

ŠIAULIŲ UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Denas Pavlavičius
Informatikos specialybės II kurso dieninio skyriaus studentas

**INFORMACINĖS SISTEMOS INVENTORIAUS VALDYMO
SISTEMA**

THE SYSTEM FOR INFORMATION SYSTEM INVENTORY MANAGEMENT

MAGISTRO DARBAS

Darbo vadovė:
Doc. S. Turskienė

Recenzentas:
Lekt. dr. G. Felinskas

Šiauliai, 2012

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

Darbo autoriaus _____

(vardas, pavardė, parašas)

DARBO TIKSLAI IR UŽDAVINIAI

Tikslas – sukurti įmonės informacinės sistemos inventoriaus apskaitos, kiekvieno įrenginio migravimo sekimo ir atliktų darbų istorijos valdymo sistemą, kuri būtų nepriklausoma nuo operacinės sistemos, valdoma mobiliuoju įrenginiu.

Uždaviniai:

1. Išanalizuoti rinkoje esančias panašios paskirties programų sistemas.
2. Išanalizuoti ir parinkti programavimo įrankius sistemos kūrimui.
3. Suprojektuoti ir realizuoti informacinės sistemos inventoriaus valdymo sistemą.
4. Ištestuoti ir įdiegti sukurtą sistemą įmonėje.

Darbo vadovė

(vardas, pavardė, parašas)

TURINYS

1. Įvadas.....	6
2. Problemos apibrėžtis.....	7
3. Teorinė dalis.....	8
3.1 Trumpas kuriamos sistemos aprašymas.....	8
3.2 Inventoriaus apskaitos sistemų lyginamoji analizė.....	9
3.3 Inventoriaus apskaitos sistemų palyginimas.....	10
4. Sistemos projektas.....	12
4.1 Sistemos funkcionalumas.....	12
4.2 Duomenų srautai sistemoje.....	13
4.3 Sistemos komponentai.....	13
4.4 Sistemos duomenų bazės struktūra.....	14
4.5 Sistemos statinis vaizdas.....	15
4.6 Pagrindinių sistemos funkcijų detalizavimas.....	16
5. Programinių įrankių pasirinkimas.....	20
5.1 Integruotos programavimo aplinkos Java kalbai.....	20
5.1.1 NetBeans.....	21
5.1.2 Eclipse.....	23
5.2 Java aplikacijų serveriai.....	24
5.2.1 GlassFish.....	24
5.2.2 Tomcat.....	25
5.3 Moduliai Android aplikacijų kūrimui.....	25
5.3.1 NBAndroid.....	25
5.3.2 Android Developer Tools.....	26
5.3.3 NBAndroid ir ADT palyginimas.....	27
5.4 Programinių įrankių pasirinkimo apibendrinimas.....	27
6. Problemos sprendimas.....	29
6.1 XHTML ir XML duomenų formato taikymo mobiliuosiuose įrenginiuose tyrimas.....	29
6.2 Interneto srauto poreikio tyrimas ir jo rezultatai.....	30
6.3 Serverio apkrovos tyrimas ir jo rezultatai.....	31
6.4 Sistemos mobiliesiems įrenginiams architektūros pasirinkimas.....	33
7. Išvados.....	34
8. Literatūros sąrašas.....	35
9. Anotacija.....	38
10. Summary.....	39
11. Priedas nr. 1.....	40
12. Priedas nr. 2.....	41
13. Priedas nr. 3.....	43
14. Priedas nr. 4.....	45
15. Priedas nr. 5.....	46

1. ĮVADAS

Turint įmonėje ar organizacijoje tam tikrus svarbius išteklius ar objektus, visuomet kyla poreikis juos inventorizuoti, sekti jų judėjimą, kaitą, būsenas ir kitą su jais susijusią informaciją. Kasdieniniame gyvenime tokių objektų nėra daug. Įmonių veikloje turimas inventorių turi būti apskaitomas ne tik patogumui užtikrinti, bet yra būtinas tinkamam įmonės darbo organizavimui.

Šio darbo **tikslas** – sukurti įmonės informacinės sistemos inventoriaus apskaitos, kiekvieno įrenginio migravimo sekimo ir atliktų darbų istorijos valdymo sistemą, kuri būtų nepriklausoma nuo operacinės sistemos, valdoma mobiliuoju įrenginiu.

Tiksliui pasiekti keliami šie **uždaviniai**:

1. Išanalizuoti rinkoje esančias panašios paskirties programų sistemas.
2. Išanalizuoti ir parinkti programavimo įrankius sistemos kūrimui.
3. Suprojektuoti ir realizuoti informacinės sistemos inventoriaus valdymo sistemą.
4. Ištestuoti ir įdiegti sukurtą sistemą įmonėje.

2. PROBLEMOS APIBRĖŽTIS

Mobiliosios programėlės kūrimui galimi pasirinkti keli architektūros variantai: naudoti internetinio tipo sistemą, prieinamą per naršyklę, arba naudoti internetinę paslaugą, valdomą sukurta specialia programėle.

Koks formatas turėtų būti naudojamas, labai priklauso nuo konkrečios situacijos. Todėl įvairūs žiniatinklio servisų vykdymo charakteristikų tyrimai dažnai pateikia skirtingus rezultatus.

Šiame darbe keliama problema – koks architektūros variantas tinkamesnis mobiliam naudojimui.

3. TEORINĖ DALIS

Kiekviena įmonė ar organizacija turinti kokį nors inventorių, turi turėti ir inventoriaus sąrašą. Nesvarbu, ar jis elektroninėje formoje patalpintas kokioje nors duomenų bazėje, ar tiesiog ranka surašytas į sąsiuvinį. Inventoriui priklauso bet koks už įmonės pinigus pirktas daiktas: kėdė, kompiuteris ir kt. Neretai įmonės nori sekti su inventoriumi susijusius darbus: tarkime kėdės tvarkymas, kada ji buvo tvarkyta, kas būtent buvo tvarkyta. Jeigu įmonė turi mažai inventoriaus, tai problemų dažnai nekyla. Jeigu organizacijos inventoriaus sąrašė yra 500 daiktų iš kurių kiekvienas gali turėti po 50 įrašų, tai tokius duomenis tampa problematiška apdoroti be papildomų sistemų.

Šiame darbe bus projektuojama ir realizuojama organizacijos informacinės sistemos inventoriaus ir darbų susijusių su juo sąrašo valdymo sistema.

3.1 Trumpas kuriamos sistemos aprašymas

Sistemos veikimo principas labai panašus į sandėlio apskaitos sistemos veikimą. Pagrindinis skirtumas yra tas, kad sistemoje galima aprašyti ir vesti atliktų darbų žurnalą su kiekvienu inventoriaus elementu. Sistema turi saugoti visus organizacijoje esančius įrenginius susijusius su informacinėmis technologijomis. Kiekvienas įrenginys turi būti smulkiai aprašytas: jo serijinis numeris, inventoriaus numeris, specifiniai atributai. Taip pat kiekvienas įrenginys su savimi turi turėti ir sąrašą komentarų-darbų, kurio kiekvienas įrašas nusako kas, kada ir kas daryta su šiuo įrenginiu.

Sistemos pagrindinis rišamasis objektas turi būti darbo vieta, kuri yra nekintama ir yra priskirta tik vienai patalpai. Kiekviena darbo vieta gali turėti kelis įrenginius ir kelis darbuotojus. Tokiu principu realizavus sistemą, joje nesudėtingai galima perkelti darbo vietas (t. y. konkrečius darbuotojus su jiems priskirta įranga) iš vienos patalpos į kitą. Taip pat nesunkiai galima migruoti įrenginius ir darbuotojus tarp darbo vietų. Kiekvienas toks migravimas turi būti užfiksuotas sistemoje, kad vėliau būtų galima nustatyti kurioje patalpoje galima rasti tam tikrą įrenginį ir kas su juo dirba, sekti visą istoriją, kas, kada, iš kur ir į kur migravo. Sistemoje turi būti galimybė pagal darbuotojo pavardę sužinoti jam priskirto kompiuterio tinklo vardą, telefono numerį bei patalpą, kurioje jį galima rasti.

Panašių programų galima rasti pasaulinėje rinkoje. Dauguma jų yra neatsiejamos nuo kitų, tokių kaip buhalterinės ar sandėlio apskaitos sistemų, todėl tokios programos ar jų paketai yra brangūs ir ne visada atitinka visus konkrečios įmonės keliamus reikalavimus.

3.2 Inventoriaus apskaitos sistemų lyginamoji analizė

InFlow programa sukurta „Visual C#“ programavimo kalba panaudojant „.NET“ karkasą. Duomenų bazė realizuota „Microsoft SQL“ serveryje, o ataskaitos išduodamos panaudojus „Crystal Reports“ branduolį.

InFlow inventorinė programa – tai puikus produktas bet kokios įmonės inventoriaus apskaitai. Ši sistema yra draugiška vartotojams ir siūlo naudingų galimybių, tokių kaip serijinių numerių kontrolė, paieška pagal kategoriją arba vietą. Šios galimybės supaprastina kasdienes užduotis su įmonės inventoriumi [20].

Šia programa naudojami „B&B Lawncare“, „Casa Chacha Lda“, „Tri State Malted Waffle Inc“, „All Home Imports S.L“.

Inventoria programa sukurta „Microsoft Visual C++“ programavimo kalba. Inventoria – tai lanksti inventoriaus valdymo sistema. Ji siūlo puikias papildomas galimybes, tokias kaip pastabų ir nuotraukų prisegimą prie kiekvieno inventoriaus elemento. Taip pat šia programa gali naudotis daugiau nei vienas vartotojas [21].

iMagic Inventory programa sukurta „Visual Basic 6“ programavimo kalba. iMagic Inventory – programa kuri tikrai padės jums valdyti savo inventorių. Ji išsiskiria tuo, kad galima valdyti vartotojų ir tiekėjų duomenų bazes [22].

Inventory Organizer Deluxe programa sukurta „Borland Delphi“ programavimo kalba. Inventory Organizer Deluxe – skirta verslo bei namų naudojimui. Ja galima valdyti verslo kontaktus ir sandėlio informaciją taip pat paprastai kaip savo namų inventorių [23].

Small Business Inventory Control programa sukurta „Microsoft Visual C++“ programavimo kalba. Duomenų bazė realizuota „Microsoft Access“ sistemoje. Small Business Inventory Control – programa skirta nedideliame verslui. Ji sugeba identifikuoti bet kokią daiktą pagal bar-kodą ir jį administruoti [24].

Inventory Pro programa sukurta „Visual C#“ programavimo kalba panaudojant „.NET“ karkasą. Duomenų bazė realizuota „Microsoft SQL“ serveryje, o ataskaitos išduodamos panaudojus „Crystal Reports“ branduolį. Inventory Pro – nedidelės įmonės inventoriaus apskaitos sistema, kuri gali būti valdoma tik vieno vartotojo vienu metu [25].

Šia programa naudojami „EMPR Holdings“.

WASP Inventory Control – tai pakankamai galinga inventoriaus apskaitos sistema. Joje galima sukurti keletą sandėlių ar patalpų, kuriose laikomos tam tikros inventoriaus dalys. Sukurti ir aprašyti kiekvieną daiktą esantį organizacijoje, formuoti užsakymo ir nurašymo ataskaitas, pilnas inventorines ataskaitas. Kartu su šia sistema galima įsigyti papildomus suderintus įrenginius, tokius kaip delninius kompiuterius, bar-kodų spausdintuvus [26.].

Šia sistema naudojasi „Maxim Integrated Products“, „Amarillo National Bank“, „Process Control Outlet“.

Altiris Inventory Solution sistema sukurta panaudojus „ASP.NET“ karkasą, o duomenų bazė realizuota „Microsoft SQL“ serveryje. „Altiris Inventory Solution“ tai Symantec kūrinys pateikiamas kartu su pilnu sistemos palaikymu. Ši sistema yra arčiausiai šiame darbe keliamos problemos sprendimo, todėl kad ji orientuota į informacines sistemas. Programa turi savyje sukurtus šablonus programinėms įrangoms, serveriams, staliniams bei nešiojamiems kompiuteriams, monitoriams, spausdintuvams ir kitoms informacinės sistemos (IS) šakoms aprašyti [27.].

Šia sistema naudojasi „Messer Information Systems“, „Lone Star College System“,

3.3 Inventoriaus apskaitos sistemų palyginimas

Apibendrinant visas analizuotas inventoriaus apskaitos sistemas, jų pagrindines savybes galima pateikti lentelėje nr. 1.

Lentelė nr. 1. Inventoriaus apskaitos sistemų savybių palyginimas

Analizuojama sistema / Analizuojama savybė	inFlow	Inventoria	iMagic Inventory	Inventory Organizer Deluxe	Small Business Inventory Control	Inventory Pro	WASP Inventory Control	Altiris Inventory Solution
Realaus laiko inventoriaus statusas	+	+	+		+	+	+	+
Elementų istorija	+	+	+	+	+	+	+	+
Kategorizavimas	+	+	+	+	+	+	+	+
Inventoriaus informacijos importas/eksportas	+	+	+	+	+		+	+
Paieška pagal vietą	+	+	+	+	+	+	+	
Bar-kodų skanavimas	+	+	+	+	+		+	+
Pastabų rašymas	+	+	+		+		+	
Vartotojų duomenų bazė	+	+	+	+	+	+	+	+
Tiekėjų duomenų bazė	+	+	+	+	+	+	+	+
Ataskaitų generavimas	+			+			+	+

Pasiekimas per Web	+	+						
Suderinta su Windows OS	+	+	+	+	+	+	+	+
Suderinta su MAC OS						+	+	+
Programavimo kalba	C#	C++	VB 6	Delphi	C++	C#	C#	C#
Kaina Lt	780	210	650	195	260	390	2070	1780

Visos šio pobūdžio sistemos gali ne pilnai atitikti anksčiau aptartą sistemos koncepciją. Deja nei viena iš jų neatitinka pagrindinio reikalavimo – darbų, susijusių su kiekvienu įrenginiu, istorijos bei įrenginių migravimo sekimo.

Bet kokia inventoriaus apskaitos sistema gali suskirstyti inventorių į skirtingas kategorijas, jas aprašyti, saugoti elementų skaičių taip pat sekti atsiradimo ir panaikinimo datas. Orientuojantis į informacines technologijas tokių funkcijų nepakanka. Bendroje inventoriaus apskaitos sistemoje paprastas kompiuteris gali būti kaip galutinis elementas, o realiai tai yra tik inventoriaus kategorija, kuri gali turėti daug skirtingų ir tarpusavyje nesusijusių elementų. Dėl šios priežasties atsiranda reikiamybė skirstyti inventorių į daugiau kategorijų ir įrenginius sekti ne kaip bendrą kategoriją, o kiekvieną asmeniškai.

Dauguma informacinės sistemos inventoriaus įrenginių yra sudaryti iš sudedamųjų dalių arba turi specifinius parametrus todėl didelis palengvinimas yra galimybė greitai apie juos sužinoti neardant arba neieškant specialios dokumentacijos.

