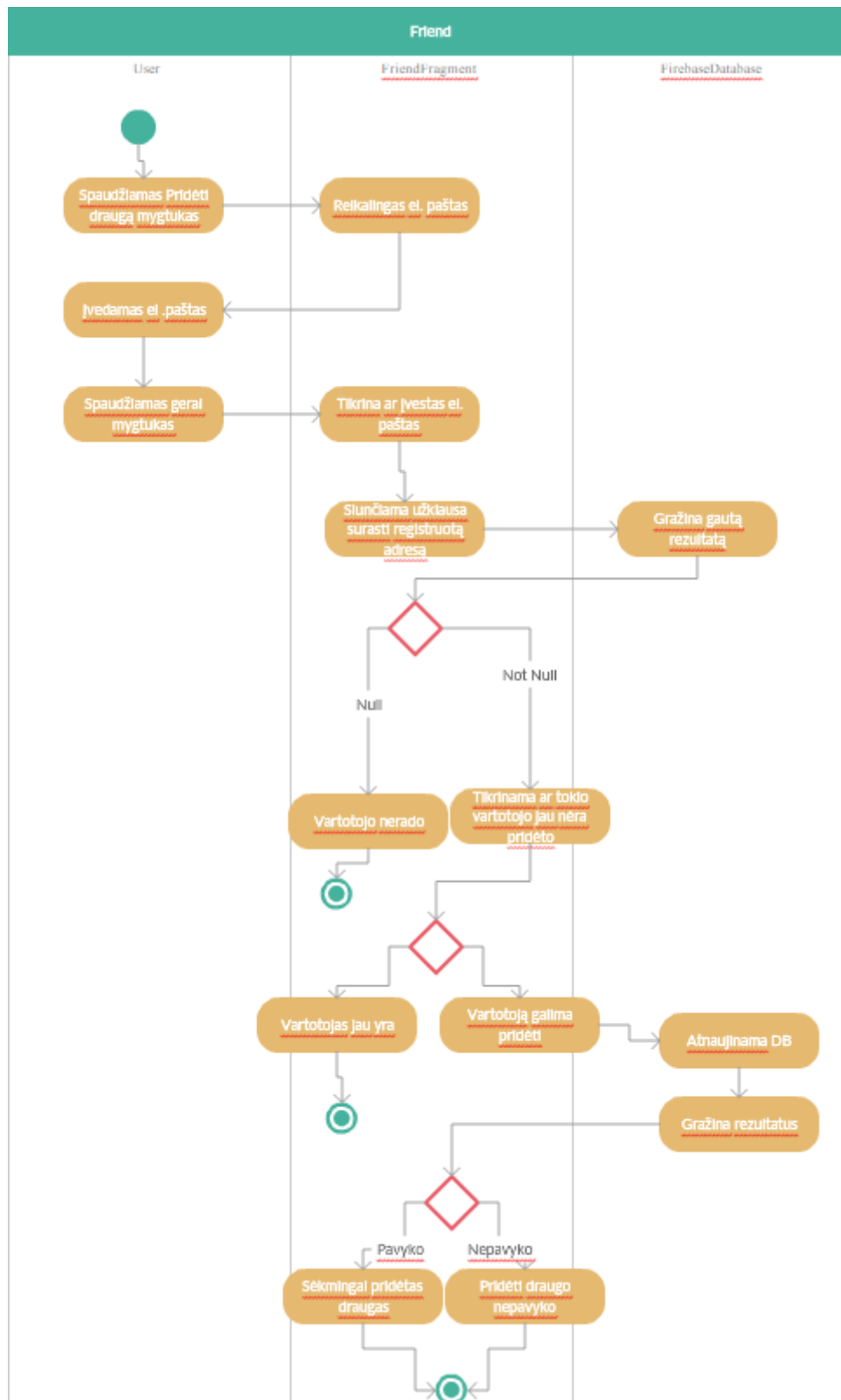
3.2 Pradinis projekto aprašymas

Vartotojo galimi veiksmai

Programėlė bus realizuota „Android“ operacinę sistemą naudojančių išmaniųjų telefonų platformoje. Remiantis išmaniųjų telefonų tyrimo rezultatais (žiūrėti Priedai. Pav. 7-1; 7-2), programėlės tiksliniai vartotojai – Šiaulių apskrities Povilo Višinskio viešosios bibliotekos jaunieji lankytojai, todėl aplikacijoje turi būti realizuoti užsakovo reikalavimai ypač atsižvelgiant į šią vartotojų grupę. Svarbus ne tik funkcionalumas, bet ir dizainas. Pagrindinis dizaino akcentas – bibliotekos kodinės, logotipe naudojamos žalia – balta spalvos. Dizainas turi būti paprastas, aiškiai suprantamas pirmą kartą atsidarius programą. Kita vertus, dizainas privalo išlikti nenuobodus, šiuolaikiškas. Vartotojas galės rasti informaciją apie bibliotekos naujienas, vykstančius renginius, parodas, prisijungti į knygų katalogą, kuriame galės susirasti norimą knygą ir ją užsisakyti. Programėlėje bus pateikta informacija apie bibliotekos siūlomas pramogas, ypač aktuali jaunimui ir ugdymo įstaigoms: kino salę, virtualios realybės akinius, LEGO robotus, 3D spausdintuvą, sensorinį kambarį, Žaisloteką. Programėlėje bus galima užsiregistruoti į edukacijas ar tiesiog susipažinti su šiuolaikinėmis kompiuterinės technikos inovacijomis. Taip pat bus pateikiami kontaktai bei bibliotekos darbo laikas. Svarbi skiltis – Žaisloteka. Jos skiltyje pateikiama informacija apie naują būsimą bibliotekos erdvę – trumpalaikį vaikų priežiūros centrą, kuriame kvalifikuotų darbuotojų priežiūroje palikti vaikai užsiims naujausiomis edukacijomis ir lavinamaisiais žaidimais. Šioje skiltyje bus pateiktas numatomų edukacijų sąrašas ir tvarkaraštis ir realizuotas susirašinėjimo kambarys, kur tėvai realiu laiku galės gauti norimą informaciją. Toliau pateikiamos veiklų diagramos, kuriose parodoma, kaip į pokalbio kambarį galima pridėti registruotą vartotoją.



pav. 3-1. Veiklų diagrama.



pav. 3-2. Veiklų diagrama.



4 DARBO EIGOS APRAŠYMAS

4.1 Darbų eigos grafikas

Darbų eiga sutampa su planuotų darbų grafiku. Visos projekto dalys užtruko tiek laiko, kaip ir buvo numatyta išankstiniame projekto vykdymo plane.

4.2 Problemos ir jų sprendimai

Naudojimuisi nepatogus grafinis dizaino kūrimo įrankis. „Android Studio“ leidžia kurti vartotojo sąsają tiesiog įtempiant objektus, mygtukus ir teksto laukelius, į programos kūrimo langą. Tačiau šia funkcija naudotis sudėtinga. Objektai nelygiuoja kaip siekiama sudėti, nušoka į kitą vietą. Problema išspręsta vartotojo išvaizdą aprašant xml kodu.

