

ŠIAULIŲ UNIVERSITETAS
INFORMATIKOS, MATEMATIKOS IR E. STUDIJŲ INSTITUTAS
INFORMATIKOS KATEDRA

Giedrius Dacys

Informatikos specialybės II kurso nuolatinių studijų studentas

Duomenų mainų išskirstytosiose sistemose,

naudojančiose Android OS, modeliai

Data Exchange Models for Distributed Android-based Systems

MAGISTRO DARBAS

Darbo vadovas:

Doc. Dr. V. Giedrimas

Recenzentas:

Doc. Dr. G. Felinskas

Šiauliai 2014/2015 m.m.

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

Darbo autorias _____

(vardas, pavardė, parašas)

Darbo tikslas ir uždaviniai

Tikslas

Išsamiai aprašyti ir empiriniu būdu pagrįsti Android OS Duomenų mainų (ir apdorojimo) su kitomis išskirstytosios sistemos dalimis metodiką.

Uždaviniai

1. Aprašyti tipinius Android OS ir jai skirtų programų darbo tinkle metodus, akcentuojant išskirstytuosius ir lygiagrečiuosius skaičiavimus.
2. Pateikti Android OS ir jai skirtų programų naudojimo probleminių atvejų pavyzdžių, analizuoti jų problematiką.
3. Pasiūlyti teorinius šių problemų sprendimo metodus ir juos patikrinti empiriniu būdu.
4. Pateikti apibendrintą metodiką, kaip turėtų būti perduodami ir apdorjami duomenys lygiagrečiose ir išskirstytose sistemose, jei vienas jų komponentų yra išmanusis Android įrenginys.

Darbo vadovo _____

(vardas, pavardė, parašas)

Turinys

1. Įvadas.....	5
2. Teorinė dalis.....	6
2.1. Temos analizė.....	6
2.2. Darbo srities analizė.....	6
2.3. Darbinės srities modelis.....	11
3. Projektinė dalis.....	12
3.1. Įrankių ir priemonių pasirinkimo analizė.....	12
3.1.1. Technologijos ir realizavimo metodai.....	12
3.1.2. Darbo priemonių galimybių analizė.....	16
3.2. Pradinis projekto aprašymas.....	19
3.2.1. Aprašymas.....	19
3.2.2. Android OS programos duomenų mainų su kitomis išskirstytųjų sistemų dalimis UML diagramomis.....	19
4. Darbo eigos aprašymas.....	22
4.1. Galutinio projekto būvio aprašymas.....	22
4.2. Problemų ir jų sprendimų aprašymai ir pagrindimai.....	25
4.3. Darbo rezultatų analizė.....	26
4.4. Patarimai, pastebėjimai ir rekomendacijos.....	27
5. Išvados.....	28
6. Literatūra.....	29
7. Anotacija.....	31
Anotation.....	31

1. Įvadas

Android operacinė sistema (toliau “OS”) tampa vis populiareesnė ir dažnai naudojama ryšiui su interneto serveriais, debesų kompiuterijos elementais ir pan. Kuriant tokias sistemas būtina atsižvelgti į tai, kad dirbame įrenginyje su Android OS, o ne su “klasikine” OS, pvz. Linux. Ypač tai pasakytina, apie situaciją, kai kuriame lygiagrečiąsias Android programas, apdorojame tinkle gautus duomenis. Darbe siekiama aprašyti charakteringus probleminius atvejus ir apibendrinti siūlomus sprendimus.

2. Teorinė dalis

2.1. Temos analizė

Šiuo metu didžioji dalis programų keliasi į debesis arba yra kokių nors sistemų dalis. Tai ypač aktualu mobiliųjų įrenginių (pavyzdžiui Android OS) programoms, kurios leidžia mobiliajame įrenginyje atlikti darbus, kuriuos anksčiau buvo galima atlikti tik asmeniniuose kompiuteriuose. Į tai įeina bendravimas socialiniuose tinkluose, elektroninė bankininkystė ar failų parsisiuntimas iš įvairių portalų ir pan.

Visa tai įgyvendinant mobiliuosiuose įrenginiuose tenka naudoti panašias priemones kaip ir asmeninių kompiuterių programose, tačiau dėl ribotų resursų, t.y. atminties, spartos, ekrano dydžio ir interneto prieigos skirtumų, tenka viska dar labiau optimizuoti t.y. kiek labiau sumažinti duomenų kiekį kurie bus siunčiami programai bendraujant serveriu, programos vartotojo sąsaja padaryti taip kad ji būtų patogi mažesniems ekranams ir turētu tik tai kas reikalinga tam, kad neekvotų papildomų resursų programos veikimo metu kadangi baterijos eikvojimas yra labai svarbus aspektas, kuris gali nulemti ar vartotojai naudosis kuriama programa.

Mobiliųjų įrenginių programos valdomos kitaip nei kompiuterių, todėl tenka atsižvelgti ir į tai. Įrenginiuose su liečiamaisiais ekranais rekomenduojama vengti smulkių detalių vartotojo sąsajoje, taip pat reikia atsižvelgti ir y tai jog nebus pelės. Todėl atkrenta kai kurie įprasti interakcijos metodai, pavyzdžiui pelės žymeklio užvedimas ant elemento.

2.2. Darbo srities analizė

Šiame darbe nagrinėjama Android OS programų komunikacija su serveriais, bei gautų duomenų apdorojimas. Šios temos kaip atskiros smulkios dalelės internete gvildenamos gan dažnai. Tačiau vis dar nėra bendros metodologijos apibendrinančios kaip realizuoti android programos ir kažkokio tai nutolusio įrenginio duomenų mainus.

Didžiosios pasaulio kompanijos savo teikiamoms paslaugoms turi parengusios priemones, kurias greitai galima integruoti į savo programą taip realizuojant programos komunikaciją su jų serveriais. Puikus pavyzdys yra Facebook ir Google+ socialinių tinklų API, kurie ne tik suteikia lengvą prieigą prie jų serverių teikiamų paslaugų, tačiau ir siūlo įrankius kaip mygtukai kurie automatiškai įvykdo prisijungimą prie jų paslaugų.

Jei nutinka taip, kad tenka kurti programą kuri turės keistis duomenimis su koku nors nutolusiu įrenginiu, tenka visą darbą atlikti pačiam programuotojui, o į tai susidės iš programos lygiagretinimo, gaunamų duomenų apdorojimo ir atvaizdavimo.

Tačiau jei serveris ar kitas įrenginys, su kuriuo bendraus kuriama Android OS programa, neturės jokios pritaikytos sąsajos tokiai komunikacijai, teks pasirūpinti ir tuo, o tai yra dar vienas nemažas darbas kuris apsunkina vien tuo jog reikia nuspręsti kaip visa tai realizuoti. Būtent tokį atvejį ir nagrinėja šis darbas.

Prieš kuriant android programą, kuri bendraus su kokia nors nutolusia sistema, reikia pasidomėti ar toks ryšys bus įmanomas, tai priklausys nuo Adnroid OS palaikomų protokolų kuriais bus atliekami duomenų mainai. Tam tikslui pagal OSI modelio programinį sluoksnį buvo sudarytas tinklo protokolų sąrašas[Lentelė 1] ir nustatyta ar jie palaikomi Android OS arba egzistuoja bibliotekos leidžiančios naudoti tą protokolą.