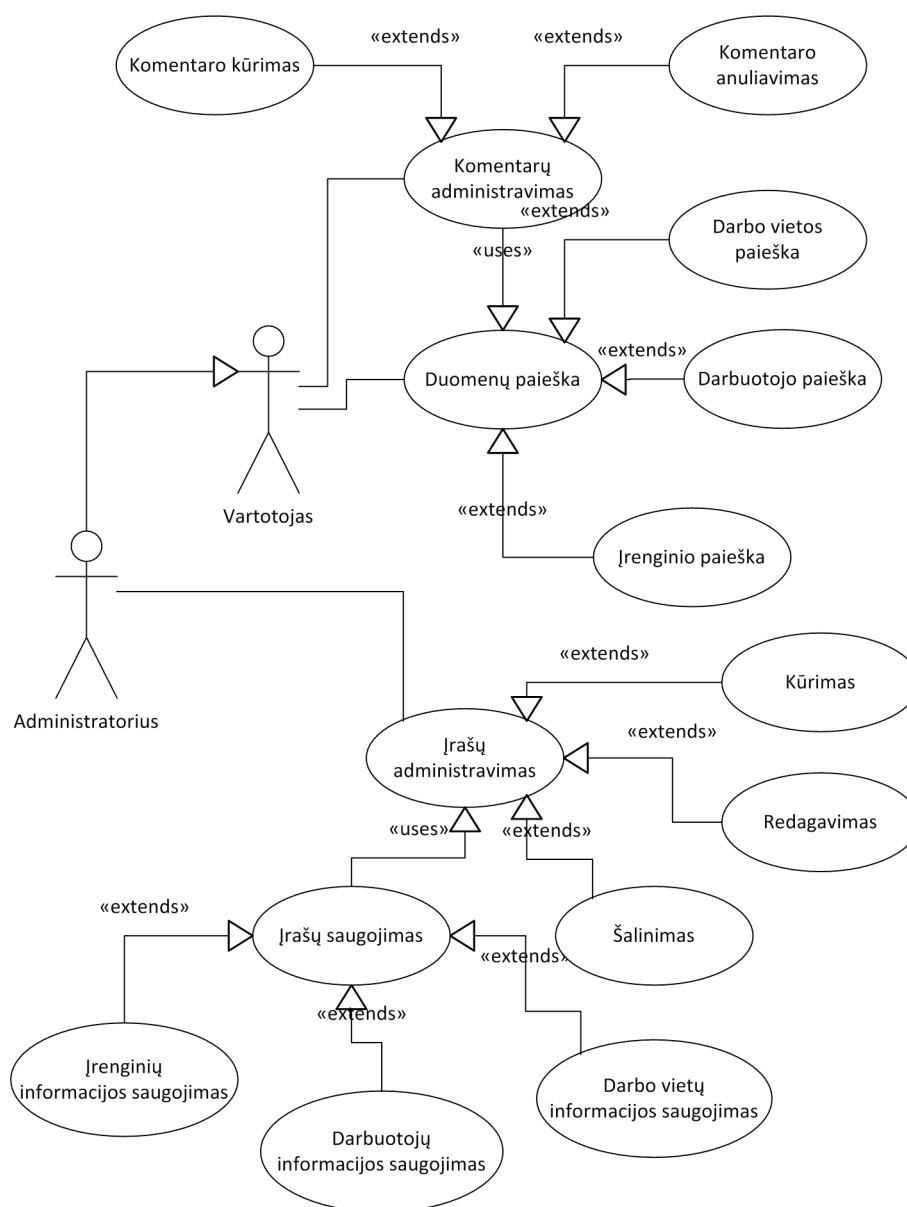
Kadangi įrenginiai gali būti sudaryti iš kelių komponentų tai laikui einant jie gali būti keičiami ar tvarkomi. Esant tokiems galimiems veiksams atsiranda reikiamybė registruoti visus su įrenginiu atliktus darbus, o jeigu su įrenginiais atliekami kažkokie darbai jo naudojimo periodu, tai būtina žinoti kur tiksliai jie randasi.

Tokia aprašyta sistema yra labiau orientuota į didesnes įmones ar organizacijas, kurių naudojami informacinės sistemos įrenginiai gali būti išdėstyti skirtingose patalpose ir visas inventorių yra sudarytas iš daugiau nei dviejų kategorijų, kurios gali turėti daugiau nei po kelis įrenginius.

4. SISTEMOS PROJEKTAS

4.1 Sistemos funkcionalumas

Bendraujant su vartotoju/užsakovu buvo nustatytos pagrindinės sistemai keliamos funkcijos, kurios gali būti apibendrinamos panaudos atvejų diagrama (žr. Pav. 1). Ši diagrama atspindi programos funkcionalumui keliamus reikalavimus.



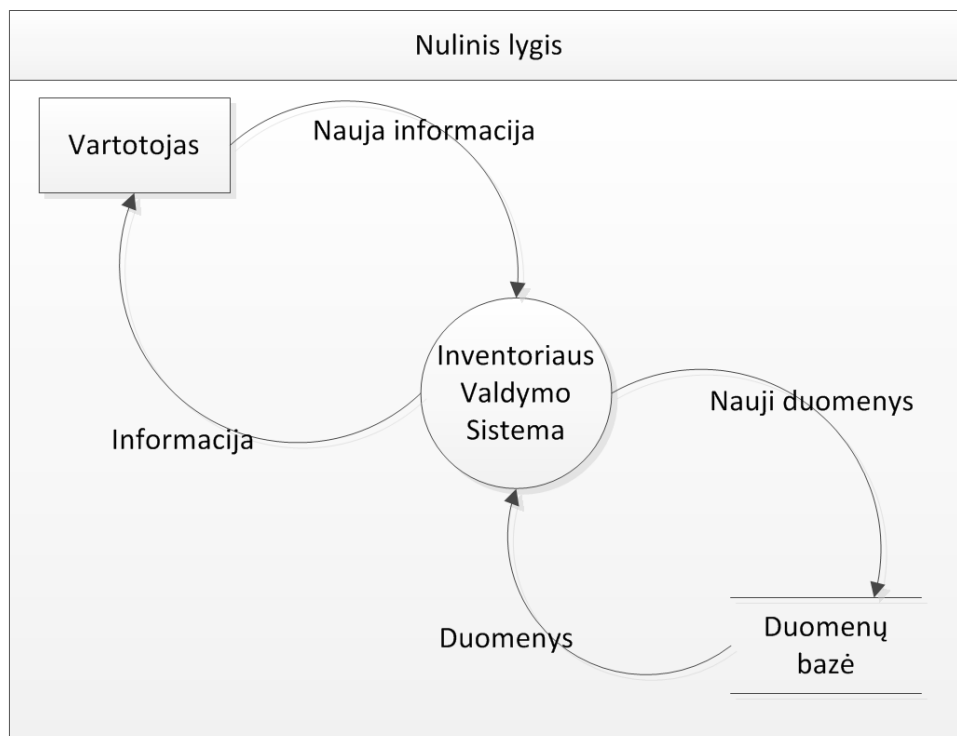
Pav 1 Sistemos teikiamos funkcijos ir jų priklausomybė

Sistemoje išskirtos dvi vartotojų rolės: administratorius ir paprastas vartotojas. Paprastas vartotojas gali atlikti darbo vietų, darbuotojų ir įrenginių paiešką bei juose kurti arba anuluoti komentarus. Administratorius – paveldi paprasto vartotojo funkcijas ir papildomai gali

administruoti visus sistemos duomenis susijusius su įrenginiais, darbo vietomis ir darbuotojais, bei papildomus atributus.

4.2 Duomenų srautai sistemoje

Panaudos atvejų diagrama nusako tik vartotojo atliekamus veiksmus sistemoje, tačiau neskiria dėmesio duomenų srautams sistemoje. Šiai daliai atvaizduoti naudojama duomenų srautų diagrama (toliau DSD). Pav. 2 pateikiama žemiausio (nulinio) lygio DSD, kurioje atvaizduojamas apibendrintas sistemos ir išorinių veiksnių apsikeitimas duomenimis.

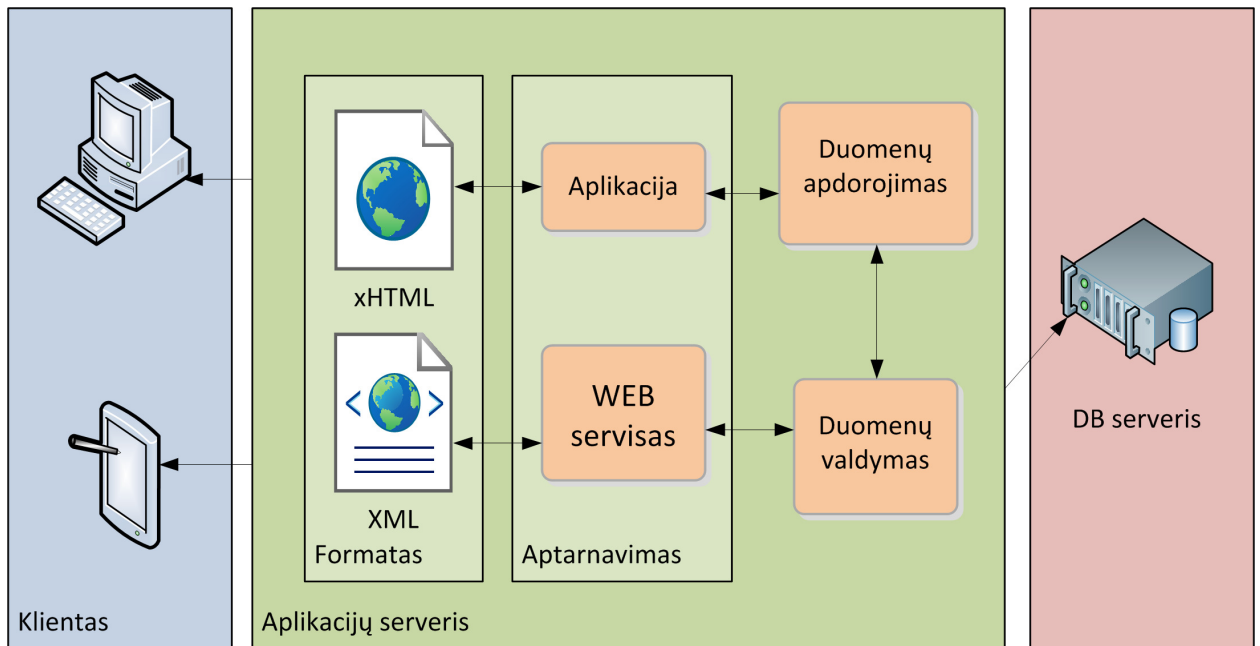


Pav 2 Sistemos nulinio lygmens duomenų srautų diagrama

Ši diagrama parodo, kad kuriama sistema turės tarnauti kaip tarpininkė tarp vartotojo ir inventoriaus duomenų bazės. Vartotojai joje pateiks naują informaciją (inventoriaus duomenis, veiksmų su juo istoriją), kuris bus saugoma duomenų bazėje, o esant poreikiui iš duomenų bazės suprantamai pateikiama vartotojui.

4.3 Sistemos komponentai

Sistemos plečiamumui padidinti, jos architektūra sudaryta iš atskirų komponentų. Siekiant aiškiau pavaizduoti sistemą sudarančius komponentus ir jų sąveiką, sekančioje diagramoje atvaizduojamas kelias nuo vartotojo iki duomenų per atskirus sistemos modulius (žr. Pav. 3).



Pav 3 Vartotojo ir DB siejantys sistemos komponentai ir jų sąveika

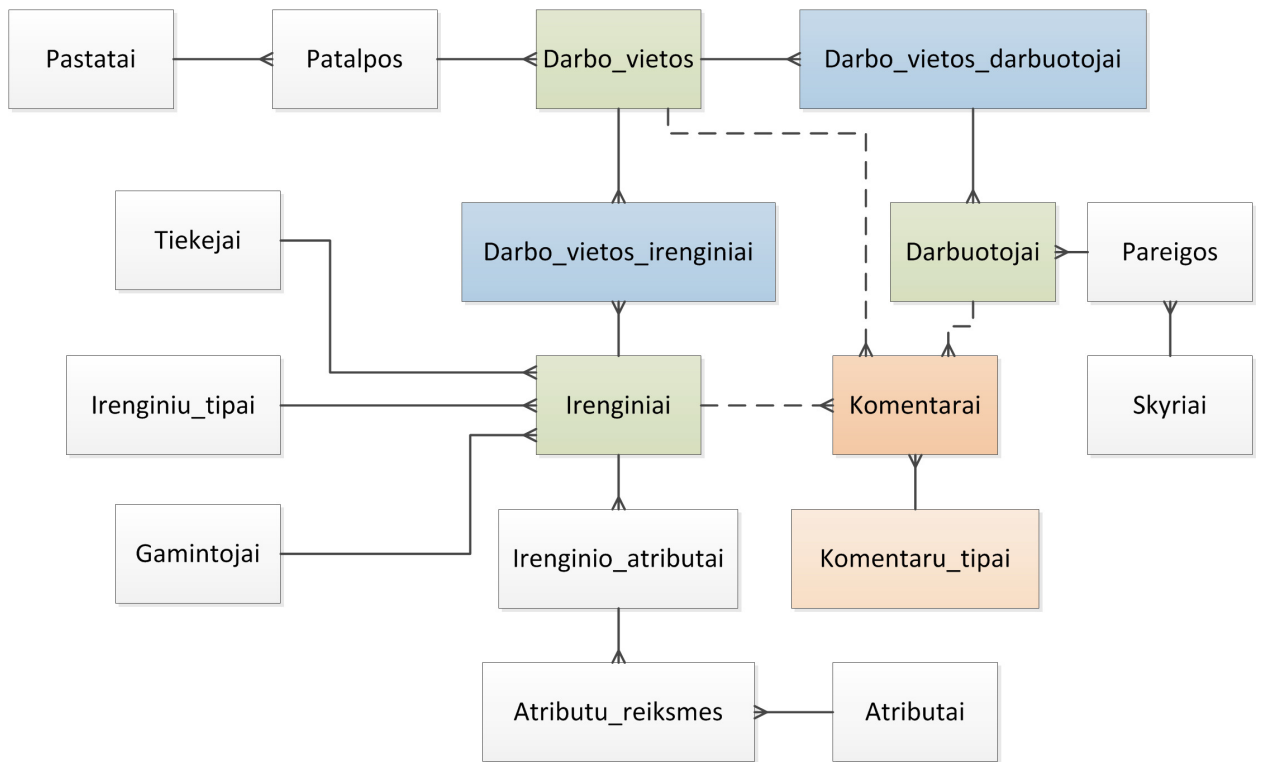
Vartotojas sistemą valdyti gali arba kompiuteriu per naršyklę, arba išmaniuoju telefonu naudojant WEB servisas. Aplikacijų serveris gavęs užklausą vienu iš dviejų formatų (xHTML arba XML) elgiasi atitinkamai: arba perduoda užklausos duomenis aplikacijai, kuri savo ruožtu juos apdoroja ir per duomenų valdymo paprogramę kreipiasi į duomenų bazę, arba per WEB servisą kreipiasi tiesiai į duomenų kontrolės paprogramę.