Sudėtingas nutolusios duomenų bazės pasiekiamumas. Lokaliai „Android“ naudojama „SQLite“ duomenų bazių valdymo sistema, tačiau „SQLite“ nėra skirta valdyti duomenis iš nutolusio serverio. Ji neturi jokių integruotų prisijungimo galimybių tinkle, todėl tenka naudotis „MySQL“. Prisijungimas prie „MySQL“ vykdomas per php failą tame pačiame serveryje. Php kode vykdomas pats prisijungimas ir užklausa, nurodanti, iš kokios lentelės kokius duomenis reikia nuskaityti. Gautus duomenis privaloma paversti į JSON („JavaScript Object Notation“) formatą. Šis perduoda duomenų objektus, sudarytus iš atributo ir reikšmės porų. JSON naudojamas apsikeisti duomenimis tarp skirtingų programų ir servisų, nepriklausomai nuo kalbos. Kiekviena kalba panaudoja savo vietinę funkciją, iškoduoja JSON tekstą ir grąžina masyvą savo kalboje. JSON pasirinktas, todėl, kad jis yra greitesnis ir lengvesnis, nei xml, kuriuo naudotis gana sudėtinga. „Android“ dalis yra dar kiek sudėtingesnė. Čia duomenų tarp mobiliojo įrenginio ir serverio perdavimui naudojami HttpPost ir HttpGet metodai. Atsakymas verčiamas į „String“ formatą. Išgaunami JSON duomenys, kuriuos jau galima laisvai naudoti. [11, 12]

Android -> AsyncTask class -> HttpPost -> PHP -> MySQL

Android <- AsyncTask <- HttpGet <- JSON <- PHP <- MySQL

pav. 4-1. Duomenų keitimosi tarp Android ir DB etapai.

Iš duomenų bazės atsisiųsti duomenys nepalaikė lietuviškų simbolių, nors duomenų bazės laukeliai turėjo nurodymą laikyti duomenis su utf-8 koduote. Unikodo simboliai buvo prarandami kituose duomenų išgavimo etapuose. Problema išspręsta nurodant šią koduotę php prisijungimo prie duomenų bazės faile.

4.3 Galutinė projekto būseną

Visos suplanuotos programos dalys veikia taip, kaip buvo numatyta.

Keitėsi vartotojo sąsajos pateikimas. Projektuojant buvo svarstoma daryti vartotojo prisijungimą visai sistemai, tačiau galutiniame projekto etape šios minties buvo atsisakyta, nes tai galimai lėtintų vartotojo sąveiką su programa, daliai vartotojų tai būtų nepriimtina ir jie galimai rečiau ar išvis nebesinaudotų šia aplikacija, todėl privalomas prisijungimas paliktas tik integruotoje katalogo sistemoje ir Žaislotekos pokalbių kambaryje bei siekiant identifikuoti skaitytoją prieš išduodant jam knygą ar kitą leidinį.