Lentelė 1 Tinklo protokolų sąrašas pagal OSI modelio programinį sluoksnį.

Nr	Trumpinys	Pavadinimas	Android OS palaikymas
1	ADC	A peer-to-peer file sharing protocol	Java
2	AFP	Apple Filing Protocol	-
3	BACnet	Building Automation and Control Network protocol	HTML, JS
4	BitTorrent	A peer-to-peer file sharing protocol	+
5	BGP	Border Gateway Protocol	-
6	BOOTP	Bootstrap Protocol	+
7	CAMEL	SS7 protocol tool for the home operator	Java
8	Diameter	authentication, authorization and accounting protocol	Java
9	DICOM	includes a network protocol definition	+
10	DICT	Dictionary protocol	Java
11	DNS	Domain Name System	Java
12	DSM-CC	Digital Storage Media Command and Control	Java
13	DSNP	Distributed Social Networking Protocol	-
14	DHCP	Dynamic Host Configuration Protocol	+

15	ED2K	A peer-to-peer file sharing protocol	Java
16	FTP	File Transfer Protocol	+
17	Finger	which gives user profile information	+
18	Gnutella	peer-to-peer file-swapping protocol	Java
19	Gopher	hierarchical hyperlinkable protocol	+
20	HTTP	Hypertext Transfer Protocol	+
21	HTTPS	Hypertext Transfer Protocol Secure	+
22	IMAP	Internet Message Access Protocol	+
23	IRC	Internet Relay Chat	+
24	ISUP	ISDN User Part	Java
25	LDAP	Lightweight Directory Access Protocol	Java
26	MIME	Multipurpose Internet Mail Extensions	+
27	MSNP	Microsoft Notification Protocol	-
28	MAP	Mobile Application Part	-
29	Mosh	Mobile Shell	+
30	NetBIOS	File Sharing and Name Resolution protocol - the basis of file sharing with Windows	-
31	NNTP	Network News Transfer Protocol	+
32	NTP	Network Time Protocol	+
33	NTCIP	National Transportation Communications for Intelligent Transportation System Protocol	-
34	POP3	Post Office Protocol Version 3	+
35	RADIUS	authentication, authorization and accounting protocol	-
36	RDP	Remote Desktop Protocol	+
37	Rlogin	UNIX remote login protocol	+
38	rsync	file transfer protocol for backups, copying and mirroring	Java
39	RTP	Real-time Transport Protocol	Java
40	RTSP	Real-time Transport Streaming Protocol	+
41	SSH	Secure Shell	+

42	SISNAPI	Siebel Internet Session Network API	-
43	SIP	Session Initiation Protocol, a signaling protocol	+
44	SMTP	Simple Mail Transfer Protocol	+
45	SNMP	Simple Network Management Protocol	+
46	SOAP	Simple Object Access Protocol	+
47	SMB	Microsoft Server Message Block Protocol	+
48	STUN	Session Traversal Utilities for NAT	Java
49	TUP	Telephone User Part	-
50	Telnet	remote terminal access protocol	+
51	TCAP	Transaction Capabilities Application Part	-
52	TFTP	Trivial File Transfer Protocol, a simple file transfer protocol	Java
53	WebDAV	Web Distributed Authoring and Versioning	Java
54	XMPP	instant-messaging protocol	+

Siekiant realizuoti duomenų mainus tarp išskirstytosios sistemos dalių, būtina pasirinkti kaip bus formatuojami duomenys, kuriais dalinamasis sistemos elementai. Duomenis galima formuoti įvairiai, tačiau, dėl didesnio patogumo ir efektyvumo reikėtų naudoti žymėjimo kalbas. Šiame darbe nagrinėjamiems atvejams dažniausiai naudojamos XML ir JSON kalbos. Kadangi nagrinėjamosiose sistemose naudojamas mobilusis įrenginys, kuris turi ribotus resursus, todėl duomenų formatavimo efektyvumas tampa labai svarbiu veiksniumi sistemos projektavime. Dėl šios priežasties, žymėjimo kalba, kuri bus naudojama sistemoje, turi pasižymėti dideliu efektyvumu. Įgyvendinant šiame darbe nagrinėjama metodiką, duomenų formatavimui buvo pasirinkta JSON žymėjimo kalba. Ši kalba buvo pasirinkta remiantis Montanos Valstijos Universiteto atliktu tyrimu [16], kuriu rezultatai ([Lentelė 2], [Lentelė 3], [Lentelė 4]) parodė, kad JSON žymėjimo kalba yra efektyvesnė ir labiau tinka nagrinėjamam atvejui.

Lentelė 2 JSON ir XML duomenų apdorojimo testų rezultatai.

Scenario/Measure	JSON	XML
Scenario 1 Total Time	78.26 seconds	75.77 minutes
Scenario 1 Average Time Per Object	0.08 ms	4.55 ms
Scenario 1 Average User CPU Utilization	86%	55%
Scenario 1 Average System CPU Utilization	13%	45%
Scenario 1 Average Memory Utilization	27%	29%
Scenario 2 Total Time for 100,000 Objects	7.5 seconds	310 seconds
Scenario 2 Average Time Per Object	0.08 ms	3.1 ms
Scenario 2 Average User CPU Utilization	83-88%	65-67%
Scenario 2 Average System CPU Utilization	11-14%	32-33%
Scenario 2 Average Memory Utilization	68%	68%

Lentelė 3 Duomenų formatavimo JSON ir XML žymėjimo kalbomis testų rezultatai.

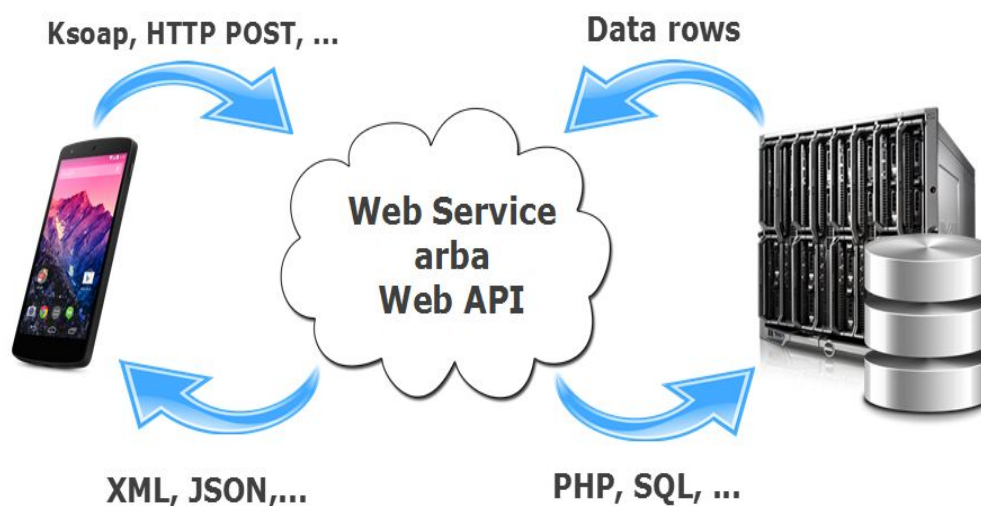
Measure	JSON Distribution	XML Distribution
Total Time (ms)	{2213.1, 3127.99, 4552.38, 6006.72, 7497.36}	{61333.68, 123854.59, 185936.27, 247639.81, 310017.47}
Average Time Per Object (ms)	{0.11, 0.08, 0.08, 0.08, 0.07}	{3.07, 3.10, 3.10, 3.10, 3.10}
Average % User CPU	{29.07, 83.84, 88.01, 88.65, 88.70}	{65.80, 67.43, 66.69, 67.24, 66.64}
Average % System CPU	{14.80, 15.84, 11.99, 11.36, 11.30}	{32.36, 32.57, 33.31, 32.76, 33.36}
Average % Memory	{67.97, 68.07, 68.06, 68.06, 68.06}	{68.08, 68.08, 68.08, 68.11, 68.79}

Lentelė 4 JSON ir XML duomenų formatavimo laikas, antrajame scenarijuje.