Duomenys vartotojui perduodami tokiu pat būdu priklausomai nuo to, kuriuo keliu vartotojas įvykdė užklausą.

4.4 Sistemos duomenų bazės struktūra

Projektuojant sistemos reliacinę duomenų bazę išskiriamos keturios duomenų bazės lentelių kategorijos skirtingo tipo objektams atspindėti (žr. Pav. 4):

- Pagrindiniai objektai (žalia spalva);
- Pagalbiniai aprašomieji objektai (balta spalva);
- Esminiai objektai (oranžinė spalva);
- Ryšio atvaizdavimo objektai (melsva spalva);



Pav 4 Supaprastinta sistemos duomenų bazės struktūra

Pagrindiniai objektai yra trys: darbo vietos, įrenginiai ir darbuotojai. Jie atspindi esminę sistemoje saugomą informaciją.

Panaudojant ryšio atvaizdavimo objektus „Darbo_vietos_darbuotojai“ ir „Darbo_vietos_irenginiai“ kiekviena darbo vieta gali būti susiejama su joje esančiais įrenginiais ir joje dirbančiais vartotojais. Tai leis realizuoti sąsajas tarp darbuotojo ir jo galimų naudoti įrenginių sąrašo.

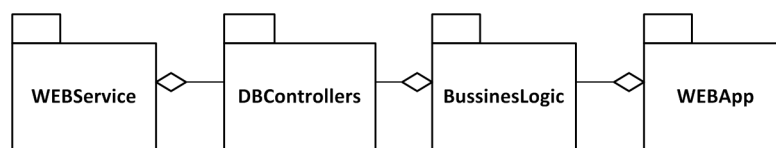
Visi trys pagrindiniai objektai turi pagalbinius aprašomuosius objektus. Jie skirti duomenų bazės normalizavimui ir kitų sistemoje svarbių duomenų atvaizdavimui.

Esminis objektas yra komentaras, kuriuo įgyvendinamas sistemos privalumas – kiekvieno pagrindinio objekto komentavimo funkcija. Kaip matoma iš supaprastintos duomenų bazės diagramos, komentarai siejasi su įrenginių, darbo vietų ir darbuotojų lentelėmis punktyrine linija. Tai reiškia, kad tarp tų objektų nėra griežto ryšio, t.y. priklausomai nuo situacijos ryšys gali būti arba ne. Taip yra todėl, kad komentarų lentelėje saugomas objekto tipas (darbo vieta, darbuotojas ar įrenginys) ir objekto identifikacijos numeris. Taip galima vienoje duomenų lentelėje saugoti kelių skirtingų objektų komentarus ir tam tikslui įgyvendinti nereikalingos ryšių lentelės.

4.5 Sistemos statinis vaizdas

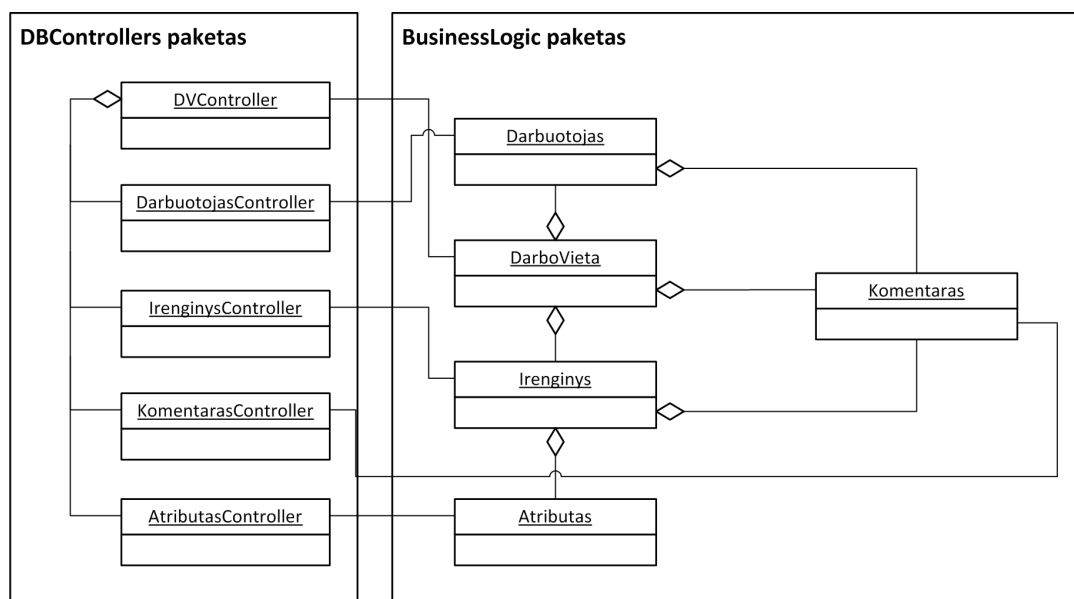
Atsižvelgiant į anksčiau pateiktą sistemos architektūros funkcionalumo diagramą galima išskirti 4 sistemos paketus: WEBSERVICE, DBCONTROLLERS, BussinesLogic, WEBApp. Kaip

parodyta paketų diagramoje (žr. Pav. 5), WEBService paketas naudoja DBControllers paketą, o WEBApp paketas naudoja BussinesLogic, kuris savo ruožtu naudoja DBController paketą.



Pav 5 Sistemos paketai ir ryšiai tarp jų

Pav. 6 pateikiamas BussinesLogic ir DBController paketų statinis vaizdas, kuriame matomas ryšys tarp šių paketų elementų.



Pav 6 DBController ir BussinesLogic paketus sudarančios klasės ir jų ryšiai

WEBService pakete yra tik viena klasė, kurioje realizuojamos pagrindinės WEB serviso funkcijos, kur kiekviena iš jų kreipiasi į atitinkamus DBController paketo elementus.

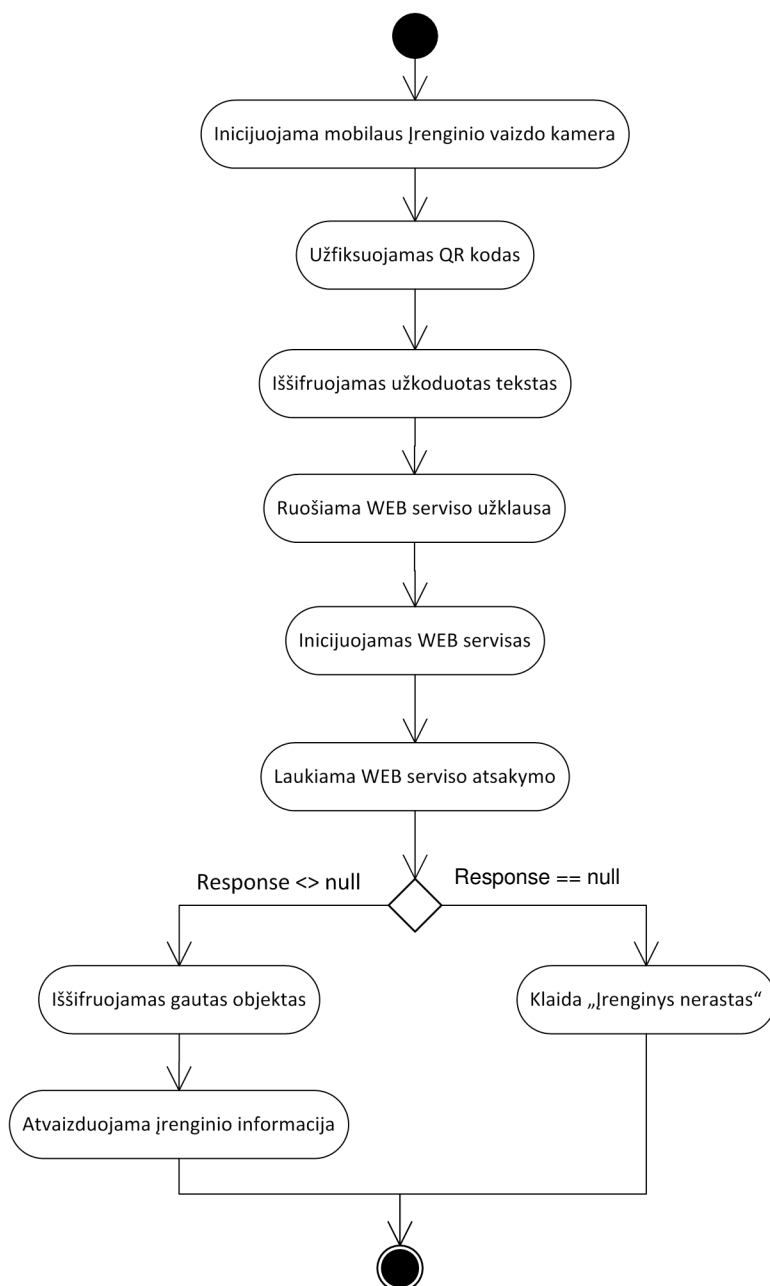
WEBApp pakete realizuoti kliento pusės puslapiai ir naudojamos funkcijos, kurios kreipiasi į BussinesLogic paketo elementus.

4.6 Pagrindinių sistemos funkcijų detalizavimas

Sistemoje kuriamos duomenų administravimui skirtos funkcijos, kurios dirba su duomenų baze ir atvaizduoja pasirinktus duomenis. Toliau pateikiamos dviejų pagrindinių sistemos funkcijų naudojant Android aplikaciją veiklos diagramos.

Pirmoji veiklos diagrama atspindi įrenginio paieškos funkcijos veikimo principą (žr. Pav 7). Iškvietus šią funkciją inicijuojama mobilaus įrenginio vaizdo kamera, naudojant ZXing QR kodų nuskaitymo biblioteką. Šios bibliotekos pagalba kamera užfiksuoja QR kodą ir jį paverčia į tekstą. Gautame tekste specialiu formatu yra užrašytas įrenginio identifikacijos numeris. Panaudojus gautą identifikacijos numerį paruošiama WEB serviso užklausa ir

inicijuojamas WEB servisas. Sulaukus serverio atsakymo tikrinama ar gautas objektas nėra tuščias, jei taip tai sistema praneša, jog nurodytas įrenginys nerastas. Jeigu gautas objektas nėra tuščias, tokiu atveju jis yra iššifruojamas ir atvaizduojamas ekrane.



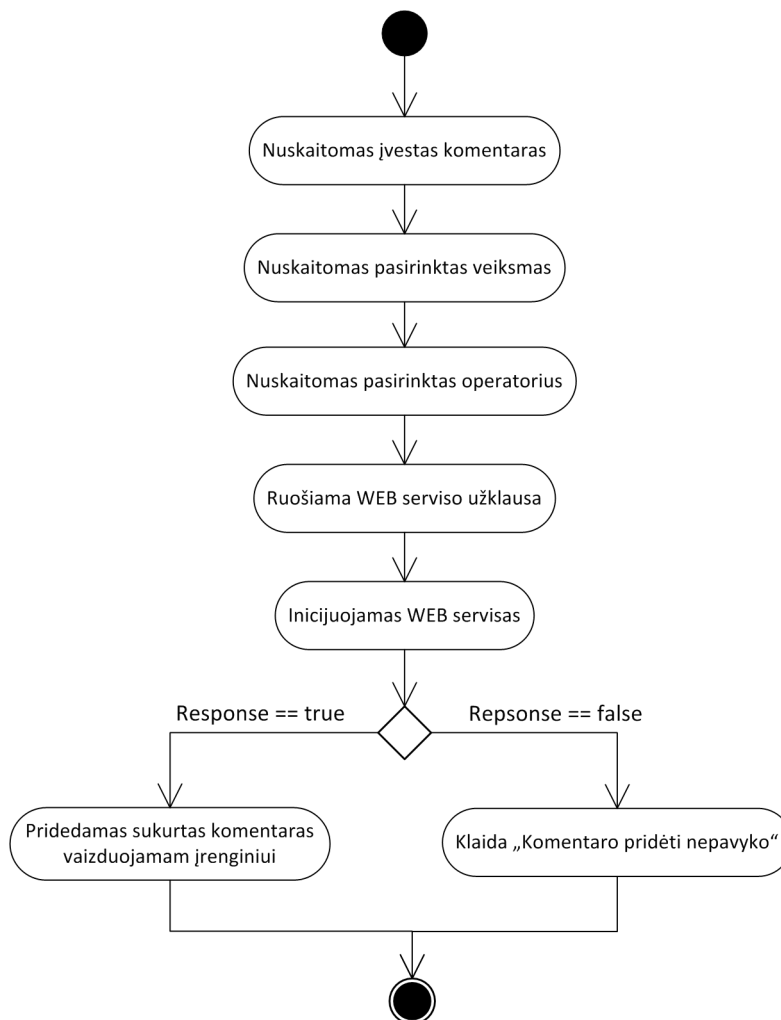
Pav 7 Įrenginio paieškos funkcijos veiklos diagrama

Kita veiklos diagrama atvaizduoja komentaro įvedimo funkcijos veikimo principą (žr. Pav 8). Komentaro įvedimo lange matomi trys laukai:

1. Komentaro veiksmo parinkimo laukas
2. Komentarą įvedusio operatoriaus parinkimo laukas
3. Komentaro įvedimo laukas

Kuomet sistemoje suvesti visi duomenys, inicijuojama komentaro įvedimo funkcija, kuri nuskaito visus duomenis ir su jais paruošia WEB servisą kreipimuisi. Kai WEB serviso