Keitėsi pačios aplikacijos nustatymų pateikimas vartotojui. Paliktas tik vienas nustatymas, leidžiantis įjungti arba išjungti naujienų pranešimus (*notifications*). Todėl atsisakyta kurti atskiros nustatymų skilties programos meniu.

Funkcionalumas išliko nepakitęs, nes visi užsakovo reikalavimai buvo įgyvendinti.

4.4 Testavimas

Testavimas vykdytas naudojantis „Android Studio“ emulatoriumi su keliais skirtingais virtualiais įrenginiais. Įrenginių sąrašas pateikiamas prieduose (žiūrėti Priedai. Pav. 7-3 „Android“ virtualių įrenginių sąrašas.). Testavimas taip pat vykdytas su fiziniu išmaniuoju telefonu „Huawei Honor 7“. Taip buvo išbandyta, kaip programa veikia, kai įrenginys turi skirtingus techninius parametrus: procesorių, darbinę atmintį, įvairių dydžių ekranus. Prieduose pateikiamas aplikacijos resursų vartojimo grafikas (žiūrėti Priedai. 7-4 Resursų naudojimo grafikas.) testuojant „Google Nexus 6P“ virtualiu įrenginiu.

Pagal gautus rezultatus aplikacijai tinkamai dirbti buvo reikalinga net iki 83% šio įrenginio procesoriaus, maksimaliai pasiekta 148MB darbinės atminties naudojimo ir iš tinklo parsisiunčiamų duomenų sekundinių šuolių iki 2MB. Programa tinkamai veikė visų galutinių testų metu. Atidarant katalogą ir kontaktus, kuriuose patalpintas „Google“ žemėlapis, juntamas nedidelis vėlinimas.

Testuojant vartotojo sąsają buvo leista aplikacija pasinaudoti IT patirties neturintiems asmenims. Jiems buvo nurodytos konkrečios užduotys, ką turi atlikti aplikacijoje. Remiantis šiuo testavimu buvo koreguojamas aplikacijos meniu. Pasiektas paprastas, nepainus programėlės valdymas, navigavimas standartiniam vartotojui. Vaizdas atrodo tvarkingai ir įskaitomai skirtingos rezoliucijos ekranuose.

Tam, kad būtų galima keistis duomenimis su bibliotekos serverio duomenų baze, reikia, kad telefonas būtų prisijungęs prie interneto tinklo. Jei interneto ryšio nėra, turinys bus užkrautas iš lokalios telefono duomenų bazės. Jei aplikacija paleidžiama pirmą kartą ar dėl kitokių aplinkybių

lokalio duomenų bazė yra tuščia, tuose languose, kur turi būti vaizduojami šie duomenys, jie bus tušti ir net atsiradus ryšiui jie neatsinaujins tol, kol nebus iš naujo paleista aplikacija.

4.5 Darbo rezultatų analizė

Šio projekto metu buvo tiriami Šiaulių apskrities Povilo Višinskio viešosios bibliotekos skaitytojų išmaniųjų telefonų naudojimo ypatumai: kokio amžiaus, kokia OS, kaip dažnai ir kokio pobūdžio programėlėmis dažniausiai naudojasi. Tyrimo rezultatai projektą pakreipė „Android“ aplikacijos naujinimo linkme. Tam buvo analizuojama, lyginama pati „Android“ operacinė sistema su kitomis: „iOS“, bei „Windows“. Analizuojamos panašios kitų bibliotekų jau sukurtos sistemos. Atsižvelgiant į užsakovo reikalavimus bei asmeninę patirtį, palyginti ir pasirinkti projektavimo ir programavimo įrankiai. Projektavimo etape diagramoms kurti buvo naudojamas „Creately“ įrankis, vartotojo sąsajos maketų projektavimui – „Marvelapp“. Išanalizavus galimas programavimo aplinkas ir įrankius, buvo pasirinkti šiuo metu laikomi geriausi: aplinka – „Android Studio“, programavimo kalba „Java“, o duomenų bazės valdymo sistema – „MySQL“. Sukurtos duomenų bazės lentelės susietos ryšiais. Visų šių įrankių pagalba buvo sukurta mobilioji programėlė, orientuota į bibliotekos skaitytoją. Į skaitytojo išmanųjį iš nutolusiame serveryje esančios duomenų bazės atkeliauja visa aktualiausia informacija, kuri pateikiama vartotojui itin patogiai ir patrauklia forma.