	JSON	XML
Trial 1 Number Of Objects	20000	20000
Trial 1 Total Time (ms)	2213.15	61333.68
Trial 1 Average Time (ms)	0.11	3.07
Trial 2 Number Of Objects	40000	40000
Trial 2 Total Time (ms)	3127.99	123854.59
Trial 2 Average Time (ms)	0.08	3.10
Trial 3 Number Of Objects	60000	60000
Trial 3 Total Time (ms)	4552.38	185936.27
Trial 3 Average Time (ms)	0.08	3.10
Trial 4 Number Of Objects	80000	80000
Trial 4 Total Time (ms)	6006.72	247639.81
Trial 4 Average Time (ms)	0.08	3.10
Trial 5 Number Of Objects	100000	100000
Trial 5 Total Time (ms)	7497.36	310017.47
Trial 5 Average Time (ms)	0.07	3.10

Lentelėse aiškiai matyti, kad JSON yra efektyvesnis už XML. Iš tyrimo rezultatus apibendrinančių lentelių ([Lentelė 2], [Lentelė 3], [Lentelė 4]) matyti, kad JSON naudoja didesnius procesoriaus resursus, nei XML, tačiau tokį pat duomenų kiekį suformatuoja apie dešimt kartų greičiau.

2.3. Darbinės srities modelis



Pav. 1 Darbinės srities modelis.

3. Projektinė dalis

3.1. Įrankių ir priemonių pasirinkimo analizė

3.1.1. Technologijos ir realizavimo metodai

Kadangi nagrinėjama Android OS ir jai skirtų programų darbas tinkle, tai tyrimui atlikti buvo pasirinktos šios technologijos:

- ✓ PHP - dinaminė interpretuojama programavimo kalba.
- ✓ Java - objektiškai orientuota programavimo kalba.
- ✓ JSON - žymėjimo kalba.
- ✓ XML - žymėjimo kalba.
- ✓ SQL - standartinė kalba darbui su duomenų bazėmis.
- ✓ UML - vieninga modeliavimo kalba.
- ✓ Web Service - dviejų įrenginių komunikavimo tarpusavyje tinkle metodas.
- ✓ Web Api - programos sąsaja su serveriu.

PHP

PHP - skriptų (angl. script) kalba, plačiai naudojama interneto serveriuose, kurie naudoja *Apache HTTP* serverio įrangą. *PHP* turi daug pritaikymo galimybių, tačiau dažniausiai ši kalba naudojama taikomosiomis interneto aplikacijoms kurti. *PHP*, *Linux*, *Apache HTTP* serverio ir *MySQL* duomenų bazės kombinacija užtikrina didelės dalies interneto serverių funkcionavimą. [1]

PHP kalba yra kilusi iš Perl kalbos, tiksliau iš Perl skriptų rinkinio, 1995 m. sukurto Rasmus Lerdof skirto asmeninei jo svetainei kurti. Šis priemonių rinkinys leido lengvai valdyti pateiktas formas, vykdyti prisijungimo ir sekimo funkcijas. [2]

Privalumai:

- Nemokama.
- Veikia įvairiose operacinėse sistemose (Linux, Windows, MacOS).
- Veikia daugelyje Web serverių.
- Programavimui PHP kalba be serverio reikalingos tik teksto redagavimo programos.
- Nesunkiai generuojama JSON.

Trūkumai:

- Interpretuojama kalba;

Java

Java - tai griežtai tipizuota nepriklausanti nuo kompiuterio architektūros, daugiaprocesė objektinio programavimo kalba, kurios esmė yra klasės. Ji priklauso tai pačiai kalbų grupei kaip C++, Pascal ir Delphi kalbos. [3]

Android aplikacijos yra kuriamos Java programavimo kalba. Android SDK (Software Development Kit) įrankiai kompiliuoja Java kalba parašytos programos kodą kartu su papildomais duomenimis ir resursų failais, į Android OS vykdomąjį failą su “apk” galūne.[4]

Privalumai:

- Galima naudoti Java bibliotekas Android OS programose.
- Plačiai paplitusi kalba, todėl yra daug pamokų ir resursų padedančių kurti programas.
- Objektiškai orientuota kalba.

Trūkumai:

- Java kalba parašytos programos veikia tik naudodamos Java virtualiąją mašiną.

JSON

JSON - tekstinis, žmogui suprantamas duomenų formatas atvaizduojantis duomenų struktūras ir asociatyvius masyvus. JSON dažniausiai naudojamas Ajax perduoti duomenims ir yra naudojamas kaip XML alternatyva. [5]

PHP ir Android SDK suteikia įrankius JSON dokumentų kūrimui bei nuskaitymui, todėl šis formatas puikiai tinka perduoti duomenis tarp išskirstytųjų sistemų.

Privalumai:

- Lengvai skaitomas
- JSON užkoduoti duomenys padidėja tik truputėlį.
- JSON generavimui nebūtini jokie įrankiai, tai galima atlikti tiesiog generuojant tekstą.
- Dauguma programavimo kalbų turi integruotus įrankius JSON nuskaitymui, generavimui.

Trūkumai:

- Galima užkoduoti tik tuos duomenis, kuriuos galima perteikti tekstu.

SQL

SQL - struktūrizuota užklausų kalba (angl. *Structured Query Language*), skirta bendravimui su duomenų baze. SQL naudojama tokioms užduotims kaip atrinkti, atnaujinti, papildyti ar trinti duomenų bazės duomenis. Duomenų bazių valdymo sistemų yra daug, populiariausios Oracle, MySQL, Microsoft SQL Server ir kit. Jos visos naudoja papildyta SQL kalbą, tačiau standartinės SQL komandos (kaip SELECT, INSERT, UPDATE, CREATE, DELETE, DROP) gali būti naudojamos su bet kuria duomenų baze. [6]

Privalumai:

- Nemokama.
- Standartinės komandos veikia visuose duomenų bazių valdymo sistemose.
- Plačiai naudojama, todėl apstu pagalbinių pamokų ir resursų palengvinačių programavimą.

Trūkumai:

- Funkcionalumas priklauso nuo duomenų bazės valdymo sistemos.

UML

UML - vieninga modeliavimo kalba (angl. *Unified Modeling Language*) kurios tikslas suteikti būdą vizualizuoti sistemų struktūrą ir veikimą.