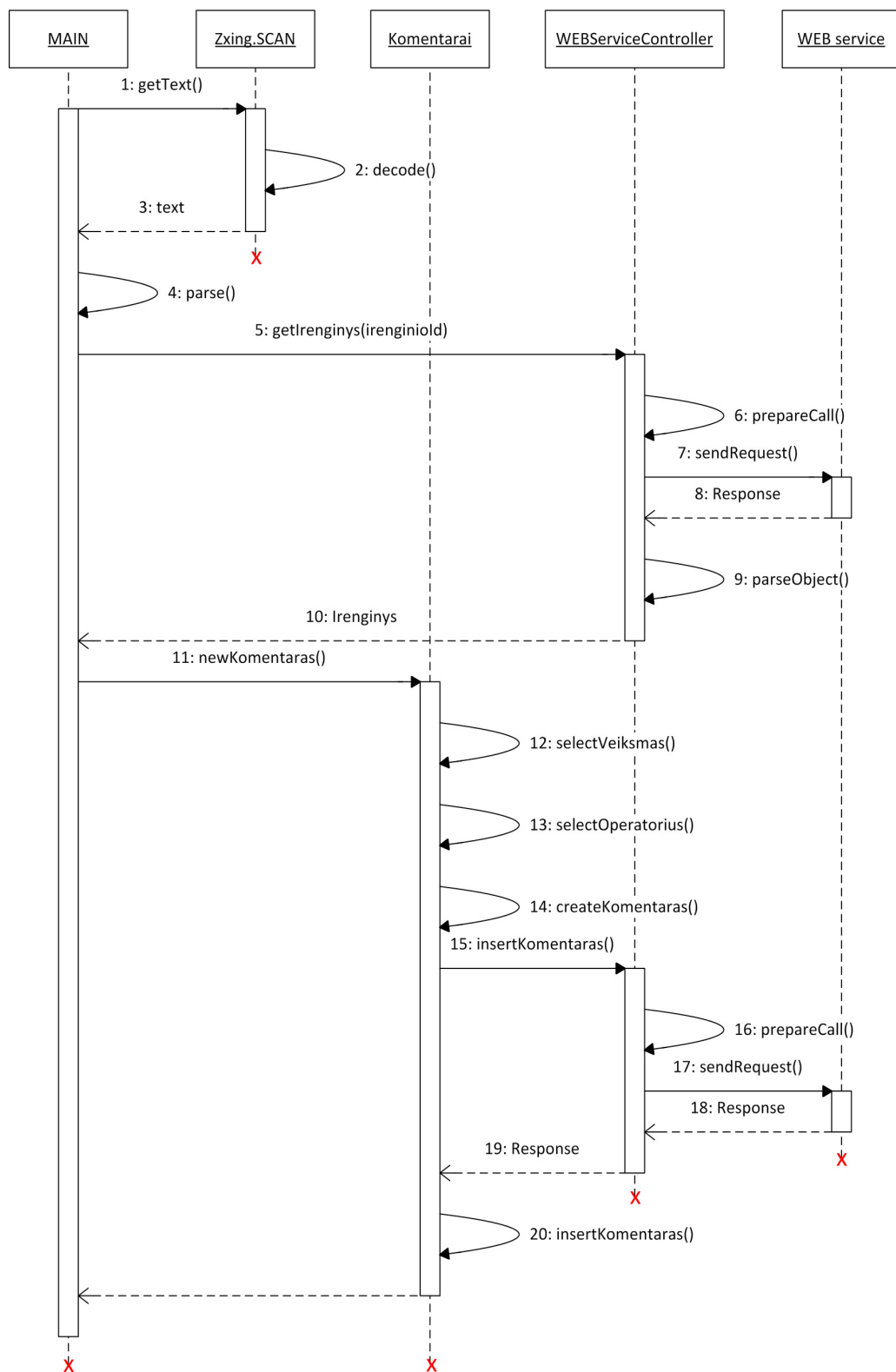
užklausa paruošta tada inicijuojamas kreipimasis į serverį ir laukiama atsakymo. Serverio atsakymas gali turėti dvi reikšmes: *true* ir *false*. Jeigu serverio atsakymas yra *false*, tai reiškia, jog serveriui nepavyko į duomenų bazę įrašyti komentaro ir vartotojui nurodoma klaida. Jeigu serverio atsakymas yra *true*, tai reiškia, jog serveriui pavyko įrašyti komentarą į duomenų bazę ir jis yra pridodamas prie kitų objekte išsaugotų komentarų.



Pav 8 Komentaro įrašymo funkcijos veiklos diagrama

Toliau pateikiama detalizuota šių dviejų funkcijų sekos diagrama (žr. Pav 9), kurioje matoma kaip einant laikui vykdomi kreipimaisi į tam tikras funkcijas. Paleidus aplikaciją yra inicijuojamas *MAIN* objektas iš kurio vėliau kreipiamasi į kitus objektus ir pateikiama informacija. Pradėjus darbą inicijuojama mobilus įrenginio kamera iš kurios nuskanavus QR kodą gaunamas tekstas. Tada jis yra iškoduojamas ir atpažįstamas įrenginio identifikacijos numeris, kuris perduodamas *WEBSERVICECONTROLLER* objektui. Šis objektas inicijuoja WEB serviso užklausa su nurodytais parametrais ir gauna iš serverio atsakymą kuris yra užšifruotas objektas. Gautas objektas iššifruojamas ir perduodamas atgal *MAIN* objektui. *MAIN* klasė gauna jau *Įrenginio* tipo objektą ir su juo galima toliau dirbti. Šiuo atveju toliau yra kviečiama *Komentarai* objekto naujo komentaro įvedimo funkcija. Naujame lange parenkami parametrai

(komentarų veiksmas, operatorius, komentaras) ir inicijuojamas *WEBSERVICECONTROLLER* objektas, kuris paruošia WEB serviso užklausą su nurodytais parametrais ir ją perduoda *WEB service* objektui. Šis savo ruožtu grąžina objektą, ir tada *Komentarai* klase įtraukia naują komentarą į sąrašą. Atlikus šiuos veiksmus grįžtama į pagrindinį langą.



Pav 9 Pagrindinių Android aplikacijos funkcijų sekų diagrama

5. PROGRAMINIŲ ĮRANKIŲ PASIRINKIMAS

Sistemai keliamas reikalavimas, kad ji veiktų skirtingose operacinėse sistemose ir taikoma ir mobiliuosiuose įrenginiuose. Tai sąlygojo, kad buvo nuspręsta pasirinkti Java programavimo kalbą, dėl jos nepriklausomumo nuo operacinės sistemos ir Android sistemos populiarumo mobiliuosiuose įrenginiuose.

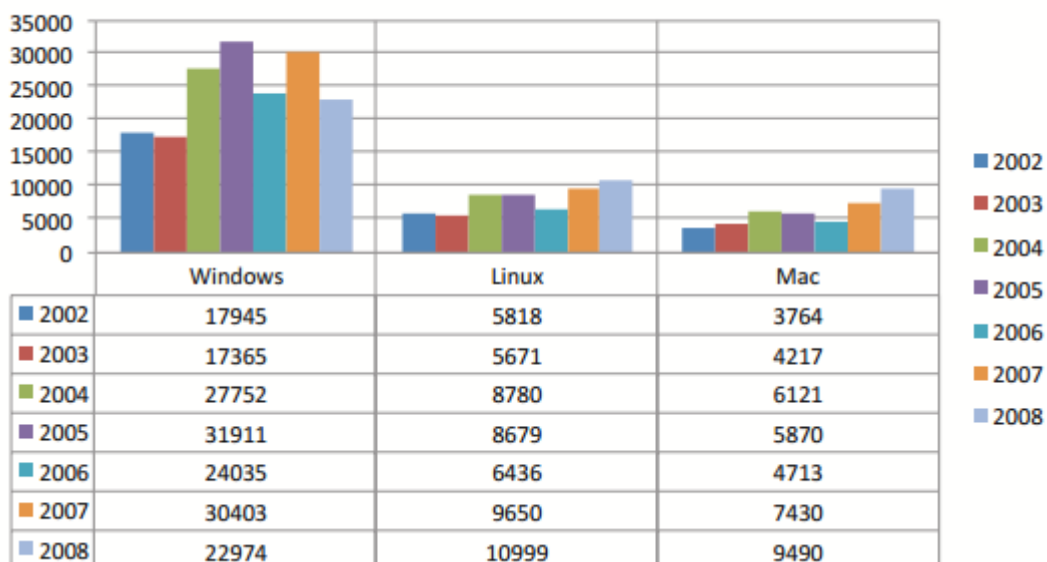
5.1 Integruotos programavimo aplinkos Java kalbai

Programinio kodo rašymui Java, kaip ir kitos populiaros programavimo kalbos, turi ne vieną programavimo įrankį. Vieniems naudoti reikia įsigyti licenciją, o kiti yra atvirojo kodo ir prieinami kiekvienam. Du pagrindiniai atviro kodo įrankiai yra „NetBeans“ ir „Eclipse“, kurie jau daugiau nei 10 metų lenktyniauja tarpusavyje dėl geresniojo vardo. Iš esmės atsižvelgiant tik į Java programavimo kalbos projektus šie du įrankiai skiriasi minimaliai. Tiek vienu, tiek kitu galima kurti visus Java projektus nuo paprasto apleto iki didelės įmonės valdymo sistemos.

Skirtumai pastebimi tada, kai susiduriama su sunkumais naudojant vieną iš šių įrankių. Kalifornijos universiteto, kompiuterijos katedros studentai Xinlei (Oscar) Wang, Eilwoo Baik, Premkumar T. Devanbu atliko tyrimą [28.], kurio metu buvo išanalizuoti abiejų sistemų klaidų pranešimai ir jų pašalinimas trijose pagrindinėse operacinėse sistemose (Windows, Linux ir MAC OS).

„NetBeans“ ir „Eclipse“ sukurtos Java programavimo kalba ir naudoja „Bugzilla“, kaip darbo su klaidomis įrankį. Išanalizavus visus klaidų pranešimus nuo 2002 metų, buvo prieita išvados, kad „Eclipse“ naudojama MAC OS operacinėje sistemoje yra stabiliausia ir greičiausiai atnaujinama atsiradus naujoms klaidoms [28.].

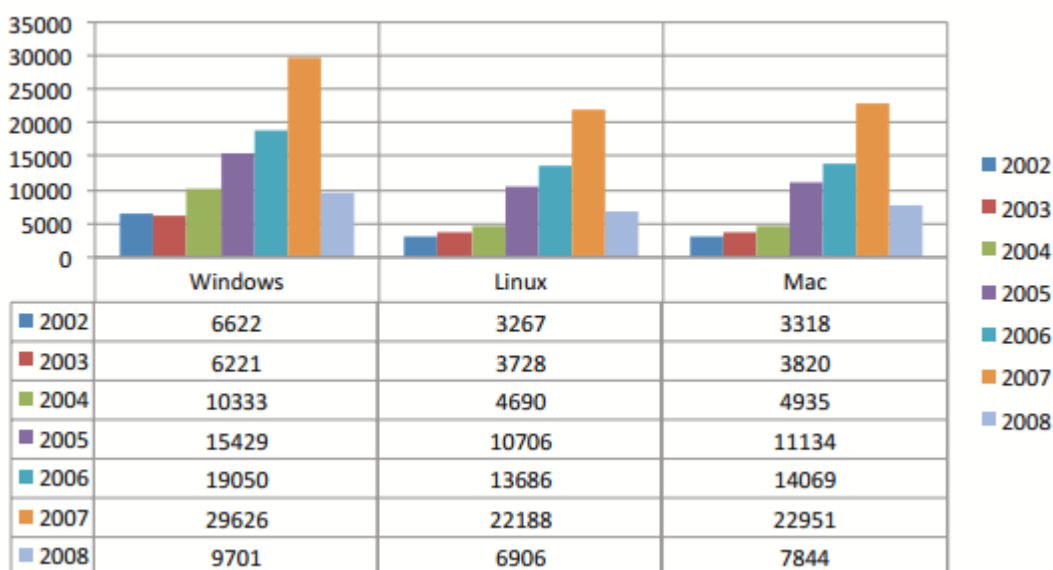
Number of Bugs for Eclipse



Pav 10 Eclipse klaidos pranešimų kiekio skirtingose operacinėse sistemose suvestinė

„Eclipse“ platformos klaidų skaičius ypatingai skiriasi Windows ir MAC OS aplinkose, beveik 10 kartų daugiau klaidų pasitaiko Windows aplinkoje, o „NetBeans“ klaidų skaičius skirtingose aplinkose beveik nekinta.

Number of Bugs for Netbeans



Pav 11 NetBeans klaidos pranešimų kiekio skirtingose operacinėse sistemose suvestinė

Nors tyrėjai teigia, kad išanalizavus duomenis „Eclipse“ galima laikyti stabilesne platforma, bet atsižvelgus tik į paskutinių metų duomenis, skirtumas tarp šių dviejų platformų yra palyginus minimalus. Be to, klaidų skaičius priklauso ir nuo pačių vartotojų. Galima daryti prielaidą, kad Windows aplinkoje su NetBeans įrankiu dirbantys vartotojai yra jautresni klaidoms, nei MAC OS aplinkoje dirbantys su „Eclipse“.

5.1.1 NetBeans

Netbeans yra nemokama integruota programavimo aplinka (angl. Integrated Developing Environment - IDE), kurti visų tipų Java aplikacijas (Java SE kartu su JavaFX, Java ME, web, EJB ir mobilias aplikacijas), JavaScript, PHP, "Groovy, C, C + + programas ar aplikacijas. NetBeans IDE yra parašyta "Java" ir gali veikti Windows, MAC OS, Linux, Solaris ir kitose platformose, kuriose palaikoma Java virtuali mašina (JVM).

NetBeans IDE interfeisas yra rinkinys nešiojamų langų išdėliojamų pagal „naršymas kairėje, redagavimas dešinėje“ formatą. Naršymo sritis sudaryta iš tipinio projektų naršymo, failinės sistemos navigacijos ir duomenų bazių peržiūros langų. Prie šio trejetuko NetBeans prideda ir web servisų naršyklę.

Web servisų naršyklė jau yra užpildyta punktais atitinkančiais Amazon, Facebook, Google ir kitus populiarius servisu. Šios naršyklės pagalba galima peržiūrėti kiekvieno iš jų teikiamas funkcijas ir jų argumentus. Taip pat didelis palengvinimas yra tas, kad nereikia rašyti programos kodo kreipiantis į norimą web serviso funkciją, užtenka ją pasirinkti iš sąrašo ir nutempti į programos kodo langą, o NetBeans visą kitą darbą padaro už jus (tik JAVA ir PHP kalbose).

NetBeans kodo redaktorius turi visus kodo užbaigimo pagrindus. Jis uždarys bet kokius skliaustelius (riestinius, laužtinius ar paprastus), ir taip pat pasiūlys objektų, funkcijų ar kintamųjų sąrašą [1.].

Plačiau apie NetBeans produktą ir jo kūrimo eigą galima paskaityti <http://netbeans.org/about/index.html>

Integruoti NetBeans moduliai

NetBeans Profiler – tai įrankis skirtas Java aplikacijų stebėjimui. Jis padeda programuotojams nustatyti atminties nutekėjimus ir optimizuoti aplikacijos vykdymo greitį. Šis įrankis gali pateikti netgi labai didelių ir kompleksinių aplikacijų vykdymo informaciją. Taip pat juo galima nustatyti ir išmatuoti tam tikrų vykdymo taškų laiką [2.].

Grafinės vartotojo sąsajos kūrimo modulis paprastai žinomas kaip Matisse projektas, suteikia programuotojams galimybę kurti Swing vartotojo sąsajas tiesiog komponentų išrinkimo ir padėjimo būdu [3.].

NetBeans JavaScript redaktorius suteikia galimybę redaguoti JavaScript, Ajax ir CSS išeities kodus.

JavaScript redaktorius paryškina sintaksę, užbaigia kodą, gali parašyti JavaScript klasės šabloną, iš esamo šablono sukuria Ajax iškvietimą ir gali nustatyti naršyklės palaikymą parašytam kodui.

CSS redaktorius taip pat gali užbaigti rašomą kodą, sugeneruoja pavyzdinį vaizdą pagal parašytą kodą ir atvaizduoja css failų struktūrą [4.].