4.6 Plėtimo galimybės

Šis projektas turi labai dideles plėtimosi galimybes. Programoje ateityje galėtų atsirasti daugiau funkcijų, skirtų ne tik skaitytojams, bet ir bibliotekos darbuotojams:

- Galima integruoti apklausų sistemą, kurioje kiekvieną savaitę ar mėnesį veiktų naujos apklausos, į kurias atsižvelgdama biblioteka galėtų priimti tam tikrus sprendimus tiesiogiai susijusius su skaitytojais.
- Virtualus bibliotekos žemėlapis. Biblioteka pakankamai didelė ir rečiau apsilankančiam žmogui gali pasirodyti klaidi, todėl bibliotekos navigacija būtų itin naudinga ieškant skirtingų skyrių, salių ar kitų erdvių.
- Naujai perdarytas knygų katalogas.
- Virtualus skaitytojo pažymėjimas. Einant pasiimti knygos žmogui nebereikėtų kartu nešti skaitytojo pažymėjimo, nes jo telefono ekrane būtų vaizduojamas brūkšninis kodas ir skaitytojo pažymėjimo numeris, kurį bibliotekininkas galėtų nuskenuoti.
- Galima sukurti ir sistemoje integruoti bibliotekos žaidimą. Šis pritrauktų daugiau vaikų ir jaunimo naudotis aplikacija, o net ir žaidimui įsijungus aplikaciją jaunuolius pirmiausia pasiektą informacija apie renginius ir kt.

- Integruoti tik darbuotojams pagal prisijungimą atsirandančią skiltį su visų skyrių darbuotojų telefono numeriais ir pašto adresais, skambinimo per aplikaciją internetu funkcija bei skiltimi su bibliotekoje vykstančiais planiniais darbais, ar informacija apie masinius gedimus.
- Programą sukurti ir „iOS“ platformai.

5 IŠVADOS

1. Atlikus išmaniųjų telefonų naudojimo tyrimą, nustatyta, kad populiariausia išmaniųjų įrenginių operacinė sistema yra „Android“, todėl kurti šiai platformai aplikaciją yra naudingiausia.
2. Atlikus panašių sistemų ir užsakovo reikalavimų analizę, nustatyta, kad sistemoje reikia realizuoti šias funkcijas: naujų knygų ir renginių sinchronizaciją su tinklalapiu, siūlomų edukacijų viešinimą ir žaislotekos susirašinėjimo langą.
3. Atlikus lyginamąją analizę tarp „backend“ kūrimo ir valdymo platformų, prieita prie išvados, kad „Firebase“ įrankis yra patikimiausias ir greičiausias realizuojant aplikacijai reikalingas paslaugas: autentifikavimą, realaus laiko duomenų bazę bei pranešimų tarp serverio ir kliento aplikacijos siuntimą.
4. Atlikus testavimą su skirtingų techninių parametrų mobiliais įrenginiais nustatyta, kad programėlė sklandžiai ir greitai veikia visuose „Android“ įrenginiuose, naudojančiuose ne žemesnę nei API (aplikacijų programavimo sąsają) 23 arba „Android 6.0.“. Tvarkingam programos duomenų pateikimui įtakos nedaro skirtingas įrenginio ekrano dydis.

6 LITERATŪROS IR INFORMACINIŲ ŠALTINIŲ SĄRAŠAS

1. Kantar TNS. Prie interneto lietuviai dažniausiai jungiasi išmaniaisiais telefonais. Prieiga per internetą: <<http://www.tns.lt/lt/news/kantar-tns-prie-interneto-lietuviai-dazniausiai-jungiasi-ismaniaisiais-telefonais>> [žiūrėta 2018-02-11]
2. Globe News 24. 5 priežastys kodėl Android laimi prieš iOS. Prieiga per internetą: <<http://globenews24.com/LT/news,5-priezastys-kodel-android-laimi-pries-ios>> [žiūrėta 2017-12-16]
3. Zigurd Mednieks, Laird Dornin, G. Blake Meike, and Masumi Nakamura. Programming Android. O'Reilly Media, Inc. 2011.
4. Vilniaus miesto savivaldybės centrinė biblioteka. Mobilioji aplikacija VilniusGO. Prieiga per internetą: <<http://www.vcb.lt/mobilioji-aplikacija-vilniusgo/>> [žiūrėta 2018-03-20]
5. Kauno apskrities viešoji biblioteka. Kauno apskrities viešosios bibliotekos mobiliosios programėlės. Prieiga per internetą: <<https://kvb.lt/lt/straipsniai/4524-kauno-apskrities-viesosios-bibliotekos-mobiliosios-aplikacijos>> [žiūrėta 2018-02-26]
6. Android Community. Introduction to Android. Prieiga per internetą: <<https://developer.android.com/guide/index.html>> [žiūrėta 2018-04-04]
7. Richard Wordsworth. Android Market reaches 500,000 app mark. Prieiga per internetą: <<http://www.t3.com/news/android-market-reaches-500000-app-mark>> [žiūrėta 2018-01-30]
8. Open handset alliance. Android overview. Prieiga per internetą: <http://www.openhandsetalliance.com/android_overview.html> [žiūrėta 2018-04-15]
9. Wei-Meng Lee. Android 4 Application Development. John Wiley & Sons, Inc, Indianapolis, Indiana, 2012.
10. GeekyAnts. Introduction to Firebase. Prieiga per internetą: <<https://hackernoon.com/introduction-to-firebase-218a23186cd7>> [žiūrėta 2018-04-27]
11. Tutorials Point Community. Android - SQLite Database. Prieiga per internetą: <https://www.tutorialspoint.com/android/android_sqlite_database.htm> [žiūrėta 2018-05-02]
12. Gabor. Connecting to MySQL database. Prieiga per internetą: <<http://www.helloandroid.com/tutorials/connecting-mysql-database>> [žiūrėta 2018-03-14]
13. Vilnius Coding Scholl. Kodėl „JAVA“ yra viena populiariausių programavimo kalbų. Prieiga per internetą: <<http://www.vilniuscoding.lt/kodel-„Java“-yra-viena-populiariausiu-programavimo-kalbu/>> [žiūrėta 2018-04-22]