Šiuo metu UML yra labiausiai paplitęs programinės įrangos specifikavimo standartas, palaikomas tokių kompanijų kaip IBM ar Borland. [7]

Privalumai:

- Visuotinai pripažintas standartas.
- Leidžia vizualizuoti sistemą norimu detalumu ir norimomis dalimis.
- Sistemos atnaujinimai yra žymiai lengvesni turint sistemos UML dokumentaciją.
- Didelis UML įrankių pasirinkimas.

Web Service

Web Service - kliento ir serverio aplikacijos kurios komunikuoja tinkle naudojant HTTP protokolą. Kaip apibrėžia W3C (angl. *World Wide Web Consortium*) web service suteikia galimybę sąveikauti aplikacijoms veikiančioms įvairiose platformose ir karkasuose, tai suteikia XML aprašai. Web service programos gali sąveikauti tarpusavyje, apjungdamos duomenis ir funkcijas suteikiant sudėtingesnes paslaugas. [8]

Android OS programos darbui su Web Service reikalinga papildoma Java biblioteka kSOAP2.

Privalumai:

- Sąveikos nelimituoja programų platformos ar karkasų skirtumai.
- XML aprašai leidžia Web Service komunikuoti tarpusavyje.

Trūkumai:

- Lėtas veikimas, dėl didelio papildomų duomenų kiekio užklausoje.
- Sudėtingas XML aprašo kūrimas.

Web Api

Web Api - aplikacijos programavimo sąsaja (angl. *Application Programming Interface*) kuri tipiškai įgyvendinta per HTTP užklausas, kuriuose paduodama ir gaunam JSON, XML arba tiesiog duomenys. Web Api neturi jokio standartizavimo, taigi nėra jokių automatinių paslaugų kvietimų ar atradimų, skirtingai nei Web Service.

Jei Web Api sukurta remiantis HTTP užklausomis, didžioji dalis Web Api būtent ir naudoja šį metodą, tai Android OS programai nereikės jokių papildomų bibliotekų, kadangi visoms HTTP užklausoms Android SDK turi integruotą Apache HttpRequest biblioteką. [9]

Privalumai:

- Greitas veikimas.
- Lengvas sukūrimas.
- Standartų nebuvimas leidžia lengviau įgyvendinti labiau specifines užduotis.
- Sąveikos nelimituoja platformų ir karkasų skirtumai.

Trūkumai:

- Naudojant ne savo kurtas Web Api, nėra aišku kiek ilgai jos dar funkcionuos.
- Būtina dokumentacija, arba trumpas aprašas, kadangi be to neįmanoma žinoti kaip naudotis specifine Web Api.

3.1.2. Darbo priemonių galimybių analizė

Android OS duomenų mainų (ir apdorojimo) su kitomis išskirstytomis sistemos dalimis tyrimui ir empiriniam pagrindimui, buvo pasirinktos šios darbo priemonės:

- ✓ Android Studio - programavimo aplinka.
- ✓ NetBeans PHP - programavimo aplinka.
- ✓ Volley - Java biblioteka.
- ✓ kSOAP2 - Java biblioteka.
- ✓ UMLet - UML diagramų kūrimo įrankis.

Android Studio

Android Studio - oficiali integruota programavimo aplinka (IDE) (angl. *Integrated Development Enviroment*) pagrįsta IntelliJ IDEA pagrindu. Lyginant su Eclipse aplinka Android Studio sutiekia naują bibliotekų, programos nustatymų sistemą Gradle. Taip pat daugybę integruotų įrankių skirtų būtent Android OS programų kūrimui, pavyzdžiui resursų redaktorių kuris leidžia lengvai viename patogiam lange redaguoti vertimus. [10]

Privalumai:

- Oficiali Android OS programavimo aplinka
- Moderni aplinka.
- Didžiulė nustatymų įvairovė.
- Vartotojo sąsaja pritaikyta Android programų kūrimui.
- Įvairių įrankių ir priemonių gausa palengvinanti programų kūrimą.

Trūkumai:

- Sunkus perėjimas iš Eclipse į Android Studio, dėl pasikeitusios Android OS programų projektų struktūros ir vartotojo sąsajos.

NetBeans PHP

NetBeans PHP - IDE skirta PHP, JavaScript ir HTML 5 aplikacijų kūrimui. Skirtingai nei programuojant PHP kalba tekstų redaktoriuje, NetBeans siūlo galingą kodo redaktorių kuris pagreitina kodo rašymą ir aptinka bei pažymi klaidas. Taip pat NetBeans gerai tuo jog galima prisijungti tiesiai prie serverio ir dirbti su ten esančiais failais. [11]

Privalumai:

- Galingas kodo redaktorius su sufleriu ir klaidų aptikimu, pažymėjimu.
- Galimybė dirbti prisijungus prie serverio.

Volley

Volley - HTTP biblioteka skirta palengvinti ir pagreitinti Android OS programų darbą tinkle, bei sumažinti kodo kiekį tinklo operacijoms atlikti. Jau standartiškai visos Volley tinklo užklausos vykdomos asinchroniškai todėl nebereikia naudoti AsyncTask arba kitų lygiagrečiam darbui skirtų klasių. [12]

Volley savyje talpina įvairias priemones ir galimybes. Kelios svarbiausios:

- Užklausų eiliškumas ir prioritetų nustatymas.
- Efektyvus užklausų talpinimas ir atminties valdymas. Tai ypač svarbu mobiliųjų įrenginių programose dėl ribotų resursų.
- Užklausų atšaukimas ar sustabdymas. Tai ypač aktualu Android OS programose, dėl jų nuolat persikuriančios ekosistemos.
- Integruoti įrankiai JSON užklausoms. Tai užtikrina, kad sėkmingai įvykdyta užklausa jau gražins JSONObject ar JSONArray objektus.
- Įrankiai darbui su nuotraukomis, tarp jų modifikuota ImageView klasė kuri pati atsiunčia ir savyje atvaizduoja nuotraukas.

Privalumai:

- Nemokama.
- Užima mažai vietos, tik 85 KB.
- Visiškai pakeičia gijų naudojimą darbui tinkle, taip sumažinant programos kodo kieki.
- Idealiai tinka Web Api užklausoms, kurios gražina JSON duomenis.
- Palaiko HTTP POST ir GET užklausas.

Trūkumai:

- Vis dar nėra Android SDK *v4 Support library* bibliotekos dalis.
- Vis dar nėra iki galo išvystytas darbas su failais.
- Netinka darbui su dideliais duomenų kiekiais, nes visus apdorojamus duomenis laiko atmintyje.

kSOAP2

kSOAP2 - biblioteka skirta darbui tinkle su Web Service paslaugomis ir orientuota į ribotų resursų įrenginius tokius kaip mobilieji telefonai. Dėl bibliotekos minimalizmo, tai nėra visiems darbams tinkamas įrankių rinkinys, tačiau jo visiškai pakanka darbui su populiariausiais SOAP varikliais. [13]

Į kSOAP2 biblioteką įeina šie įrankiai:

- XML Parser - įrankis skirtas darbui su XML duomenimis, kuriuos naudoja SOAP užklausa.
- De/Serializer - įrankis atsakingas už objektų užkodavimą atkodavimą, tam kad juos būtų galima naudoti užklausoms.
- Transport Layer - užklausa tarp serverio ir kliento vykdomas įrankis.