Papildomi NetBeans moduliai

Web and Java EE modulis suteikia programuotojui visas galimybes kurti aplikacijas pagal Java EE 6 standartus į kuriuos įeina naujas Java EE 6 Web Profile, Enterprise Java Beans (EJBs), servletai, Java Persistence API, web servaisi ir anotacijos. Šis modulis taip pat palaiko ir JSF 2.0 (Facelets), JavaServer Pages (JSP), Hibernate, Spring ir Struts karkasus. Taip pat palaikomos ir Java EE5 bei J2EE 1.4 platformos. Kartu su šiuo moduliu pridedami ir GlassFish bei Apache Tomcat aplikacijų serveriai [5.].

Java ME modulis – tai įrankis, suteikiantis programuotojui galimybes kurti aplikacijas mobiliems įrenginiams, paprastai mobiliems telefonams, bet šis modulis taip pat turi galimybes kurti ir pradinio lygio delniniams kompiuteriams (PDA) [6.].

Su **PHP** moduliu NetBeans tampa pilnaverčiu PHP kodo redaktorium su visomis kodo užbaigimo bei sintaksės klaidų taisymo galimybėmis [7.].

Pilnas NetBeans IDE

Taip pat yra ir pilna NetBeans IDE versija, kurioje yra integruoti visi moduliai [8.]:

- Pagrindinės NetBeans IDE funkcijos
- Java SE, JavaFX
- Web ir Java EE
- Java ME
- C/C++
- PHP
- GlassFish
- Apache Tomcat

5.1.2 Eclipse

Eclipse – tai atvirojo kodo produktai, kuriuos kuria bendruomenė. Šios bendruomenės valdytojas yra ne pelno siekianti organizacija Eclipse Foundation.

Kita vertus kalbant apie programinę įrangą Eclipse dažniausiai vadinama Eclipse programų kūrimo įrankių rinkinys (SDK), kuris susideda iš Eclipse platformos, Java programavimo įrankių ir įskiepių kūrimo aplinkos. Eclipse platform yra daugiakalbė programinės įrangos kūrimo aplinka, kurią sudaro integruota programavimo aplinka ir įskiepių sistema. Esama sukurta daug įskiepių, kurių pagalba Eclipse programavimo aplinka galima kurti aplikacijas skirtingomis programavimo kalbomis, tokiomis kaip Ada, C, C++, COBOL, Java, Perl, PHP, Python, R, Ruby (Ruby on Rails), Scala, Clojure, Groovy ir Scheme.

Kadangi Eclipse SDK yra parašyta pagrinde Java kalba, tai ji labiau skirta ir pritaikyta Java aplikacijų kūrimui [9.].

Plačiau apie Eclipse produktą ir kūrimo eigą galima pasiskaityti tinklapyje <http://www.eclipse.org/org/>

Eclipse platformos

Eclipse pateikia **RCP** (Rich Client Platform) aplikacijų kūrimui. Šią platformą sudaro tokie komponentai [10.]:

- Equinox OSGi – standartinis kūrimo karkasas
- Core platform – Eclipse pagrindas
- Standard Widget Toolkit (SWT) – portatyvinių valdiklių sąranka
- JFace – vartotojo sąsajos bibliotekos
- Eclipse Workbench – programavimo aplinkos darbalaukis: projektų administravimas, redaktoriai ir t.t.

Server platformos pagalba į Eclipse galima integruoti aplikacijų serverius tokius kaip Apache Tomcat, GlassFish ir kt. Platforma suteikia klaidų aptikimo bei taisymo galimybes realiu laiku [11.].

Eclipse Web Tools Platform projektas tai Eclipse platformos galimybių papildymas darbui su Web ir Java EE aplikacijomis. Jį sudaro išeities kodų ir grafiniai redaktoriai, vedliai ir integruotos aplikacijos programavimo darbams palengvinti. Taip pat įrankiai skirti aplikacijų užkrovimui, paleidimui ir testavimui [12.].

Modeliavimo platforma susideda iš visi oficialių Eclipse Foundation modeliavimo technologijų projektų. Jie visi yra suderinti su Eclipse Modeling Framework, kurį sukūrė IBM korporacija. Visi esami modeliavimo projektai yra suskirstyti į kelias kategorijas: Modelių transformavimas, Modelių kūrimo įrankiai, Konkrečios sintaksės kūrimas, Abstrakčios sintaksės kūrimas, Technologijos ir Tyrimai bei Amalgamos [13.].

5.2 Java aplikacijų serveriai

Šiuo metu vyrauja du pagrindiniai Java aplikacijų serveriai – GlassFish ir Tomcat.

5.2.1 *GlassFish*

GlassFish – tai atvirojo kodo aplikacijų serverio projektas skirtas Java EE platformai. Šį projektą pradėjo Sun Microsystems 2005 metais ir dabar jį tęsia Oracle organizacija. GlassFish palaiko visas Java EE API specifikacijas, tokias kaip JDBC, RMI, e-mail, JMS, web servisai, XML ir t.t. Taip pat šis serveris palaiko ir papildomas Java EE funkcijas: Enterprise JavaBeans, Connectors, servlets, portlets, JavaServer Pages ir keletą web servisų technologijų. Visa tai suteikia programuotojams daugiau galimybių kurti lankstesnes bei lengviau pritaikomas aplikacijas [14.].

Naujausia GlassFish projekto stabili versija yra 3.1.1 Ši versija yra maža, greita ir modulinė. Jos užkrovimo laikas neviršija 1 sekundės. Dėl serverio „lengvumo“ įgyvendintos tokios galimybės kaip aplikacijos atnaujinimas neuždarius sesijų, pridėtinis kodo atnaujinimas, kai nenutraukiamas aplikacijos veikimas atnaujinant tik tas dalis, kuriose buvo padaryti pakeitimai [16.].

Kadangi šis serveris yra modulinis, tai dėl šios priežasties suteikiama klasterizavimo galimybė, t. y. atskirose serveriuose paleisti GlassFish servisai gali būti apjungti į vieną serverį ir tokiu būdu kuo efektyviau išnaudoti turimus resursus. Taip pat yra ir galimybė perduoti sesijos kintamuosius tarp aplikacijų [17.].

5.2.2 *Tomcat*

Tomcat – Apache Group sukurtas atviro kodo interneto serveris. Apache Tomcat yra servletų konteineris, naudojamas su Java Servlet ir Java Server Page technologijomis. Norint Tomcat naudoti kaip aplikacijų serverį, papildomai reikia įdiegti jam skirtus papildomus komponentus.

Dabartinė Tomcat versija yra leidžiama su Catalina servletų konteineriu, Coyote HTTP konteineriu ir Jasper JSP varikliu. Tačiau šie komponentai gali būti keičiami.

Vienas iš Tomcat trūkumų, kad jis nepalaiko JavaEE. Norint tai eliminuoti, reikėtų naudoti TomEE, kuris apjungia Tomcat ir JavaEE, įtraukdamas Apache OpenEJB, Apache OpenWebBeans, Apache OpenJPA, Apache MyFaces ir kt. Tačiau ši savybė kartais laikoma ir privalumu, nes sistemose, kuriose nereikia daug papildomų funkcijų, Tomcat suteikia visą reikiamą funkcionalumą ir neapkrauna sistemos nereikalingomis funkcijomis [15.].

5.3 Moduliai Android aplikacijų kūrimui

Android aplikacijų kūrimui integruotose programavimo aplinkose dažnai diegiami papildomi moduliai, kurie palengvina tam tikrų kodo vietų rašymą ar jį net dalinai automatizuoja. Šiuo metu egzistuoja du moduliai, skirti skirtingiems įrankiams.

5.3.1 NBAndroid

NBAndroid – tai NetBeans platformos modulis, suteikiantis galimybę kurti Android aplikacijas.

Šį moduli sudaro keturios dalys [18.]:

- Android SDK branduolio palaikymas
- Projekto palaikymas
- LogCat peržiūra
- Android XML redaktorius

Android SDK branduolio palaikymas. Ši modulio dalis atsakinga už programavimo aplinkos ir Android SDK tarpusavio susijungimą. Šio modulio pagalba galima valdyti visus virtualius Android įrenginius esančius jūsų kompiuteryje [18.].

Android projektų palaikymas NetBeans programavimo aplinkoje sukurtas taip, kad būtų suderinamas su Android SDK. Tai reiškia, kad projekto struktūra yra tokia pat kaip ir kitų aplikacijų kuriamų Android projektų. Tokiu būdu suteikiama galimybė tą patį projektą vystyti skirtingais įrankiais [18.].

LogCat – tai speciali komanda, kuri suteikia galimybę peržiūrėti Android sistemos LOG failų turinį. Panaudojus šią komandą galima paleisti Android aplikaciją klaidų aptikimo režimu ir stebėti sistemos būsenas [18.].

Android XML redaktoriaus modulis įgalina NetBeans platformą redaguoti specifinius XML failus, kurie priklauso tik Android aplikacijoms. Šiuo metu yra atskiriama AndroidManifest.xml failo sintaksė ir pagrindinė layout failų struktūra [18.].

Apibendrinimas

NBAndroid modulis suteikia NetBeans platformai pagrindines galimybes kurti Android aplikacijoms ir jos vienintelis trūkumas yra tas, kad nėra galimybės redaguoti Android aplikacijos langų (layout) grafiniame režime.

5.3.2 *Android Developer Tools*

ADT (Android Developer Tools) – tai Eclipse platformos įskiepis, kuris pasiūlo jums priėjimą prie visų galimybių padėsiančių greitai ir kokybiškai kurti Android aplikacijas. ADT suteikia daugelio Android SDK įrankių komandų grafinį valdymą, taip pat ir grafinės vartotojo sąsajos kūrimo įrankį su prieinamomis visomis galimybėmis grafiniame režime [19.].

- Pagrindinės ADT galimybės:
- Integruotas Android projektų kūrimas, instaliavimas ir klaidų taisymas
- Android SDK įrankių grafinis valdymas
- Java programavimo kalbos ir XML redaktoriai

Integruotas Android projektų kūrimas, instaliavimas ir klaidų taisymas

ADT suteikia visas darbo su Android projektu galimybes, palengvinančias Android aplikacijų kūrimą. Kuriami Android projektai yra tam tikro standarto, kuriuos galima redaguoti ir kitomis Android aplikacijų kūrimo platformomis. Taip pat suteikiamos galimybės sukurtų aplikacijų klaidų aptikimui (debugging) [19.].

Android SDK įrankių grafinis valdymas. Dauguma Android SDK įrankių komandų yra įtrauktos į Eclipse platformos meniu punktus, perspektyvas ar kaip atskiri procesai atliekant tam tikrus veiksmus. Galimybės valdyti virtualius Android įrenginius [19.].

Java programavimo kalbos ir XML redaktoriai pilnai palaiko Android karkasą, ir pateikia visas galimas jo klases bei jų funkcijas. Šis įskiepis taip pat suteikia visas galimybes redaguoti specifinius Android XML failus grafinių formų bei vedlių pagalba. Taip pat grafinis Android aplikacijos langų kūrimas labai supaprastina ir palengvina darbą [19.].

Apibendrinimas

Kadangi ADT yra pirminis Android aplikacijų kūrimo įrankis, tai jis turi visas galimybes. Visos naujovės pirmiausiai įgyvendinamos šiame Eclipse įskiepyje, todėl jis visada yra naujausias Android aplikacijų galimybių panaudojimo įrankis.

5.3.3 *NBAndroid ir ADT palyginimas*

Atlikus Android modulių apžvalgą, jų savybes galima apibendrinti lentelėje nr. 2.

Lentelė nr. 2. NBAndroid ir ADT modulių savybių palyginimas

Savybė	Modulis	NBAndroid	Android Developer Tools
Android SDK branduolio palaikymas		+	+
Projekto palaikymas		+	+
Android SDK įrankių grafinis valdymas			+
Java programavimo kalbos ir XML redaktoriai		+	+
Klaidų aptikimo įrankis		+	+
Grafinis vartotojo sąsajos kūrimas			+

Kaip pastebima, ADT modulis turi daugiau naudingų savybių, kurios gali palengvinti Android aplikacijų kūrimą.

5.4 Programinių įrankių pasirinkimo apibendrinimas

Remiantis Kalifornijos universiteto atliktu tyrimu apie integruotų programavimo aplinkų patikimumą [28.] sistemos branduolio kūrimui pasirinktas Java programavimo įrankis „NetBeans“. Norint nustatyti ar bus susidurta su kokiais nors sunkumais ir kaip juos bus galima išspręsti.

Taip pat nuspręsta naudoti GlassFish aplikacijų serverį, nes ateityje norit tobulinti sistemą, GlassFish turės daugiau galimybių, nei šiuo metu siūlomos Tomcat funkcijos.

Išanalizavus Android aplikacijų kūrimo modulius ir jų galimybes, nutarta mobiliąją programėlę kurti Eclipse įrankiu, panaudojant ADT modulį.

6. PROBLEMOS SPRENDIMAS

H. Zhai-wei, Z. Hai-xia ir G. Guo-hong analizavo metodus, kaip galima optimizuoti žiniatinklio servisu [30.]. Jie analizavo kaip išsidalina užklauso aptarnavimo serveryje laikas, priklausomai nuo apdoravimo veiksmo (sisteminiai kvietimai, XML nagrinėjimas, rezultato išvedimas ir pan.). Taip pat stebėjo kaip keičiasi siuntimo laikas naudojant duomenų kodavimą arba skirtingo tipo spartinančiosios atminties technologijas.

Darbe „Technology and Recent Development of XML Web Services“ [29.] analizuojami skirtingi būdai XML kodo sumažinimui, kad XML žiniatinklio servisas veiktų kuo greičiau. Šio darbo autoriai taip pat lygina kiek efektyvios jų analizuotos XML kodo mažinimo technologijos.

M. Tian ir kiti analizavo kaip mobilių žiniatinklio servisu greitis priklauso nuo XML kodo suspaudimo, nes jį naudojant sumažėja XML kodo dydis, tačiau pailgėja serviso vykdymo laikas [31.].

Darbe „Web Services on Embedded Devices“ taip pat analizuojamos žiniatinklio servisu charakteristikos mobiliuosiuose įrenginiuose, tačiau šie autoriai tyrė serviso veikimo greitį skirtingose aplikacijose [32.].

Christin Grob ir kiti darbe „Web services on embedded systems - A performance study“ vertino žiniatinklio serviso generuojamo XML kodo savybes [33.].

Nors tyrimų apie žiniatinklio servisu atlikta nemažai, tačiau palyginimo tarp XHTML ir XML formato duomenų nepavyko rasti.

6.1 XHTML ir XML duomenų formato taikymo mobiliuosiuose įrenginiuose tyrimas

Siekiant palyginti žiniatinklio serviso vykdymo charakteristikas naudojant skirtingo tipo duomenų gražinimo formatus, buvo sukurta organizacijos informacinės sistemos inventoriaus ir jo naudojimo apskaitos sistema. Joje realizuotas skirtingo formato duomenų išvedimas. Naudojant XHTML formato servisą mobiliajame įrenginyje būtina interneto naršyklė, o naudojant XML formatą – būtina papildoma rezultatų pateikimo posistemė, todėl buvo sukurta speciali Android programėlė.