ANOTACIJA

Autorius: Matas Dudoravičius

Tema: Šiaulių apskrities Povilo Višinskio viešosios bibliotekos mobiliosios aplikacijos atnaujinimas.

Šio darbo pagrindinis tikslas – atnaujinti Šiaulių apskrities Povilo Višinskio viešosios bibliotekos mobiliąją aplikaciją, skirtą jos lankytojams.

Atliktas išmaniųjų telefonų naudojimo tyrimas. Analizuota populiariausia išmaniųjų įrenginių OS „Android“. Apibrėžti programai keliami reikalavimai, palygintos jau sukurtos kitų bibliotekų aplikacijos. Pasirinkti projektavimo, programavimo įrankiai bei aplinka, sukurta programėlė aptarnaujanti nutolusi „MySQL“ duomenų bazė. Aplikacijos „backend“ kūrimui pasirinkta „Google Firebase“. Naudojantis šiais įrankiais sukurta galutinė, skirtingais įrenginiais ištestuota aplikacijos versija.

SUMMARY

Author: Matas Dudoravičius

Subject: Upgrading of Android Mobile Application for Šiauliai County Povilas Višinskis Public Library.

The main subject of this work is to upgrade the “Android” application for Šiauliai County Povilas Višinskis Public Library, created for its visitor.

On the start of this project smartphone usage research was done. Analyzed the most popular smart device OS – Android. Defined requests for app, compared applications of the other libraries which were created before. Suitable designing, programming ideas and tools were selected. MySQL database was made for serving application remotely. Created a cloud backend for app using Firebase. With these instruments the final app version was developed and tested with different devices.

PRIEDAI

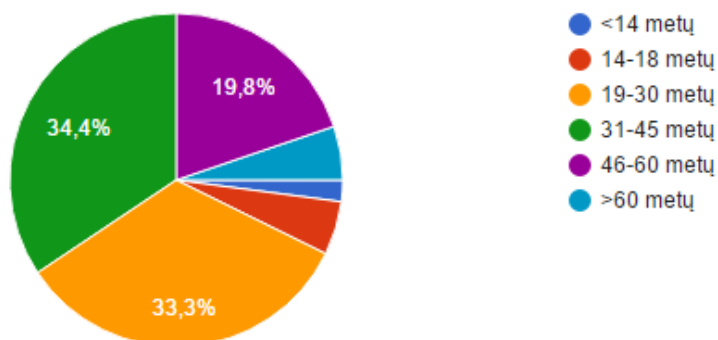
Bibliotekos lankytojų išmaniųjų telefonų naudojimo tyrimas

<https://goo.gl/forms/b644oL4OEs7RjMqm1>

Rezultatai:

Amžius

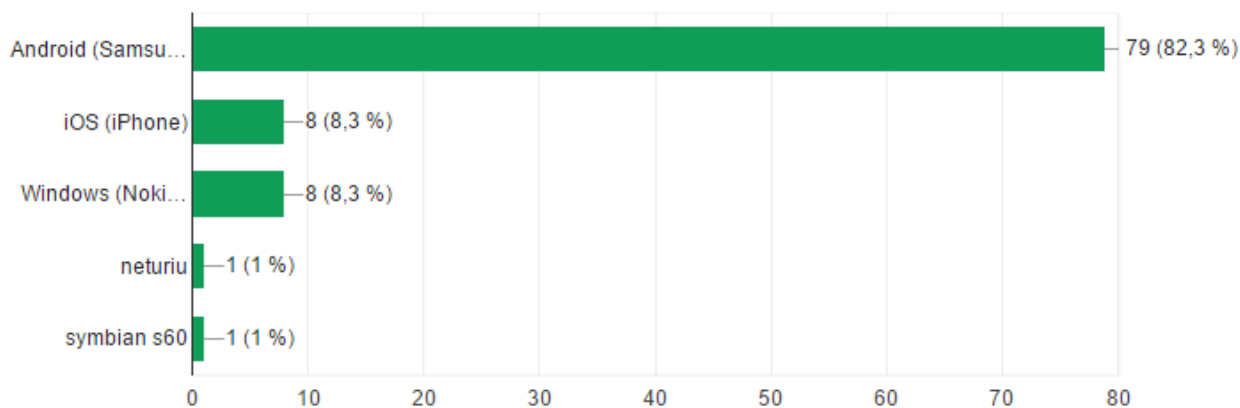
96 atsakymai



pav. 0-1. Apklausos dalyvių pasiskirstymas amžiaus grupėse.

Su kokia OS yra Jūsų išmanusis? (Jei turite kelis skirtingus, pažymėkite)

96 atsakymai



pav. 0-2. Android populiariausia OS tarp apklausos dalyvių.

Išmaniaisiais įrenginiais ir jų programėlėmis daugiausia naudojasi jauni, pilnamečiai asmenys.

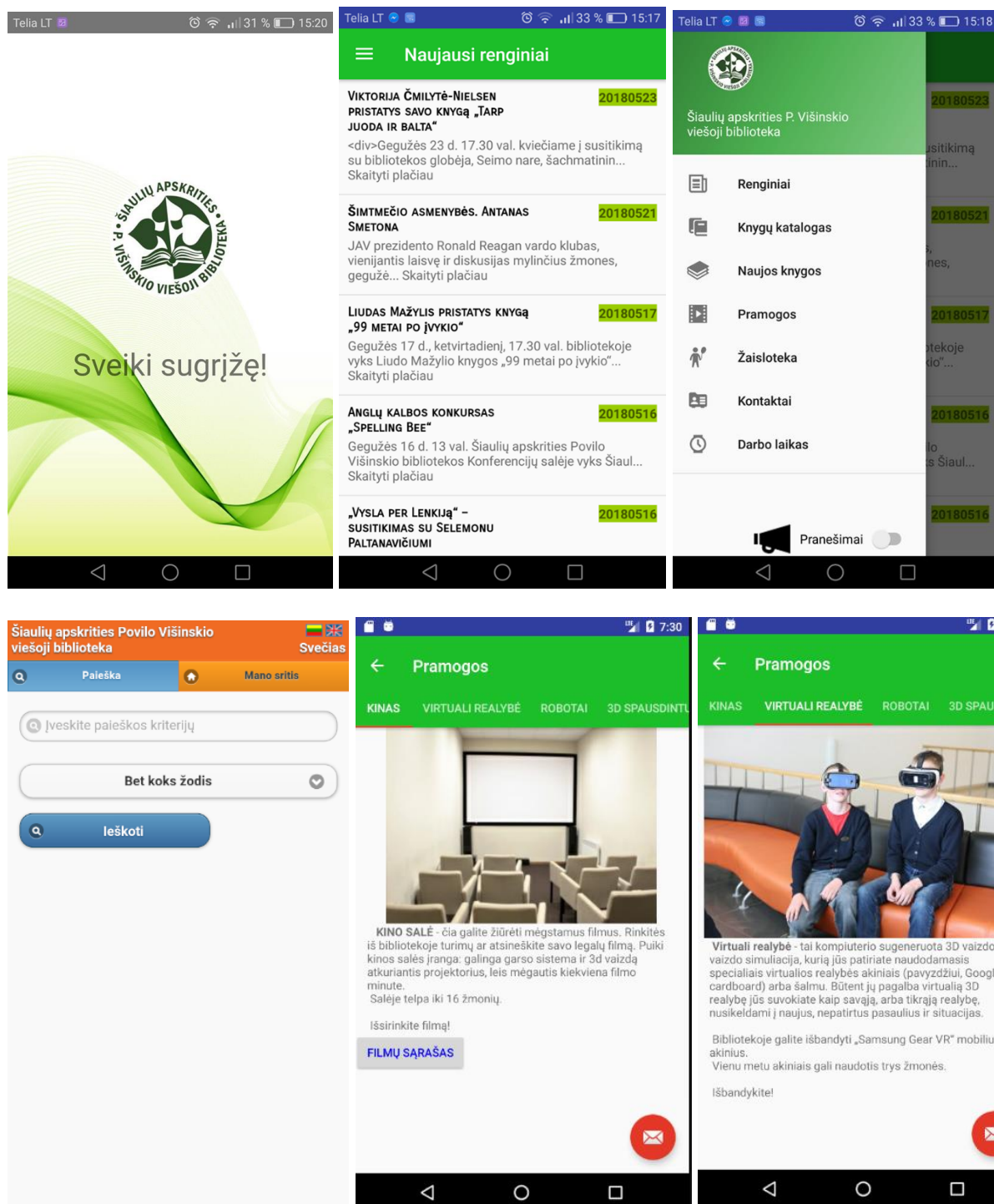
„Android“ telefonais naudojasi daugiau nei 80% visų respondentų.