Privalumai:

- Nemokama

UMLet

UMLet - nemokamas, atviro kodo UML įrankis su paprasta vartotojo sąsaja, kuris leidžia greitai kurti UML diagramas, generuoti sekų ir veiksmų diagramas iš teksto. UMLet veikia kaip atskira programa arba kaip Eclipse įskiepis, Windows, Mac OS X ir Linux operacinėse sistemose. [14]

Privalumai:

- Nemokama.
- Nereikalauja įdiegimo.
- Nesunku pridėti trūkstamų elementų diagramoms.

Trūkumai:

- Nėra integruoto diagramų į kodą generavimo galimybės, tačiau šį funkcionalumą galima pasidaryti pačiam, kadangi diagramos saugomos XML formatu.

3.2. Pradinis projekto aprašymas

3.2.1. Aprašymas

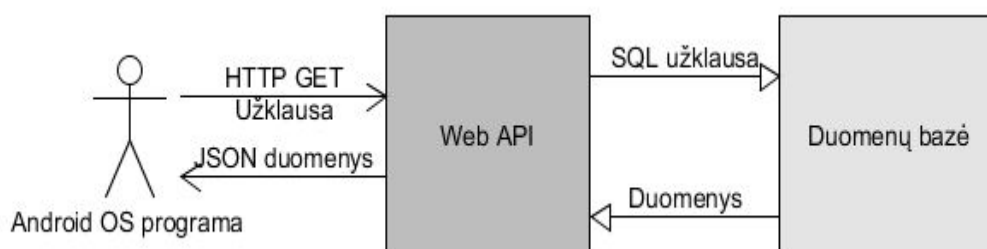
Siekiant aprašyti ir empiriniu būdu pagrįsti Android OS programų duomenų mainų ir apdorojimo su kitomis išskirstytosios sistemos dalimis metodiką, buvo nuspręsta pasitelkti du projektus:

1. Interneto portalo klientinę programos ir jai skirtos Web Api kūrimą.
2. Programą kuri kas numatytą laiko intervalą kreipsis į prietaisą prijungtą prie tinklo, kuris praneš apie tai ar užrakintos durys.

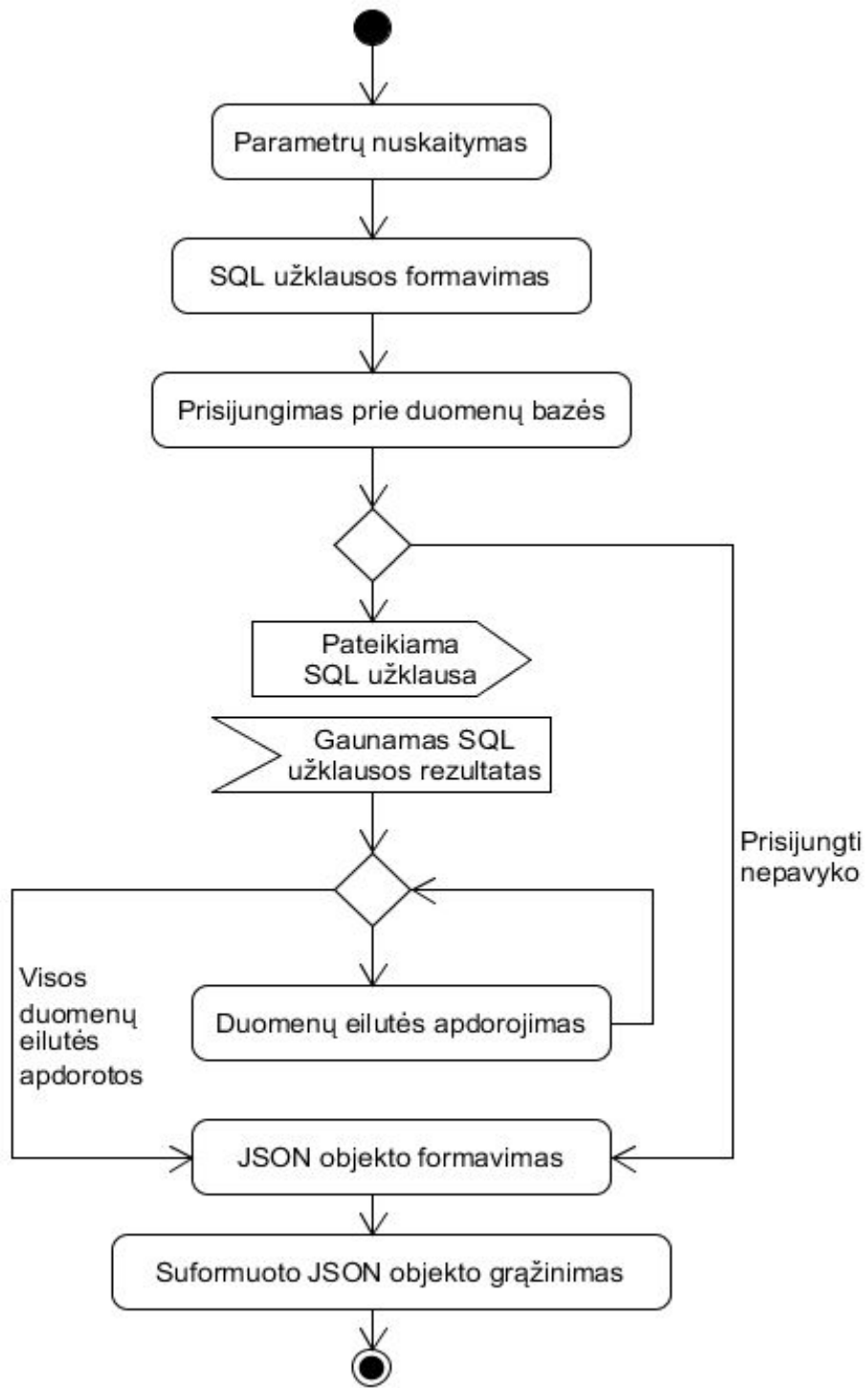
Abiejų projektų programos, kaip užklausų rezultatus, gaus JSON tipo duomenis, kuriuos turės apdoroti ir atvaizduoti vartotojo sąsajoje. Taip pat abi programos naudos Volley biblioteką užklausoms pateikti ir gauti rezultata, Taip nuspręsta dėl jau paruoštų įrankių gausos kuriuos siūlo Volley biblioteka darbui su JSON duomenimis.

Pirmajai programai bus kuriamas Web Api PHP kalba kuris užklausų rezultatams reikalingus duomenis ims iš MySQL duomenų bazės. Web Api vietoj Web Service buvo pasirinktas dėl spartumo ir realizavimo laisvumo. Tai pat tai palengvino testavimą, kadangi Web Api grąžinamus duomenis buvo galima pasiekti iš bet kokios naršyklės be jokių papildomų įrankių.