Tyrimo metu analizuojamos dvi pagrindinės charakteristikos:

- Interneto srauto poreikis – internetu gaunamų duomenų kiekis;
- Serverio apkrova – vienos užklauso aptarnavimo laikas serveryje.

Tyrimo metu buvo naudojamos šešios skirtingos situacijos, kurios tarpusavyje skyrėsi duomenų kiekiu – buvo vaizduojama pasirinkto įrenginio informacija ir panaudojimo istorija,

kurioje yra nuo 1 iki 36 įrašų. Bendras įrenginio duomenų kiekis visiems įrenginiais yra labai panašus, todėl situacijos išskiriamas pagal įrenginio panaudojimo įrašų kiekį:

- Inventorius nr. 1 – nurodytas 1 panaudojimo įvykis;
- Inventorius nr. 2 – nurodyti 2 panaudojimo įvykiai;
- Inventorius nr. 3 – nurodyti 4 panaudojimo įvykiai;
- Inventorius nr. 4 – nurodyti 7 panaudojimo įvykiai;
- Inventorius nr. 5 – nurodyta 10 panaudojimo įvykių;
- Inventorius nr. 6 – nurodyti 36 panaudojimo įvykiai.

Tyrimas buvo atliekamas su keliais mobiliaisiais telefonais, turinčiais Android operacinę sistemą. Jo metu kiekviename iš mobiliųjų telefonų buvo tiriami abu architektūros sprendimai ir vertinamos vidutinės stebimų charakteristikų reikšmės skirtingoms situacijoms.

Realizuotoje sistemoje informacijos apie įrenginį peržiūrai turi būti pateikiama įrenginio informacija, jo naudojimo istorija, galimų naudoti operatorių ir veiksmų su įrenginiu sąrašai. XHTML formatu visa ši informacija siunčiama vienu metu. Tuo tarpu XML formato servise plečiamumui padidinti naudojamos trys atskiros funkcijos: įrenginio informacijos ir jo panaudojimo informacija; operatorių sąrašas; galimų veiksmų su įrenginiu sąrašas. Todėl XML formato atveju objektyvumui pasiekti turi būti analizuojama bendra šių trijų funkcijų reikšmė.

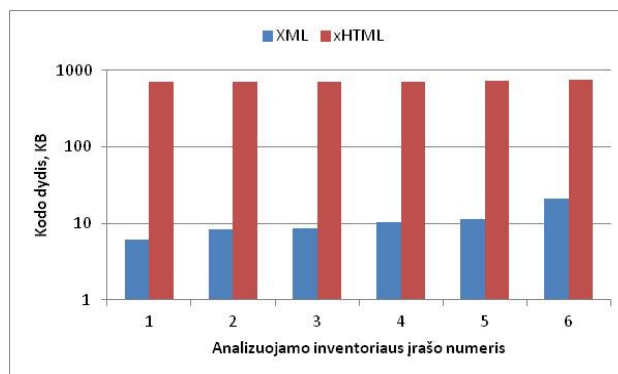
6.2 Interneto srauto poreikio tyrimas ir jo rezultatai

Interneto srauto poreikio tyrimui buvo analizuojamas XML ir XHTML kodo dydžiai. XML faile aprašyta įrenginio informacija (duomenys ir kontekstas), o XHTML kode naudojami ne tik reikiami duomenys, bet ir jų atvaizdavimo vartotojui kodas. Todėl, kaip ir buvo tikėtasi, XHTML atveju yra siunčiami didesni duomenų kiekiai iš serverio klientui (žr. Lentelė nr. 3).

Lentelė nr. 3. Duomenų kiekių palyginimas skirtingiems analizuojamiems inventoriaus įrašams.

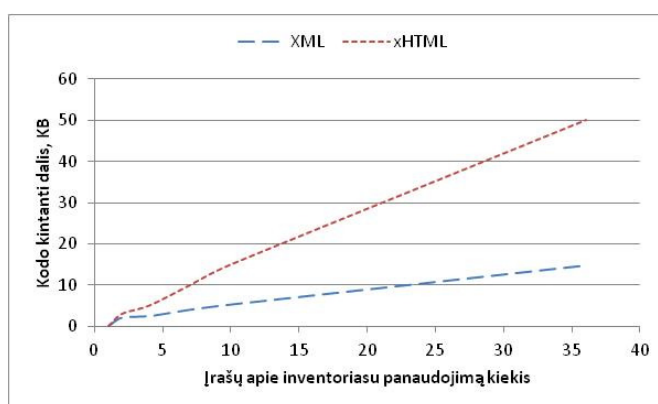
Formatas	Siunčiami duomenys	Inventoriaus numeris					
		1/1	2/2	3/4	4/7	5/10	6/36
XML	Operatorių sąrašas	1,119 KB					
	Veiksmų sąrašas	1,644 KB					
	Inventoriaus informacija	3,447 KB	5,498 KB	5,902 KB	7,471 KB	8,740 KB	18,297 KB
	Bendra inventoriaus informacija	6,210 KB	8,261 KB	8,665 KB	10,234 KB	11,503 KB	21,060 KB
xHTML	Bendra inventoriaus informacija	704,512 KB	707,458 KB	709,254 KB	714,621 KB	719,628 KB	754,951 KB

Didėjant įrašų apie inventoriaus panaudojimą kiekiui tiesiškai auga ir siunčiamo kodo dydis (žr. Pav 12).



Pav 12 Siunčiamo kodo dydis

Kodo dydis akivaizdžiai didesnis xHTML kodo atveju, tačiau tai labai priklauso nuo sistemos naudojamo dizaino. Todėl palyginome kaip keičiasi kodo dydis, priklausomai nuo įrašų apie inventorių skaičiaus (žr. Pav 13).



Pav 13 Kodo dydžio priklausomybė nuo įrašų kiekio

Kodo pokyčio analizė parodė, kad mūsų analizuotu atveju xHTML kodo dydis kito vidutiniškai 2,48 karto greičiau nei XML kodas. Teorinės prielaidos rodo, kad labai retais atvejais XML kodo kitimas gali būti greitesnis jei xHTML (xHTML dizainas turėtų būti visiškai minimalus, o XML faile turėtų būti naudojama labai gili ir sudėtinga duomenų struktūra, ilgi žymių pavadinimai).

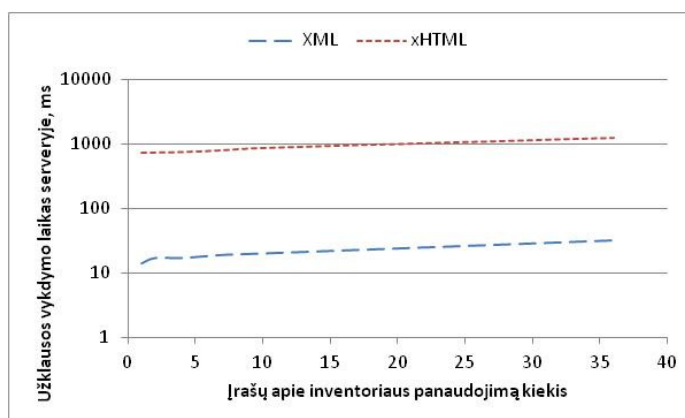
6.3 Serverio apkrovos tyrimas ir jo rezultatai

Viena iš XML formato naudojimo privalumų laikoma savybė, kad dalis apkrovos padalinama patiems klientams, todėl mažėja serverio apkrova. Įvertinti xHTML ir XML formato atvejais tenkančią apkrovą serveriui išmatavome kiek vidutiniškai laiko trunka vienos vartotojo užklausos aptarnavimas serveryje (žr Lentelė nr. 4).

Lentelė nr. 4. Aptarnaujamų vartotojo užklausų laiko palyginimas skirtingiems inventoriaus įrašams.

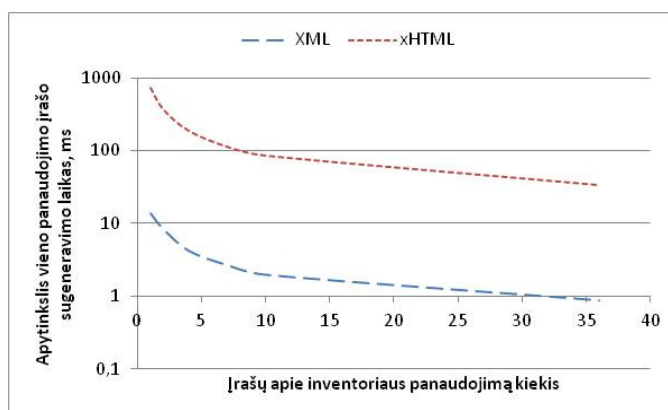
Formatas	Generuojamas kodas	Inventoriaus numeris					
		1/1	2/2	3/4	4/7	5/10	6/36
XML	Operatorių sąrašas	2 ms					
	Veiksmų sąrašas	1 ms					
	Inventoriaus informacija	11 ms	14 ms	14 ms	16 ms	17 ms	29 ms
	Bendra inventoriaus informacija	14 ms	17 ms	17 ms	19 ms	20 ms	32 ms
xHTML	Bendra inventoriaus informacija	732 ms	740 ms	753 ms	800 ms	867 ms	1231 ms

Rezultatai parodė, kad čia taip pat galioja priklausomybė nuo inventoriaus panaudojimo įrašų kiekio (žr. Pav 14).



Pav 14 Laiko priklausomybė nuo įrašų kiekio

Vertinant vidutinį vieno inventoriaus panaudojimo įrašo sugeneravimo laiką, pastebima, kad galioja laipsninė regresija (žr. Pav 15).



Pav 15 Vieno įrašo generavimo laiko ir įrašų kiekio ryšys

Tai paaiškinama tuo, kad sistemoje nėra naudojama po užklausą kiekvienam įrašui apie panaudojimą gauti, o visi vieno inventoriaus panaudojimo įrašai gaunami vienu metu.

6.4 Sistemos mobiliesiems įrenginiams architektūros pasirinkimas

Įvertinus mobiliajame įrenginyje veikiančiai sistemai keliamus reikalavimus ir atlikto tyrimo rezultatus nuspręsta kurti Android programėlę. Toks pasirinkimas turėtų sumažinti aplikacijų serverio apkrovą ir bendrą sistemos veikimo laiką.

7. IŠVADOS

1. Iš rinkoje esančių panašaus pobūdžio sistemų analizės seka, jog visos yra orientuotos į inventoriaus apskaitą, todėl sukurta sistema už jas yra pranašesnė inventoriaus valdymo funkcijomis.
2. Atlikus programavimo kalbų analizę buvo atsižvelgta į sistemai keliamus architektūros reikalavimus bei finansavimą ir pagal tai pasirinkta – JAVA programavimo kalba, kurios populiariausi programavimo įrankiai yra Eclipse ir NetBeans. Šių dviejų platformų palyginimo išvados nurodo, jog NetBeans platforma Windows aplinkoje turi daugiau klaidų ir yra sunkiau valdoma, todėl ji pasirinkta siekiant nustatyti ar bus susidurta su kokiais nors sunkumais kūrimo metu. Iš Android aplikacijų kūrimo įrankių analizės buvo nustatyta, jog daugiau funkcijų turi ir yra lengviau valdomas Eclipse Android modulis, o išanalizavus du pagrindinius JAVA WEB aplikacijų serverius buvo nuspręsta rinktis GlassFish, nes pastarasis atitiko serveriui keliamus reikalavimus.
3. Sistemos projektavimo metu buvo apžvelgti sistemos funkciniai bei architektūros reikalavimai, kurių pagrindu suprojektuota ir realizuota reliacinė duomenų bazė, WEB aplikacija, WEB servisas bei Android aplikacija
4. Atlikus sistemos funkcijų veikimo ir našumo su įmonės pateiktais duomenimis testavimo darbus nepastebėta jokių nesklandumų, įtakojančių sistemos ar vartotojo darbą, todėl sistema buvo sėkmingai įdiegta įmonėje.
5. Sukurtą sistemą galima pilnai naudoti įmonės reikmėms (žr. Priedas nr. 1).

8. LITERATŪROS SĄRAŠAS

1. Rick Grehan, NetBeans IDE review. Prieiga per internetą: <<http://review.techworld.com/applications/3212767/netbeans-ide-review/>> [žiūrėta: 2012-01-30]
2. Oficialus NetBeans Profiler modulio puslapis. Prieiga per internetą: <<http://profiler.netbeans.org/>> [žiūrėta: 2012-01-30]
3. NetBeans IDE Features: Swing GUI Builder. Prieiga per internetą: <<http://netbeans.org/features/java/swing.html>> [žiūrėta: 2012-01-30]
4. Oficialus NetBeans JavaScript palaikymo modulio puslapis. Prieiga per internetą: <<http://wiki.netbeans.org/JavaScript>> [žiūrėta: 2012-01-30]
5. NetBeans IDE Features: Java EE and Java Web Applications. Prieiga per internetą: <<http://netbeans.org/features/web/index.html>> [žiūrėta: 2012-01-30]
6. NetBeans IDE Features: Java ME (Micro Edition) & Embedded. Prieiga per internetą: <<http://netbeans.org/features/javame/index.html>> [žiūrėta: 2012-01-30]
7. NetBeans IDE Features: NetBeans IDE Dynamic Languages. Prieiga per internetą: <<http://netbeans.org/features/scripting/index.html>> [žiūrėta: 2012-01-30]
8. NetBeans IDE Supported technologies. Prieiga per internetą: <<http://netbeans.org/downloads/>> [žiūrėta: 2012-01-30]
9. About the Eclipse Foundation. Prieiga per internetą: <<http://www.eclipse.org/org/>> [žiūrėta: 2012-01-30]
10. Oficialus Eclipse Rich Client Platform puslapis. Prieiga per internetą: <http://wiki.eclipse.org/index.php/Rich_Client_Platform> [žiūrėta: 2012-01-30]
11. Eclipse on Wikipedia. Prieiga per internetą: <http://en.wikipedia.org/wiki/Eclipse_%28software%29#cite_note-13> [žiūrėta: 2012-01-30]
12. Eclipse Web Tools Platform Project. Prieiga per internetą: <<http://www.eclipse.org/projects/project.php?id=webtools>> [žiūrėta: 2012-01-30]
13. Eclipse Modeling Project. Prieiga per internetą: <<http://www.eclipse.org/projects/project.php?id=modeling>> [žiūrėta: 2012-01-30]
14. GlassFish on Wikipedia. Prieiga per internetą: <<http://en.wikipedia.org/wiki/GlassFish>> [žiūrėta: 2012-01-30]
15. Apache Tomcat Documentation. Prieiga per internetą: <<http://tomcat.apache.org/tomcat-7.0-doc/index.html>> [žiūrėta: 2012-01-30]