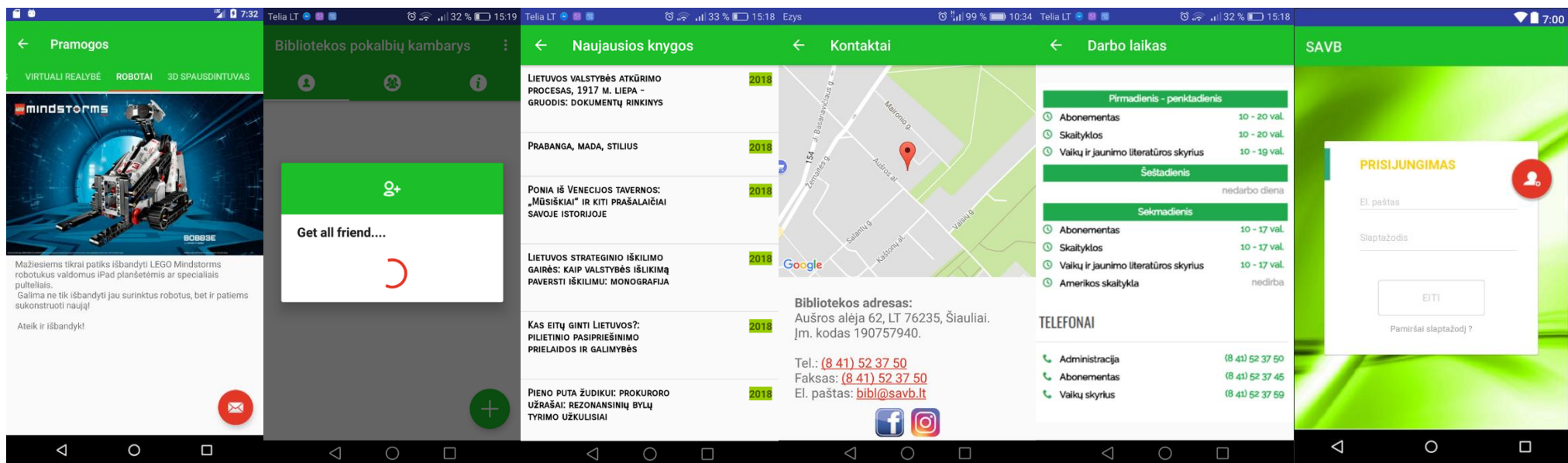
Vartotojo sąsajos maketai

Pradinio projekto išvaizda buvo maketuojama naudojantis „MarvelApp“ įrankiu.

Nuoroda į projektą: <https://marvelapp.com/22ac01a>

Galutinė programos išvaizda :