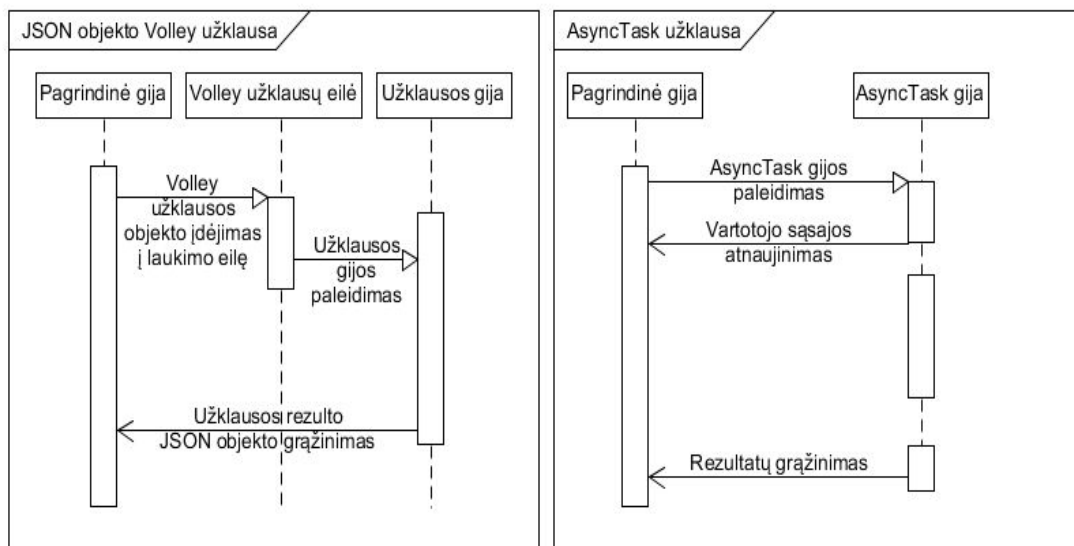
3.2.2. Android OS programos duomenų mainų su kitomis išskirstytųjų sistemų dalimis UML diagramomis



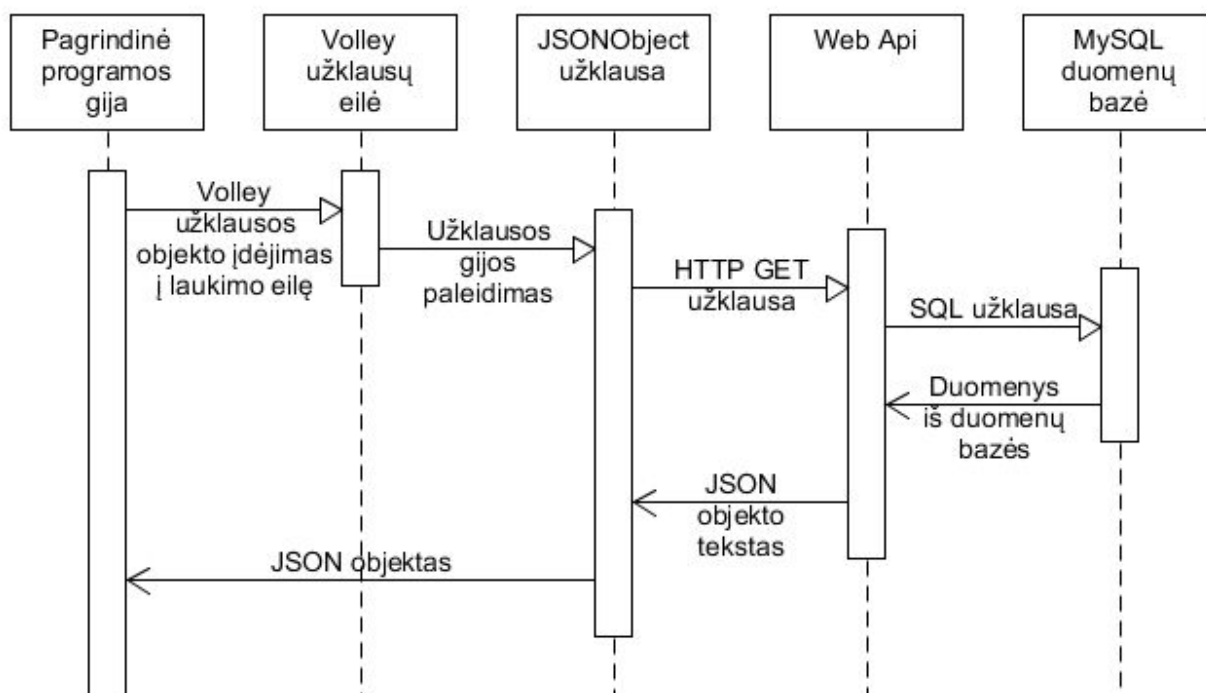
Pav. 2 AndroidOS programos duomenų maiani su duomenų baze.



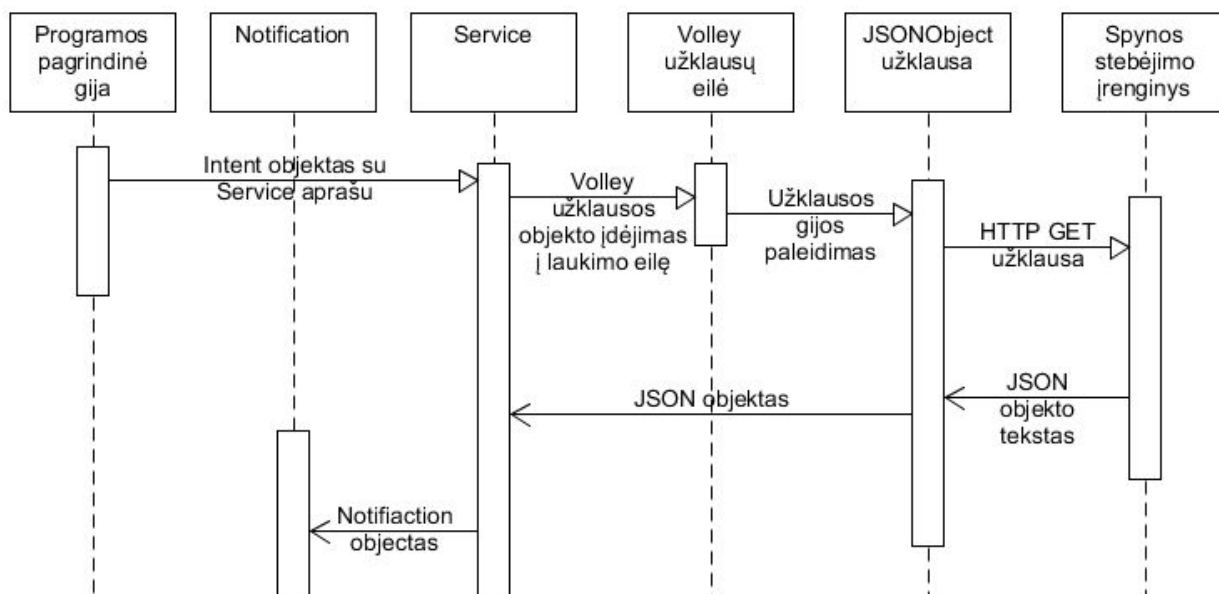
Pav. 3 Web API veiksmų diagrama.



Pav. 4 Android OS programos užklausa, naudojant Volley biblioteką ir naudojant AsyncTask klasę, sekų diagramos.



Pav. 5 Klinetinės programos duomenų mainai su internetinio portalo Web Api.



Pav. 6 Android programos duomenų mainai spynos būseną tikrinančiu įrenginiu.