16. What is GlassFish v3? Prieiga per internetą: <<https://wikis.oracle.com/display/GlassFish/PlanForGlassFishV3>> [žiūrėta: 2012-01-30]
17. Sang Shin, GlassFish Overview. Prieiga per internetą: <http://fmisociety.org/ITLeadersAcademy/index_php/lectures/2008-2009/5-sun/presentation-part-3.pdf> [žiūrėta: 2012-01-30]
18. Oficialus NBandroid puslapis. Prieiga per internetą: <<http://www.nbandroid.org/p/features.html>> [žiūrėta: 2012-01-31]
19. Oficialus Android Developer Tools puslapis. Prieiga per internetą: <<http://developer.android.com/guide/developing/tools/adt.html>> [žiūrėta: 2012-01-31]
20. InFlow Features. Prieiga per internetą: <<http://www.inflowinventory.com/software-features.aspx>> [žiūrėta: 2012-01-31]
21. Inventoria Professional Inventory Software. Prieiga per internetą: <<http://store.hermanstreet.com/other/inventoria-download/?&ICID=pin-Inventoria%202011-05-19&ofm>> [žiūrėta: 2012-01-31]
22. iMagic Inventory. Prieiga per internetą: <<http://www.imagicinventorysoftware.com/InventoryFeatureList.php>> [žiūrėta: 2012-01-31]
23. Inventory Organizer Deluxe. Prieiga per internetą: <<https://www.regnow.com/checkout/product/8107-15>> [žiūrėta: 2012-01-31]
24. Small Business Inventory Controll. Prieiga per internetą: <<http://www.nstarsolutions.com/store/product.php?productid=9834&cat=0&page=1&featured>> [žiūrėta: 2012-01-31]
25. Inventory Pro Features. Prieiga per internetą: <<http://www.mrtec.com/ProductFeatures.htm>> [žiūrėta: 2012-01-31]
26. Wasp Inventory Control – Inventory Software. Prieiga per internetą: <http://www.waspbarcode.com/inventory_control/inventory_control.asp> [žiūrėta: 2012-01-31]
27. Altiris Inventory Solution. Prieiga per internetą: <<http://www.symantec.com/inventory-solution>> [žiūrėta: 2012-01-31]
28. Xinlei (Oscar) Wang, Eilwoo Baik, Premkumar T. Devanbu. Operating System Compatibility Analysis of Eclipse and Netbeans Based on Bug Data (2011) http://csiflabs.cs.ucdavis.edu/~xlwang/public/msr_2011_wang.pdf
29. Hazem M. El-Bakry, Alaa M. Riad, Ahmed E. Hassan, Nikos Mastorakis, Qusay F. Hassan. Technology and Recent Development of XML Web Services. RECENT

ADVANCES in BUSINESS ADMINISTRATION (2010) < <http://www.wseas.us/e-library/conferences/2010/Cambridge/ICBA/ICBA-14.pdf> >

30. Hou Zhai-wei, ZHAI Hai-xia, ir Gao Guo-hong. A Study on Web Services Performance Optimization. Prieiga per internetą: <<http://www.academypublisher.com/proc/isecs10w/papers/isecs10wp184.pdf>> [žiūrėta: 2012-01-12]
31. M. Tian, T. Voigt, T. Naumowicz, H. Ritter, J. Schiller. Performance Considerations for Mobile Web Services. Prieiga per internetą: <<http://www.sics.se/~thiemo/ASWN2003.pdf>> [žiūrėta: 2012-01-12]
32. Daniel Schall, Marco Aiello, Schahram Dustdar. Web Services on Embedded Devices. Prieiga per internetą: <<http://www.infosys.tuwien.ac.at/staff/sd/papers/Web%20Services%20on%20Embedded%20Devices.pdf>> [žiūrėta: 2012-01-12]
33. Christin Groba, Siobhan Clarke. Web services on embedded systems - A performance study. Prieiga per internetą: <<http://www.webofthings.org/wot/2010/pdfs/152.pdf>> [žiūrėta: 2012-01-12]

9. ANOTACIJA

Turint įmonėje ar organizacijoje tam tikrus svarbius išteklius ar objektus, visuomet kyla poreikis juos inventorizuoti, sekti jų judėjimą, kaitą, būsenas ir kitą su jais susijusią informaciją. Įmonių veikloje turimas inventorių turi būti apskaitomas ne tik patogumui užtikrinti, bet yra būtinas tinkamam įmonės darbo organizavimui.

Šio darbo tikslas – sukurti įmonės informacinės sistemos inventoriaus apskaitos, kiekvieno įrenginio migravimo sekimo ir atliktų darbų istorijos valdymo sistemą, kuri būtų nepriklausoma nuo operacinės sistemos ir galėtų būti valdoma mobiliuoju įrenginiu.

Darbe išanalizuotos rinkoje esančios inventoriaus valdymo sistemos, nurodyti jų privalumai ir trūkumai. Aprašyti įrankių bei serverių pasirinkimo kriterijai. Taip pat pateikti įmonės informacinės sistemos inventoriaus valdymo sistemos projektavimo etapai: sistemos architektūra, duomenų srautai, duomenų bazė, sistemos paketai bei jų tarpusavio ryšiai, kurie remiasi atlikto tyrimo sistemos architektūros savybių pasirinkimui įvertinti rezultatais.

Šio darbo metu sukurta sistema jau yra naudojama užsakovo įmonėje ir tenkina visus užsakovo iškeltus reikalavimus jos panaudojimui ir funkcionalumui. Tačiau ši sistema yra lanksti ir gali būti taikoma ir kitų įmonių informacinės sistemos inventoriaus valdymui.

10. SUMMARY

With the company or organization in a number of important resources or objects, is always a need to inventory, track their movement, change, states and other related information. Business inventory is available to be accounted for not only convenience, but it is essential for the proper organization of the company's work.

The aim of this work - to create enterprise information systems inventory accounting, tracking each device migration and done job history management system that is independent of the operating system and can be controlled by a mobile device.

The paper analyzes the market for existing inventory management system and points out their advantages and disadvantages. There are also described the tools and servers selection criteria. It also provides enterprise IT management system design phases: architecture, data flows, data base, system packages and their mutual relations, which are based on a study of the characteristics of the system architecture choice results.

The developed system is already being used in company client and client meets all the requirements raised by its use and functionality. However, this system is flexible and can be applied to other companies' information systems inventory management.

11. PRIEDAS NR. 1



AB „ŠIAULIŲ ENERGIJA“

Šiaulių universitetui
Matematikos ir informatikos fakultetui
Informatikos katedrai

PAŽYMA

2012-05-02
Šiauliai

Patvirtiname, kad AB „Šiaulių energija“ informatikos skyriaus darbuotojo, Deno Pavlavičiaus, sukurta informacinių sistemų inventoriaus valdymo sistema yra įdiegta ir naudojama AB „Šiaulių energija“ įmonėje.

Informatikos skyriaus viršininkas

informatikos skyriaus
viršininkas
Evaldas Pilūra

(vardas pavardė parašas)

12. PRIEDAS NR. 2

REKOMENDACIJOS

Sistemos naudingumo rekomendacijos

Norint pasiekti maksimalų sistemos naudingumą rekomenduojama laikytis tokių taisyklių:

- Stengtis sistemoje atkartoti realią situaciją, t. y. Suvesti į sistemą tikrus pastatus, juose esančias patalpas bei darbo vietas, kuriose patalpinti įrenginiai ir dirba darbuotojai.
- Jeigu vienoje patalpoje yra daugiau nei vienas darbuotojas, bet kiekvienas iš jų dirba tik su jiems skirtais įrenginiais, tai rekomenduojama kurti atskiras darbo vietas, ir atitinkamai, jeigu patalpoje yra vienas ar daugiau darbuotojų bet jie gali dirbti su bet kuriuo patalpoje esančiu įrenginiu, tai rekomenduojama kurti tik vieną darbo vietą.
- Kiekvienas įrenginys sistemoje turi turėti tik jam priskirtą virtualų vardą, kurį rekomenduojama parinkti pagal įrenginio tipą, pvz.: kompiuteris gali būti pavadintas K00001, monitorius – M00001 ir t. t.
- Kiekvienas įrenginys bei darbo vieta gali turėti savo komentarus. Jais galima sekti kiekvieno objekto istoriją, todėl rekomenduojama atlikus bet kokį veiksmą darbo vietoje ar su tam tikru įrenginiu, jo komentarų sąrašė užregistruoti naują komentarą.
- Siekiant jog įrenginių paieška pagal atributus būtų lengvesnė ir aiškesnė rekomenduojama įrenginių atributus vesti bendrinius ir logiškai suprantamus.

Sistemos našumo rekomendacijos

Sistemos našumas priklauso nuo serverio resursų bei apkrovimo. Kadangi sistemai reikalingi du servisai: aplikacijų serveris GlassFish V3 ir duomenų bazių serveris MySQL, tai siekiant didesnio našumo rekomenduojama juos išskirti į atskirus serverius arba jiems priskirti atskirus procesorius, operatyviają atmintį bei kietą diską, kad abu galėtų dirbti vienu metu ir netrukdydami vienas kitam.

Kadangi sistema pasiekama nuotoliniu būdu per naršyklę arba Android aplikaciją, rekomenduojama užtikrinti tinklo ryšį tarp serverio ir klientų.

Kliento kompiuterio minimalūs reikalavimai:

- Užtikrintas tinklo ryšys

- Firefox/Chrome/Opera/IE7 ir naujesnēs

Minimalūs serverio reikalavimai:

- MySQL 5.1
- GlassFish v3.0.1
- 2 GB operatyvinės atminties
- Intel(R) Xeon(R) CPU 2,00 GHZ spartos
- OS – Linux/Solaris/Windows/Mac OS

13. PRIEDAS NR. 3

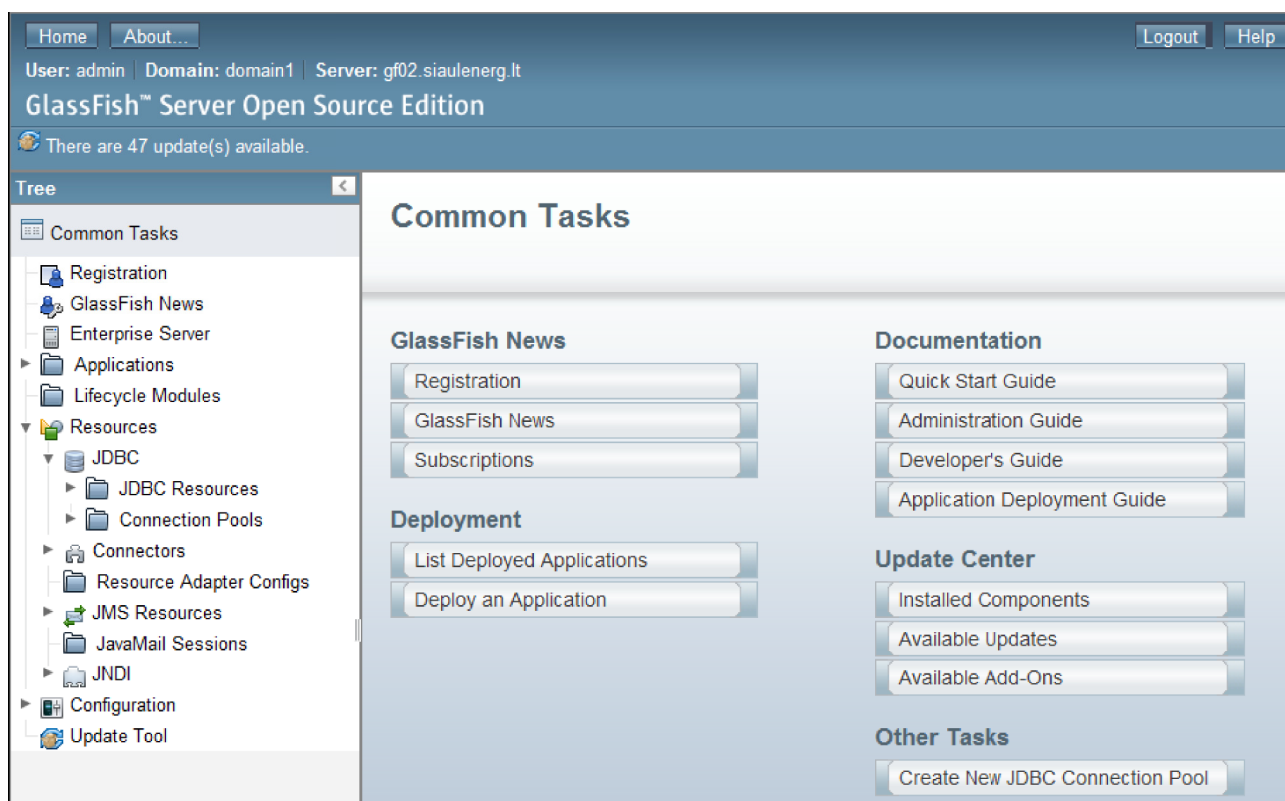
DIEGIMAS

Sistema gali būti įdiegta tik į GlassFish V3 aplikacijų serverį. Diegimui pateikiami du failai:

- DB.sql – tai duomenų bazės sukūrimo failas, kurį reikia paleisty MySQL duomenų bazių serveryje.
- KompiuteriuApskaita.war – tai aplikacijos diegimo failas, kurį reikia išskeisti GlassFish V3 aplikacijų serveryje.

Aplikacijos diegimas GlassFish V3 serveryje

Norint įdiegti aplikaciją GlassFish V3 serveryje, reikia patekti į administravimo aplinką. Standartiškai ją pasiekti galima per naršyklę nurodžius serverio adresą ir 4848 prievadą. Patekus į administravimo aplinką iš pagrindinio meniu išrenkamas punktas „Deploy an Application“. Atsidariusiame naujame lange parenkamas KompiuteriuApskaita.war failas ir paleidžiamas diegimas.



Pav 1 GlassFish administravimo aplikacija

Kai aplikacija įdiegta reikia sukurti JDBC duomenų bazės jungimosi profilį. Tai padaryti galima per „Resources > JDBC“ meniu punktą. Pirmą sukuriamas „Connection Pool“ pavadinimu „kompApsk“.

Antras žingsnis – sukurti JDBC resurso profilį pavadinimu „jdbc/kompApsk“ nurodantį į prieš tai sukurtą „Connection Pool“.

Atlikus šiuos veiksmus sistemą galima pasiekti per naršyklę įvedus serverio adresą ir aplikacijos pavadinimą, kurį suteikėte instaliacijos metu. Android aplikaciją pradėti naudoti galima nustačius serverio adresą.

14. PRIEDAS NR. 4

TESTAVIMAS

Sistemos funkcionalumas

Realizavus sistemą ir surašius visus duomenis į duomenų bazę, atliekamas funkcionalumo testavimas. Testavimo metu tikrinama ar veikia visos programos funkcijos, ar visos funkcijos veikia teisingai ir rezultatai tenkina vartotoją.

Norint užtikrinti sistemos duomenų išrinkimo teisingumą buvo atliekamas toks bandymas. Kadangi apie dabartinę inventoriaus būseną galima gauti tik sąrašo formatu iš organizacijos buhalterinės apskaitos sistemos, tai atliekamas tik vienas palyginimo bandymas. Iš buhalterinės apskaitos atspausdinamas informatikos skyriaus inventoriaus sąrašas su tokiais filtrais:

- Pavadinimas prasideda žodžiu „Kompiuteris“.
- Įsigijimo data vėlesnė už 2000-01-01.