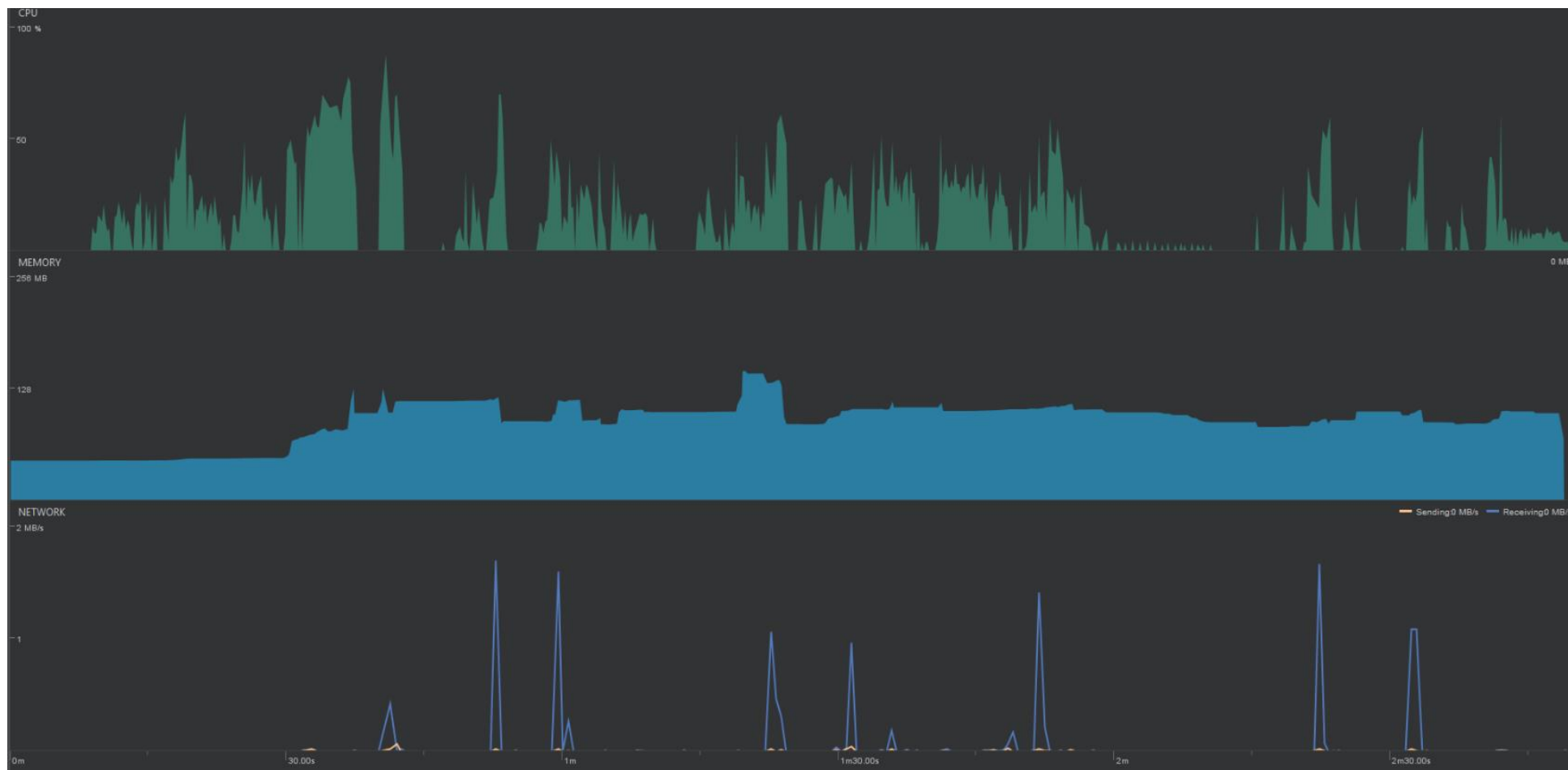




Virtualūs įrenginiai aplikacijos testavimui:

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk
	Htc One A9 API 24		1080 × 1920: xxhdpi	24	Android 7.0 (Google APIs)	x86	1 GB
	Nexus 5X API 23		1080 × 1920: 420dpi	23	Android 6.0 (Google APIs)	x86_64	4 GB
	Nexus One API 25		480 × 800: hdpi	25	Android 7.1.1 (Google APIs)	x86	1 GB
	Pixel XL API 24		1440 × 2560: 560dpi	24	Android 7.0 (Google APIs)	x86	1 GB
	Samsung Galaxy Note 4 API 23		1440 × 2560: xxhdpi	23	Android 6.0 (Google APIs)	x86	4 GB

pav. 7-3. Android virtualių įrenginių sąrašas.



pav. 7-4. Resursų naudojimo grafikas.

ATSILIEPIMAS

Apie Šiaulių apskrities Povilo Višinskio bibliotekos mobiliosios Android aplikacijos kūrimą

2018-05-31

Šiauliai

Studento dalykinių žinių ir gebėjimų vertinimas:

Projekto metu buvo pateikta užduotis – sukurti bibliotekos lankytojams skirtą mobiliąją Android programėlę. Užduočiai atlikti pasirinkti tinkami įrankiai. Sukurtas patrauklus, šiuolaikiškas aplikacijos dizainas. Elementai išdėstyti aiškiai, patogus ir lengvas valdymas. Iki nurodyto termino gegužės 31d. realizuotos numatytos funkcijos: atidaromas knygų katalogas, pateikiama informacija apie renginius, naujas knygas, edukacijas. Sukurta Žaisloteikos skiltis, kurioje vartotojas gali susisiekti su bibliotekininku susirašinėjimo forma. Iki Rugsėjo 10 aplikacija bus toliau testuojama ir tobulinama, siekiant vartotojui pateikti aukštos kokybės produktą. Studento dalykinių žinių užteko įgyvendinti visiems reikalavimams. Gebėjimai vertinami puikiai.

Studento gebėjimas planuoti laiką, planuoti ir vykdyti darbus, spręsti kylančias problemas:

Studentas geba planuoti laiką ir vykdyti darbus pagal pateiktą grafiką. Aplikacija sukurta laiku. Iškilusios problemos buvo išspręstos, rasti racionalūs sprendimai.

Studento komunikaciniai gebėjimai:

Projekto vykdymo metu komunikacija su studentu vyko sklandžiai. Matas geba greitai išanalizuoti ir suprasti pateiktus reikalavimus. Teikia savo pasiūlymus, diskutuoja.

Studento iniciatyvumas ir motyvacija:

Studentas yra iniciatyvus, pasižymi pareigingumu, domisi tuo ką daro, aplikacija kuriama neatmestina su motyvacija.

Studento Mato Dudoravičiaus darbas vertinamas puikiai.

Projekto vadovė

Urtė Vaserienė