4. Darbo eigos aprašymas

4.1. Galutinio projekto būvio aprašymas

Realizuoti abu projektai, interneto portalo klientinė programa ir jai skirtos Web Api kūrimas, bei spynos būseną stebinti programa. Pirmajam projektui kurtas Web Api buvo suprojektuotas taip, kad tiktų daugeliui atvejų, gaunant ar keičiant duomenų bazės duomenis. Visa tai įgyvendinta PHP ir SQL programavimo kalbomis. Kiekvieną Web Api galima suskirstyti į šias dalis:

- Parametrų nuskaitymas

Priklausomai nuo to kokio tipo užklausa kreipėsi į Web Api php failą, bus nuskaityti parametrai PHP kalboje. Yra du HTTP užklausų tipai:

- POST

```

$par = "";
if(isset($_POST["par1"])){
    $par = $_POST["par1"];
}
  
```

Pav. 7 HTTP POST užklausos parametro gavimas PHP kalboje.

- GET

```
$par = "";  
if(isset($_GET["par1"])){  
    $par = $_GET["par1"];  
}
```

Pav. 8 HTTP GET užklauso parametro gavimas PHP kalboje.

- SQL užklauso generavimas PHP kalboje

Generuojama SQL užklausa kuri bus vykdoma duomenų bazėje ir gražins rezultatą. Jei Web Api užklausa buvo su parametrais, parametrai įterpiami į užklauso tekstą taip patikslinant, kas turi būti gražinama.

```
$sql = "SELECT id, title, date, logo_name, logo_path  
FROM lh_events WHERE active=1 ORDER BY date DESC LIMIT " . $par . ", 20"
```

Pav. 9 SQL užklauso teksto generavimas PHP kalboje.

- SQL užklauso rezultatų apdorojimas PHP kalboje

PHP kalboje rezultatai gražinami dvimačiuose masyvuose, kiekviena eilutė talpina stulpelių, kurie buvo nurodyti SQL užklausoje, reikšmes, kurie pasiekiami nurodžius jų pavadinimus.

```
$data = array();  
while ($row = mysqli_fetch_array($result)) {  
    $tmp = array();  
    $tmp['id'] = $row['id'];  
    $tmp['title'] = $row['title'];  
    $tmp['logo_url'] = "http://domen.com/" . $row['logo_path'] . $row['logo_name'];  
    $tmp['date'] = $row['date'];  
    array_push($data, $tmp);  
}
```

Pav. 10 SQL užklauso rezultatų apdorojimas ir sudėjimas į dvimatį masyvą.

- **JSON objekto formavimas PHP kalboje**

Formuojant JSON objektą geriausia nurodyti Web Api užklauso būseną nurodant, pavyzdžiui įterpian "status" reikšmę kuri nurodys ar užklausa buvo sėkminga ir ar verata toliau nuskaityti gautus duomenis.

```
$obj = array();
$obj['status'] = $status;
$obj['offset'] = $par + count($data);
$obj['images'] = $data;
$json = json_encode($obj);
```

Pav. 11 JSON objekto sukūrimas PHP kalboje.

- **JSON objekto kaip rezultato grąžinimas**

Rezultatas grąžinamas naudojantis funkcija *echo*, tačiau prieš ją reiktų nurodyti kokio tipo duomenys bus grąžinami. Tai smarkiai palengvins testavimą naršyklėje kurioje įdiegtas įskiepis kuris struktūrizuoja JSON duomenis.

```
header('Content-Type: application/json');
echo $json;
```

Pav. 12 JSON objekto grąžinimas PHP kalboje.

Klientinės Android OS programos kurios dalinsis duomenimis su Web Api ar kitu išskirstytosios sistemos elementu, užklausoms naudoja Volley biblioteką. Taip buvo nuspręsta dėl Volley bibliotekos parankumo naudojant JSON duomenis, užklausų valdymo, bei sumažėjusio kodo kiekio.

Volley JSON užklauso sukuriamos sukūrus *JsonObjectRequest* arba *JsonArrayRequest* objektą, jam nurodžius Web Api adresą, bei pavykusios užklauso ir klaidos klausymosi objektus, bei visa tai įdėjus į Volley užklausų eilę[Pav.12].


```

JsonObjectRequest request = new JsonObjectRequest(url, obj,
    new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {
            //gautų duomenų apdorojimas ir atvaizdavimas
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            //pranešimas apie nepavykusią užklausą
        }
    });
Volley.newRequestQueue(getActivity()).add(request);

```

Pav. 13 JSON objekto užklausa naudojant Volley biblioteką Android OS programoje.

4.2. Problemų ir jų sprendimų aprašymai ir pagrindimai

Gauti duomenys su Volley bibliotekos užklausa apdorojami pagrindinėje gijoje, tai nekelia problemų jei duomenys nereikalauja specialaus apdorojimo. Jei vis dėlto gauti duomenys yra sudėtingesni nei paprastas tekstas ar skaičiai, pavyzdžiui HTML kodas, kurį reikia atvaizduoti vartotojo sąsajos *TextView* elemente.

Tokiems atvejams tenka naudoti *AsyncTask* klasę, kuri leidžia daug resursų reikalaujančias užduotis atlikti kitoje gijoje ir nereikalauja naudoti *Handler* konstrukcijų atnaujinant vartotojo sąsają.

Tipinis šios atvejis iškilo gaunant puslapio vartotojų komentarų duomenis ir bandant juos atvaizduoti vartotojo sąsajos *ListView* elemente, tai buvo išspręsta apjungus Volley užklausas ir kai kurių duomenų apdorojimą *AsyncTask* gijoje.

```

private class KomentarioFormatavimas extends AsyncTask<Void, Void, Coment> {
    int index = 0;
    public KomentarioFormatavimas(int index){
        this.index = index;
    }
    protected Long doInBackground(Void... parameters) {
        Coment coment = comentsList.get(index);
        String tmp = coment.text;
        //Nustatoma citat7 spalva naudojant HTML
        tmp = comnew.replaceAll("<a href=", "<font href=");
        tmp = comnew.replaceAll("</a>", "</font>");
        tmp = comnew.replaceAll("<span ", "<font ");
        tmp = comnew.replaceAll("</span>", "</font>");
        tmp = comnew.replaceAll("class=\"quote\"", "color=\"" + Const.QouteColor + "\"");
        //Iš pakeisto HTML sugeneruojamas SpannableStringBuilder objektas
        SpannableStringBuilder ssb = (SpannableStringBuilder) Html.fromHtml(tmp);
        ClickableSpan cs;
        //Sukuriamas regex šablonas kuris ieškos nuorodų tekste
        Matcher murl = main.pattern_url.matcher(ssb);
        while (murl.find()) {
            final String url = ssb.toString().substring(murl.start(), murl.end());
            //Sukuriamas ClickableSpan objektas su veiksmu priskirtą adresą atidaryti naršyklėje
            cs = new ClickableSpan() {
                @Override
                public void onClick(View widget) {
                    Tools.OpenInBrowser(url);
                }
            };
            //Priskiriamas ClickableSpan objektas toje teksto vietoje kur yra rastasis adresas
            ssb.setSpan(cs, murl.start(), murl.end(), Spannable.SPAN_EXCLUSIVE_EXCLUSIVE);
        }
        //Atnaujinamas komentaro objektas
        coment.textDisplay = ssb;
        return coment;
    }
    protected void onPostExecute(Coment result) {
        //Atnaujinamas duomenų sąrašassu atnaujintu objektu
        comentsList.set(index, result);
        //Pranešama ListView elemntui kad duomenys pasikeitė ir reikia perpiešimo
        adapter.notifyDataSetChanged();
    }
}
}

```

Pav. 14 AsyncTask kalsė skirta formatuoti ListView elemento duomenis.