Gautą sąrašą sudaro 40 įrašų su unikaliais inventoriniais numeriais.

Toliau atliekamas tokio pat sąrašo generavimas sukurtoje sistemoje ir gaunamas toks pat sąrašas su 40 įrašų, kurių aprašai ir inventoriniai numeriai sutampa su buhalterinės apskaitos sąrašu.

Toliau atliekamas visų sistemos funkcijų veikimo patikrinimas naudojant skirtingas naršykles. Išbandytos šiuo metu populiariausios naršyklių versijos: Mozilla Firefox 11, Google Chrome, Opera ir IE. Bandymų metu pastebėta, jog naudojant naršyklę IE6 ir senesnę išsikraipo sistemos vaizdas ir neveikia kai kurios funkcijos, kurios atnaujina puslapio duomenis visiškai jo neperkrovus (naujų įrašų kūrimas lentelėse ir esamų trynimai).

Sekantis bandymas – Android aplikacijos funkcionalumo patikrinimas. Pasirinkus kelis kompiuterius ir jų inventorinius numerius atliekama paieška Android aplikacijoje ir kompiuterio naršyklėje. Gauti rezultatai yra identiški tiek įrenginių informacija, tiek komentarai.

Sistemos našumas

Sistemos našumui nustatyti buvo filtruojami ir pasirenkami peržiūrai įrenginiai, kurie turi mažiausiai ir daugiausiai atributų bei komentarų. Bandymų metu buvo pastebėta, jog didžiausią įtaką rezultatų užkrovimo laikui per naršyklę daro puslapio dizainas, nors jis yra minimalus, bet sudaro beveik 700 KB informacijos kiekio, kai patys įrenginio duomenys esant minimaliam įrenginio atributų bei komentarų kiekiui sudaro tik apytiksliai 3,5 KB.

Atsižvelgiant į našumą buvo realizuotas WEB servisas ir Android aplikacija, kuri tą pačią informaciją vartotojui suteikia apytiksliai 40 kartų greičiau esant labai geram interneto ryšiui.

15. PRIEDAS NR. 5

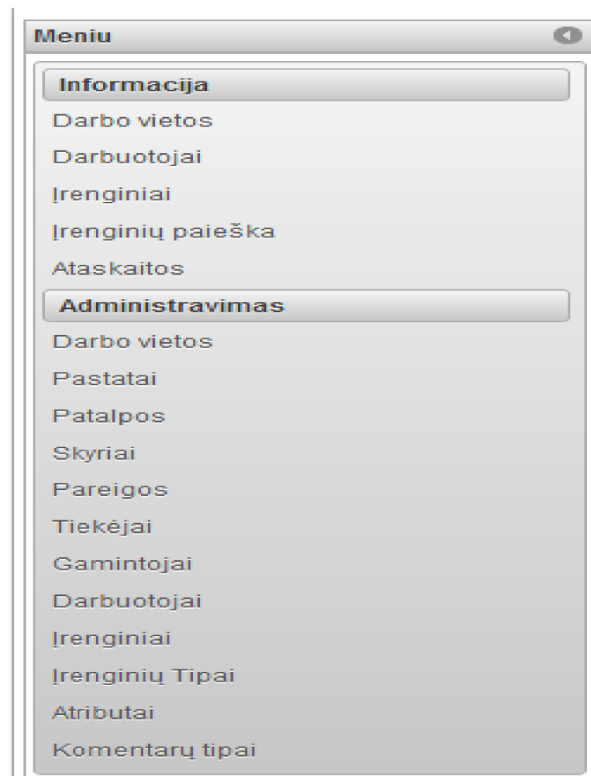
PROGRAMINIO PRODUKTO APRAŠAS

Sukurta sistema galima valdyti dviem būdais:

- Per naršyklę
- Išmaniojo telefono pagalba per Android aplikaciją

Pirmasis būdas įgalina vartotoją valdyti ir peržiūrėti visą informaciją. Naršyklinės aplikacijos pagalba galima administruoti visus sistemos objektus (žr. Pav 1):

- Pastatus
- Patalpas
- Darbo vietas
- Skyrius
- Pareigas
- Darbuotojus
- Įrenginius
- Įrenginių tipus
- Gamintojus
- Tiekėjus
- Atributus
- Komentarų tipus



Pav 1 Naršyklinės aplikacijos meniu

Visų objektų (išskyrus darbo vietas) administravimas yra paprastas: redaguojant ar kuriant naują objektą tiesiog suvedami to objekto parametrai (pvz.: pavadinimas).

Administruojant darbo vietas visų pirma sistemoje turi būti įvesta bent viena patalpa, kuriai bus priskirta nauja darbo vieta. Kuriama darbo vieta gali visiškai neturėti nei vieno darbuotojo ar įrenginio. Naujos darbo vietos kūrimo lange (žr. Pav 2) matomi laisvų įrenginių ir darbuotojų sąrašai, kuriuos galima priskirti darbo vietai. Sudarius darbo vietos darbuotojų ir įrenginių sąrašą parenkamas pastatas bei patalpa, kurioje fiziškai yra kuriama darbo vieta.

Laisvi įrenginiai

(1 of 9) 1 2 3 4 5 6 7 8 9 5

	Tipas	Pavadinimas	Virtualus vardas	Gamintojas	Tiekejas	Serijos nr.	Inventorinis Nr.	Įsigijimo data
	Mon							
+	Monitorius			Nenurodytas	Nenurodytas			
+	Monitorius			Nenurodytas	Nenurodytas			
+	Monitorius	Q790		Image	Nenurodytas			
+	Monitorius			Nenurodytas	Nenurodytas			
+	Monitorius	2233SN		Samsung	Nenurodytas			

(1 of 9) 1 2 3 4 5 6 7 8 9 5

Darbo vietos įrenginiai

Tipas	Pavadinimas	Virtualus vardas	Gamintojas	Tiekejas	Serijos nr.	Inventorinis Nr.	Veiksmas
Kompiuteris							
Monitorius							
Telefonas							
UPS							
Monitorius							

Laisvi darbuotojai

(1 of 58) 1 2 3 4 5 6 7 8 9 10 5

	Pavardė	Vardas	Pareigos	Skyrius
+				
+				
+				
+				
+				

(1 of 58) 1 2 3 4 5 6 7 8 9 10 5

Darbo vietos darbuotojai

Pavardė	Vardas	Pareigos	Skyrius	Veiksmas
Pavliavičius	Denas	Inžinierius-programuotojas	INFORMATIKOS SKYRIUS	

Darbo vietos informacija

Pavadinimas:

Pastatas:

Patalpa:

Išsaugoti

Pav 2 Darbo vietos administravimas

Informacijos paieška aplikacijoje taip pat suskirstyta į atskiras dalis:

- Darbo vietų
- Darbuotojų
- Įrenginių
- Įrenginių paieška pagal atributus

Darbo vietų paieška atliekama pagal darbuotojus arba įrenginius. Išsirinkus paieškos tipą galima atlikti atskirus kiekvieno stulpelio filtrus (žr. Pav 3). Tokiu būdu suradus norimą darbo vietą galima pereiti iškart prie jos redagavimo arba peržiūros.

Darbo vietų filtras pagal:

Tipas	Pavadinimas	Virtualus vardas	Gamintojas	Tiekejas	Serijos nr.	Inventoris Nr.	Darbo vieta	Pastatas	Aukštas	Kabineto nr.	Veiksmai
Kompiuteris	[P] [A]
Kompiuteris	[P] [A]
Kompiuteris	[P] [A]
Kompiuteris	[P] [A]
Kompiuteris	[P] [A]
Kompiuteris	[P] [A]
Kompiuteris	[P] [A]
Kompiuteris	[P] [A]
Kompiuteris	[P] [A]
Kompiuteris	[P] [A]

503

Pav 3 Darbo vietų paieška

Darbo vietos informacijos lange (žr. Pav 4) matomas darbo vietos aprašymas, darbuotojų informacija, darbo vietai priskirtų įrenginių sąrašas ir darbo vietos komentarai, kuriuos galima pridėti bei šalinti šiame lange.

Informacija -

Darbo vieta

Pavadinimas: 'Prie lango'

Pastatas: Administracinis

Aukstas: 4

Kabineto nr.: 411

Greita informacija:

Darbuotojas 1

Vardas: Denas

Pavarde: Pavlavičius

Pareigos: Inžinierius-programuotojas

Skyrius: INFORMATIKOS SKYRIUS

Darbo vietos irenginiai

Tipas	Pavadinimas	Virtualus vardas	Gamintojas	Tiekejias	Serijos nr.	Inventorinis Nr.	Veiksmai
Kompiuteris							
Monitorius							
Monitorius							
Telefonas							
UPS							

Darbo vietos komentarai

Įrašo data	Autorius	Veiksmas	Tekstas
Nėra komentarų...			
+			

Pav 4 Darbo vietos informacija

Iš darbo vietos informacijos lango galima patekti tiesiai į įrenginio informacijos langą (žr. Pav 5), kuriame rodoma pagrindinė įrenginio informacija (pavadinimas, virtualus vardas, gamintojas, tiekėjas, serijos numeris, inventorinis numeris bei įsigijimo data), priskirtos darbo vietos informacija, atributų sąrašas, bei redaguojamas komentarų sąrašas.

Informacija -

Įrenginys

Tipas: 'Kompiuteris'

Pavadinimas: [redacted]

Virtualus vardas: [redacted]

Gamintojas: [redacted]

Tiekėjas: [redacted]

Serijos nr.: [redacted]

Inventorinis nr.: [redacted]

Įsigyjimo data: 2010-06-07

Darbo vieta



Pavadinimas: 'Prie lango'

Pastatas: Administracinis

Aukstas: 4

Kabineto nr.: 411

Atributai	
Atributo pavadinimas	Reikšmė
Naršyklė	Mozilla Firefox
Naršyklė	Chrome
RAM tipas	DDR3
RAM talpa	2X2048 Mb
RAM talpa	2x1024 Mb
CPU tipas	P Dual Core Quad
CPU sparta	Q8400
Office	OpenOffice 3.3
Diskasukis	DVD-RW
Operacinė sistema	WIN 7 Pro (x64)
HDD tipas	SATA
HDD talpa	500
HDD talpa	250
Statusas	Darbo vietoje
OS product key	[redacted]
Garantija	3 m.
MAC adresas	[redacted]
[redacted]	[redacted]

Komentariai				
Įrašo data	Autorius	Veiksmai	Komentaras	
2012-03-14 09:27:53	DenasP	Win Os peristaliavimas	Pereita ant Windows 7 x64	
2011-05-05 16:30:17	DenasP		Pereita prie Mozilla Firebird	
2010-06-09 00:00:00	LinasC	Win Os peristaliavimas	PROFILAKTIKA pilna po suinstaliavimo	
+				

Pav 5 Įrenginio informacija

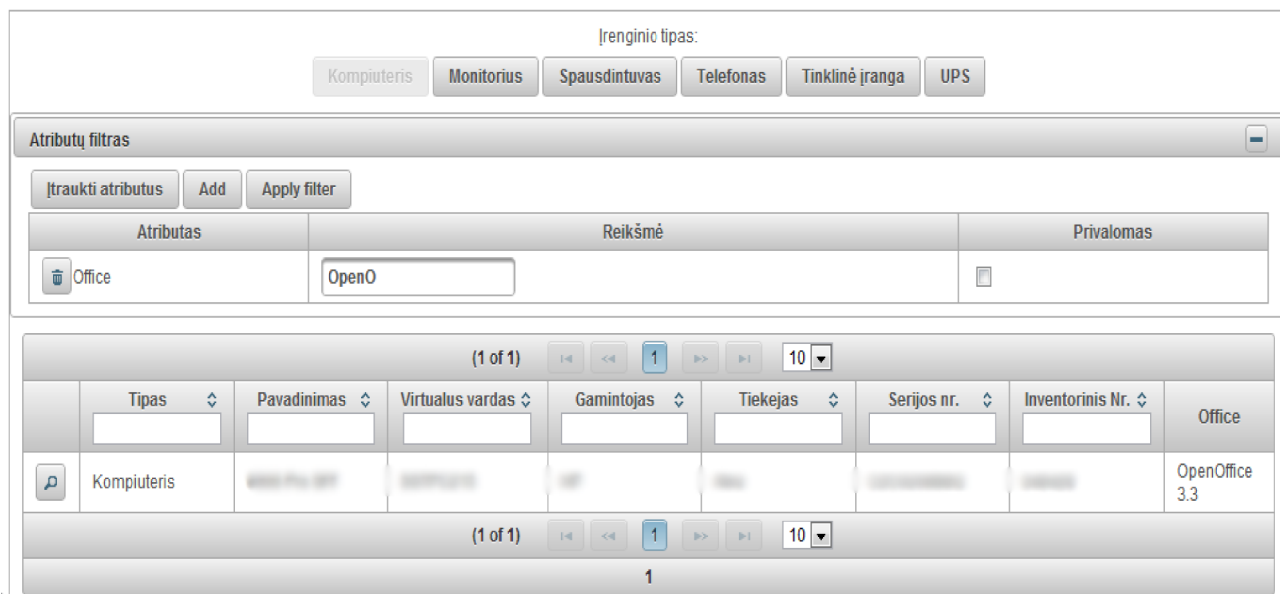
Naujas komentaras kuriamas „+“ ženklo, esančio komentarų sąrašo lentelės paraštėje, pagalba. Vedant naują komentarą reikia suvesti darbuotojo vardą, atliktą veiksmą bei komentaro tekstą (žr. Pav 6).

Iš šio sąrašo komentarai negali būti ištrinti, jie tik gali būti anuluoti. Anuliavus komentarą sąrašė jis išlieka, tik pridamas priedas nurodantis jog komentaras yra anuluotas ir darbuotojo (anuliavusio komentarą) vardas.



Pav 6 Naujo komentaro forma

Įrenginių paiešką galima atlikti pagal tam tikrus atributus (žr. Pav 7). Visų pirma parenkamas ieškomo įrenginio tipas, tada išrenkami atributai, kuriuos bus galima matyti sudarytame sąrašė. Toliau išrinktus atributus galima įtraukti į filtrą ir jį aktyvuoti.



Pav 7 Įrenginių paieška pagal atributus

Likusios informacijos atvaizdavimo skiltys (darbuotojų ir įrenginių) tiesiog atvaizduoja sąrašus su pagrindine informacija, kuriuos vėliau galima filtruoti pagal kiekvieną lauką.

Antrasis sistemos valdymo būdas orientuotas tik į tokias funkcijas, kurios gali būti naudojamos nuotoliniu būdu esant prie tam tikro įrenginio ir neturint priėjimo prie kompiuterio su naršykle. Išmaniojo telefono pagalba (žr. Pav 8) galima nuskanuoti arba įvesti įrenginio kodą ir pagal tai gauti išsamią jo informaciją. Taip pat galima peržiūrėti ir tuo pat metu įvesti naują su šiuo įrenginiu atliktą darbą neatsitraukus nuo jo.



Pav 8 Android aplikacijos langai