4.3. Darbo rezultatų analizė

Aprašyti tipiniai Android OS programų darbo tinkle metodai, akcentuojant išskirstytuosius ir lygiagrečiuosius skaičiavimus.

Pateikti Android OS programų probleminių atvejų pavyzdžiai, analizuojant jų problematiką.

Pasiūlyti teoriniai šių problemų sprendimo metodai, kurie buvo patikrinti empiriniu būdu, t.y. įgyvendinant 2 Android OS programų projektus.

Pateikta apibendrinta metodika, kaip turētu būti realizuojamas duomenų, gavimas, perdavimas ir apdorojimas lygiagrečiuose ir išskirstytose sistemose, kurių vienas iš elementų jei vienas iš jų yra išmanusis Android įrenginys.

4.4. Patarimai, pastebėjimai ir rekomendacijos

Kuriant Android OS programas, rekomenduotina pasidomėti įvairiomis bibliotekomis specifinėms užduotims įgyvendinti. Google siūlo net keturių rūšių *Support* bibliotekas, skirtas naujausių Android OS ekosistemos ir vartotojo sąsajos elementų suderinamumą su senesniais įrenginiais. O kadangi Android OS yra nemokama atviro kodo operacinė sistema, jai yra prikurta daugybė įvairiausių bibliotekų ir įskiepių palengvinančių programavimą ar suteikiančių papildomą funkcionalumą.

Dėl duomenų srauto limitų rekomenduotina kiek įmanoma vengti naudoti XML duomenų formatavimą, jį pakeičiant JSON ar dar koku paprastesniu formatu, kuris tik minimaliai pridės papildomų duomenų formatuojant.

Renkanti kaip realizuoti išmaniojo įrenginio duomenų mainus su duomenų baze, rekomenduotina vengti naudoti Web Service ir vietoj to kurti Web Api kuri grąžins JSON duomenis.

5. Išvados

1. Empiriniu būdu tikrinant analizuojamus probleminius atvejus nustatyta, kad Android OS programos duomenų mainams su nutolusiu įrenginiu, išskirstytoje sistemoje, būtinas tarpininkas, kadangi jis rūpinasi užklausomis ir duomenų mainais.

2. Remiantis Web Service ir Web Api dokumentacijomis nustatyta, kad realizuojant tarpininką tarp Android OS programos ir duomenų bazės, geriau kurti Web Api, nes Web Api neturi griežtai apibrėžtų reikalavimų (nervarbūs duomenų tipai, platformos, kalbos, ...), kas leidžia tai realizuoti bet kokioje sistemoje.

3. Išnagrinėjus Montanos Valstijos Universiteto atlikto tyrimo rezultatus, nustatyta, kad geriau naudoti JSON žymėjimo kalbą duomenų formatavimui, dėl efektyvesnio duomenų apdorojimo greičio ir mažesnių papildomų duomenų kiekio, nei XML žymėjimo kalbą.

4. Atlikus empirinį tyrimą paaiškėjo, kad šiame darbe aprašytas Android OS programų darbo tinkle metodikas, galima pritaikyti didžiajai daliai projektų kuriuose Android išmanusis įrenginys bus kažkokios išskirstytos sistemos dalis.

6. Literatūra

1. What is PHP. Prieiga per internetą: <http://php.net/manual/en/intro-whatis.php> [Žiūrėta 2015-05-03]
2. Toby Joe Boudreaux, PHP 5 vaizdžiai. Kaunas, 2007
3. Aleksas Riškus, Programavimas Java Primoji pažintis, mokomi knyga, 2012. Prieiga per internetą: <https://www.ebooks.ktu.lt/eb/664/programavimas-java-pirmoji-pazintis/> [Žiūrėta 2015-05-03]
4. Application Fundamentals. Prieiga per internetą: <http://developer.android.com/guide/components/fundamentals.html> [Žiūrėta 2015-05-03]
5. Introducing JSON. Prieiga per internetą: <http://www.json.org/> [Žiūrėta 2015-05-03]
6. What is SQL. Prieiga per internetą: <http://www.sqlcourse.com/intro.html> [Žiūrėta 2015-05-03]
7. Introduction To OMG's Unified Modeling Language. Prieiga per internetą: http://www.omg.org/gettingstarted/what_is_uml.htm [Žiūrėta 2015-05-03]
8. TheJava EE 6 Tutorial. Prieiga per internetą: <http://docs.oracle.com/javaee/6/tutorial/doc/gijvh.html> [Žiūrėta 2015-05-03]
9. Brian Proffitt, What APIs Are And Why They're Important. Prieiga per internetą: <http://readwrite.com/2013/09/19/api-defined> [Žiūrėta 2015-05-03]
10. Android Studio Overview. Prieiga per internetą: <http://developer.android.com/tools/studio/index.html> [Žiūrėta 2015-05-03]
11. NetBeans IDE Features. Prieiga per internetą: <https://netbeans.org/features/php/index.html> [Žiūrėta 2015-05-03]
12. Volley: Easy, Fast Networking for Android. Prieiga per internetą: <https://developers.google.com/events/io/sessions/325304728> [Žiūrėta 2015-05-03]
13. kSOAP 2 About. Prieiga per internetą: <http://kobjects.org/ksoap2/index.html> [Žiūrėta 2015-05-03]
14. UMLet 13.2 Free UML Tool for Fast UML Diagrams. Prieiga per internetą: <http://www.umlet.com/> [Žiūrėta 2015-05-03]
15. Support Library. Prieiga per internetą: <http://developer.android.com/tools/support-library/index.html> [Žiūrėta 2015-05-03]

16. Nurzhan Nurseitov, Michael Paulson, Randall Reynolds, Clemente Izurieta, Comparison of JSON and XML Data Interchange Formats: A Case Study. 2009. Prieiga per internetą: <http://www.cs.montana.edu/izurieta/pubs/caine2009.pdf> [Žiūrėta 2015-05-03]

7. Anotacija

Darbo tema: „Duomenų mainų išskirstytosiose sistemose, naudojančiose Android OS, modeliai“.

Šių dienų pasaulio nebe galime įsivaizduoti be išmaniųjų mobiliųjų įrenginių, kaip įrenginių su Android operacine sistema. Šie įrenginiai dažnai naudojami ryšiui su nutolusiais įrenginiais (interneto serveriais, debesų kompiuterijos elementais ir pan.). Dėl šios priežasties iškyla duomenų mainų problemų, tokių kaip mobiliųjų Android OS įrenginių duomenų mainai išskirstytosiose sistemose ar su nutolusiais įrenginiais. Šių problemų sprendimo metodikos yra nagrinėjamos šiame darbe.

Anotation

Topic: „Data Exchange Models for Distributed Android-based Systems“.

In today's world we can no longer imagine without smart mobile devices, as devices with the Android operating system. These devices are often used to communicate with remote devices (web servers, cloud computing elements, and so on.). For this reason, there is a data exchange problems, such as mobile Android OS devices distributed data interchange systems or remote devices. The problem-solving techniques are examined in this work.